WEAKLY SELECTIVE TRAINING INDUCES SPECIALIZATION WITHIN POPULATIONS OF SENSORY NEURONS

Dissertation
for the award of the degree
"Doctor rerum naturalium"
of the Georg-August-Universität Göttingen

within the doctoral program
Sensory and Motor Neuroscience
of the Georg-August University School of Science (GAUSS)

submitted by
JULIA HILLMANN

from Karlsruhe, Germany
Göttingen 2015

THESIS COMMITTEE

FIRST REFEREE AND SUPERVISOR

Dr. Robert Gütig,
Dept. of Theoretical Neuroscience,
Max Planck Institute for Experimental Medicine, Göttingen

SECOND REFEREE

Prof. Dr. Tim Gollisch,
Dept. of Opthalmology,
University Medical Center Göttingen

Prof. Dr. Fred Wolf,
Dept. of Theoretical Neurophysics,
Max Planck Institute for Dynamics and Self-organization, Göttingen

OTHER MEMBERS OF THE EXAMINATION BOARD

Prof. Dr. Thomas Kneib,
Dept. of Statistics,
Georg-August-Universität Göttingen

Prof. Dr. Alexander Gail,
Sensorimotor Group,
German Primate Center Göttingen

Prof. Dr. Tobias Moser,
Institute for Auditory Neuroscience,
University Medical Center Göttingen

Date of oral examination: 11.01.2016

AFFIDAVIT

Here I declare that my doctoral thesis entitled "Weakly Selective Training induces Specialization within Populations of Sensory Neurons" has been written independently with no other sources or aids than quoted.

Julia Hillmann
Göttingen, November2015

ABSTRACT

Many neurons in sensory pathways respond selectively to a narrow class of stimuli such as faces or specific communication calls. At the same time neural processing is robust to a large degree of natural variablity within such complex stimulus classes. For instance, face detection must be robust with respect to a particular hair or eye color and speech processing must tolerate large differences between female and male vocalizations. It is hypothesized that such difficult perceptual invariances might be subserved by populations of neurons that specialize on different substructures within a sensory object category.

However, it is unclear how specialization emerges within a population of neurons during learning. Coordination of individual learning processes between neurons, that would enable a division of the task, has remained biologically challenging. State-of-the-art machine learning approaches that focus mainly on generating diversity to improve classification performance, assume full transparency of all learning processes and therefore provide only limited insights into the mechanisms that lead to specialization in networks of neurons. By contrast, biologically plausible models that are based on majority voting do not result in specialized neural ensembles.

The goal of this thesis is to bridge the gap between biological plausibility and high classification performance by means of specialization. We show that specialization within populations of sensory neurons can emerge by a weakly selective training algorithm that only relies on a global supervisory feedback signal, the Tagging algorithm. In response to this global feedback, neurons decide individually whether to engage in a learning step based on the confidence of their individual decision. We show that the Tagging algorithm induces specialization within neuronal populations not only for specifically tailored classification problems, but also for a real-world spoken digit detection task. Additionally, we demonstrate that weakly selective learning can be applied to a broad range of neuron models, ranging from perceptrons to spike escape models, and can be combined with other learning concepts such as reinforcement learning.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

When we meet a friend in a crowded bar, we usually identify him without mentionable effort. Our brain performs this detection and identification of our friend within a few hundred milliseconds (Thorpe et al. 1996, Hung et al. 2005), despite a visually cluttered environment, including the presence of distracting objects, other bar guests and difficult light conditions. Even though it seems effortless for our brain, visual object recognition is subject to different complex constraints.

On the one hand, object recognition must be highly specific. For instance, we need to separate the face of our friend from faces of similar looking bar guests. On the other hand, we want to detect our friend, regardless of whether we see his face in profile or straight, whether he hides in a dark corner or stands in the spotlight performing a karaoke song. Boundaries between different categories can be highly nonlinear. Sometimes physically similar objects need to be discriminated (the face of our friend and the one of his brother) whereas physically diverse objects need to be grouped together (the face of our friend from the front and from the side). These antagonistic requirements of high *specificity* on the one hand and high *invariance* on the other make visual object recognition a difficult problem. Decades of research in machine learning have tried to understand and recreate the impressive performance of our brain on these complex tasks, however only with very limited success (Pinto et al. 2008, Kreiman 2013).

Identifying visual objects in a natural scene is crucial not only for meeting a friend in a bar but also when it comes to detecting food or predators. A large number of different sensory tasks can be boiled down to binary classification problems where we need to distinguish behaviorally relevant sensory *target* cues from a non-relevant stimulus *null* class. Examples can be found in all sensory modalities: Distinguishing the smell of fire from irrelevant other odors appears as important as the ability of animals to identify alarm calls that indicate the approach of a predator (Seyfarth et al. 1980). All of these tasks share the requirement of tolerating category-irrelevant features while selectively responding to task-relevant features.

How can such a complicated problem be solved? One possible way would be to distribute the complex task across a large number of classifying units and thereby divide the labor. The large number of densely connected neurons suggests the use of a similar strategy also in the human brain. Intuitively, if all neurons involved in a sensory detection task respond similarly, i.e. in a highly correlated manner to the same inputs, they do not provide additional information over a single unit. However, when individual units divide the task among each other and specialize on easier subproblems, a population solution to the complex problem becomes available. Indeed, a large number of experimental studies report that neurons in several higher brain areas respond

selectively to small subsets of sensory stimuli, suggesting a specialization on subsets of the problem space. Examples include visual neurons that are solely activated by the presence of a hand in the visual scene (for example Desimone et al. 1984) or auditory neurons that respond selectively to food calls (Gifford et al. 2005). Together with the finding that learning sharpens this selectivity (Sigala and Logothetis 2002), the high abundance of selective cells provides evidence for the theory of specialization as a means of high population accuracy on complex sensory detection problems.

Also several studies in machine learning have highlighted the role of specialization within a group of individual classifying units. Classification performance of an ensemble of learners improves when the responses of individual learners are diverse. Under certain conditions, this relation was even mathematically proved (Krogh and Vedelsby 1995). Hence, several attempts have been made to develop learning mechanisms that decorrelate the individual learners' responses. Most of these mechanisms aim for optimal classification performance and to this end treat the ensemble as omniscient to utilize all available information from the individual learners. Neurons, however, do not have information about internal states of neighboring neurons in a population. Therefore, the impact of standard machine learning techniques on explaining selectivity and learning in neurobiological ensembles has been limited. Only very few methods actually take into account biological constraints when training a population of artificial neurons. However, these models rely on complex multi-level neuronal architectures or allow for specialization within the population only to a very limited degree.
The goal of this thesis is to bridge the gap between biological plausibility and high classification performance. We propose a neuronal population learning method that relies on a simple two layer architecture. Its key component is a weakly selective training mechanism that induces learning only within subsets of neurons, as opposed to unselective training on all neurons. Selection of neurons is based on a local rule of synaptic plasticity and thereby enables biologically plausible specialization within the population.

Biological plausibility, however, can be interpreted differently, especially with respect to the information that is accessible to individual neurons. In the following section we define our understanding of biological plausibility and thereby set up the framework for this thesis. State-of-the-art machine learning models to generate diverse ensembles are summarized and discussed also in the context of this definition. It will turn out that the only method that agrees completely with these constraints offers specialization only to a very limited degree. To motivate that specialization is a common principle in sensory areas and thereby a desired property of neuronal ensembles after learning, we briefly review experimental evidence for highly selective neurons in the mammalian brain, before we present a guideline through this thesis. In the following we assume that the reader is familiar with a basic neuroscientific vocabulary, including the notion of spikes, postsynaptic potentials and synaptic transmission. For the naive reader, however, we provide a very short summary about the commonly used terms in the appendix (Section A.1).

1.1 THE CLASSIFICATION PROBLEM AND BIOLOGICAL CONSTRAINTS

To give a precise understanding of what we expect from the population algorithm that we propose in this thesis, we define the general framework including the underlying classification problem and the biological constraints in this section.

Classification Task

Throughout the thesis, we consider a *binary classification task*, i.e. the input pattern x (a sensory stimulus, represented by spike patterns) belongs to either a behaviorally relevant *target class* or a distracting *null class*. The class affiliation of each sensory input pattern is indicated by a binary label $\ell \in \{0, 1\}$ where $\ell = 1$ for patterns from the target class. We assume that this target class contains structure that is not resolved by the label. Specifically, the target class can be either partitioned into distinct discrete sub classes or it contains objects that vary with a continuous parameter. Examples for the first type are two different types of food calls by monkeys in a study of Gifford et al. (2005), but one might also think of different visual food objects in an abstract classification problem to distinguish food- from non-food objects. Continuously varying subclasses include different head orientations in face detection problems (Freiwald and Tsao 2010) or different handwriting styles in handwritten digit recognition.

Biological Plausibility

We base the thesis on the intuition that populations of neurons achieve better accuracy on this kind of classification problems if they divide the problem space among its members. With unrestricted information exchange, neurons could agree on splitting up responsibilities according to relative preferences or competences during the learning process. However, neurons in the brain do not have access to internal states of neighboring neurons. To set up the constraints for the neuronal population, we define a neuronal ensemble[1] as *biological plausible* if it fulfills the following requirements:

1. The inputs to processing neurons of the ensemble are represented in the form of spike patterns, defined as a set of action potential times.

2. The processing neurons communicate via action potentials with the output neuron.

3. The processing neurons only receive a common global teacher signal. This signal carries information if the population decision is correct or, in case of a misclassification, whether the input pattern belonged to the target or null class.

Note, that these constraints are conservative in the sense that the only information available to each neuron is its own state and the global teacher signal. We will review the state-of-the-art machine learning approaches for populations of classifying units with a focus on how well they fit into our framework of a biologically plausible learning mechanism.

1 We will use the notations *neuronal ensemble* and *neuronal population* equivalently throughout the thesis.

1.2 MODELS TO INDUCE SPECIALIZATION

In their role as object or feature detectors, neurons can also be idealized as binary classifiers and consequently, a neuronal population as a combination of classifiers. General findings in machine learning ensemble methods hence also have implications for models of biological neuronal populations.

Machine learning ensembles

Combining different classifiers or *learners* to improve classification performance has a fifty year old history in machine learning (see Sharkey 1996) with an upswing in the 90s. Around that time, Shapire (1990) proved that so called *strongly learnable* and *weakly learnable* problems are equivalent. The theorem states that weak classifiers that solve the problem each with performances slightly above chance level can be combined to achieve close to perfect classification. Shapire's proof is constructive, leading to the first version of *Boosting*. The later published version of AdaBoost (Freund and Shapire 1997, Freund and Shapire 1999) is still treated as one of the best off-the-shelf ensemble methods and forms the basis of many different Boosting algorithms (Kuncheva 2004, Mayr et al. 2014). It iteratively trains classifiers (for example also neural networks) particularly on data examples that the previous learners have misclassified and thereby uncorrelates the errors within the ensemble. This idea of uncorrelating the errors or generating diversity within the ensemble is the crucial component for a performance improvement. For regression problems, Krogh and Vedelsby (1995) indeed showed that the generalization error of a group of classifiers that bases its decision on a weighted average can be decomposed into two components. One component describes the weighted average of the generalization errors of the individual learners, while the second negative summand denotes the weighted average of the squared deviations of the individual decisions from the population decision. The more diverse the learners are, the larger the second summand gets, reducing the overall population error[2].

Generating diversity among members of a classifier ensemble can be achieved in a number of ways. Especially for neural networks, this includes initializing the neuronal learners with different synaptic weights, different network topology or training them with different learning algorithms (Sharkey 1996). While at least the first two ideas are consistent with a biologically plausible training, none of these simple methods provides substantial classification improvements.

Another approach is to train neurons on slightly different training data. Apart from Boosting, also *Bagging* (Breiman 1996) and *cross-validation training* (Krogh and Vedelsby 1995) make use of this strategy. In Boosting, as mentioned above, neurons are trained sequentially on input patterns that the previously trained neuron have misclassified. This implies that one neuron first learns to solve the problem as well as possible, before the second neuron is trained on the misclassified patterns of the

2 It should be noted, however, that for (binary) classification problems, no such neat theory exist, see also Brown, Wyatt, Harris, et al. (2005)

previous neuron. This learning scheme violates several of the biological constraints, including that neurons receive information about the performance of previously trained neurons. Bagging, on the other hand, does not make use of sequential training to diversify the training data, but trains individual neurons on random bootstrap samples of the input data. This strategy enforces a kind of randomized specialization. Even though this procedure does not violate any of the constraints on biological plausibility explicitly, we do not expect emergence of a clear specialization within the set of neurons by this random training. Cross-validation training works in a similar manner, except that the individual classifiers are trained on distinct cross-validation subsets instead of potentially overlapping bootstrap samples. This arbitrary separation of the training data set would even imply knowledge of which neuron learns which patterns to ensure non-overlapping cross-validation samples. In the above population approaches, once diverse learners have been generated, their outputs are combined via (weighted) averaging or voting.

Often, classifiers are trained to minimize an objective function that incorporates the deviations of the pattern labels from the population outputs. For this kind of learning, diversity can also be explicitly induced by adding a penalty term to the objective function that actively forces learners to avoid correlated decisions (*negative correlation learning*, Liu and Yao 1999). Specifically, for online updates in neural networks, synaptic weights are adjusted to simultaneously minimize deviations of the individual response from the pattern label as well as from the average population output. For this setting, individual neurons require knowledge about the label as well as the population decision, which is indeed in agreement with the information available in a biologically plausible population as defined before. However, neurons in the populations are non-spiking and inputs are represented by analog numbers. Additionally, Liu and Yao (1999) utilize the standard backpropagation algorithm to perform the actual weight changes. This algorithm assumes that all layers in a multi-layer network have access to the error at the final output stage, which is biologically unlikely. While a spiking implementation of this model might be an interesting research project, it is beyond the scope of this thesis.

Reviews on classifier combinations (Kuncheva 2004, Sharkey 1996) distinguish the above *ensemble methods* where every learner basically solves the same problem, from *modular approaches* that divide the problem into smaller sub problems that are tackled by individual experts. The most common representative of this class of algorithms is the mixture of experts model (Jacobs et al. 1991). It translates the mathematical theorem of total probability to a two layer network structure where a set of gating neurons models the probabilities that an input pattern belongs to a certain subregion of the input space. They gate input patterns to specific expert neurons that then classify the observations that fall into their specific expert region. Those expert neurons represent the conditional probabilities that the pattern corresponds to the target class, given that it belongs to the expert's subregion of the problem space. This model yields highly specialized expert neurons after training and thereby improves classification performance significantly over individual neurons. Individual gating and expert networks

can even be stacked in a hierarchical architecture to improve performance on difficult classification problems (Yuksel et al. 2012, Jordan 1994). However, due to the complex gating and expert structure that requires internal information of neighboring neurons during the learning process, a biological implementation of mixture of experts models has remained challenging. Even though it violates the third constraint on biological plausibility, we will use the basic version of this method for comparisons in a later chapter of this thesis and hence will introduce it in Section 2.2.2 in greater detail.

More recent attempts in the field of classifier combination also include multistage (S. Yang and Browne 2004, J. Yang et al. 2013) and hybrid methods (such as combining negative correlation learning with mixture of experts, Masoudnia et al. 2012) as well as training on input feature subsets to induce diversity (Kheradpisheh et al. 2014). For neural networks, especially the dropout procedure has recently gained attention and is noted here even though it strictly does not belong to the family of classifier ensemble methods (Hinton, Srivastava, et al. 2012). It randomly disables 50% of the units in the hidden layers of a multilayer network (and additionally 20% of the input layer) during training. With this strategy, the neurons within the network develop a relatively robust representation of the input space because they can not rely on responses of individual units to generate the final output.

In general, all of these methods focus on optimizing classification performance and hence usually utilize omniscient classifiers and complex mathematical operations to achieve this goal. Due to violations of the assumptions on a biological plausible population, most of these methods only have limited impact on explaining neuronal selectivity.

Biologically inspired models

For visual object recognition, many attempts have been made to explain the great selectivity and invariance of neurons in higher visual areas (for a recent review, see Kreiman 2013). Several studies (Riesenhuber and Poggio 2002, Riesenhuber and Poggio 2000, Serre et al. 2007, Anselmi et al. 2015, LeCun 2012) have suggested a feedforward-processing network, motivated by the fast activity of neurons in the inferotemporal cortex (*ITC*), one of the main regions involved in visual object recognition (Hung et al. 2005). Their proposed models are based on a hierarchical structure that is inspired by Hubel and Wiesel (1962)'s simple and complex cells. Both cell types were found in the cat visual cortex and are characterized by the following properties: Simple cells are maximally active if edge-like structures are present in their receptive field. Complex cells differ from simple cells in the size of their receptive fields. Additionally, their responses are invariant to the position of the object in their receptive field. The above theoretical approaches stack models of these cell types in multiple hierarchies, every simple cell layer (S-layer) is followed by a complex cell layer (C-layer). Simple cells combine information from a set of connected complex cells to generate features of higher complexity and thereby increase selectivity within this layer. On the other hand, complex cells pool responses from simple cells that are tuned to the same

feature but respond when the feature is located at different spatial positions, consequently increasing the invariance to position in the visual space. The operation to pool responses from S-layers is often equivalent to a MAX-operation (or OR-operation), ensuring invariance to position, whereas simple cells subsequently combine responses from the C-layer in an AND-like operation, generating selective responses to more complex features. By hierarchically mixing these two operations, highly selective neurons with highly invariant responses are generated on the upper layer. Combined with a supervised learning step on top of the hierarchy, these models yield classification performances similar to human performance on a visual 'animal-vs-non-animal' classification task (Serre et al. 2007).

Different models inspired by simple and complex cells differ for example in the way simple cells pool the responses from complex cell layers, the actual stacking of layers or the activation function of the neurons. One important subclass of these hierarchical models with simple and complex-like cells are *convolutional neural networks*. LeCun et al. (2015) characterize them by four main concepts: local connections (in contrast to fully connected networks, yielding the notion of receptive fields of cells), shared weights between neurons in the simple cell layer (to obtain a common preferred visual feature within one layer), pooling and the use of many layers. Herein, the layers of simple cells are equivalently denoted by convolutional layers since the filtering operation based on shared weights to generate the simple cell activation mathematically agrees with a discrete convolution function. Generally, all of these methods can be summarized under the name *deep learning* or more specifically *deep neural networks* that have gained a lot of attraction across different research fields within the last decades (LeCun et al. 2015). Especially convolutional neural networks have shown striking success in different areas and beat standard machine learning approaches on complex classification tasks such as visual image recognition when used in combination with the previously described dropout procedure (Krizhevsky et al. 2012).

In the previously described models, 'neurons' process information by mathematical operations and synaptic weights are optimized to minimize an objective error function. The actual learning process usually follows the standard backpropagation scheme, which requires knowledge of the errors within all layers of the network, and hence, biologically questionable. Additionally, the inputs and outputs of individual units in the network are continuous numbers and not spike patterns. This representation of inputs and outputs contradicts the first two requirements on information transmission via spiking responses in our definition of a biologically plausible population.

Only very few studies intended to explain how specificity in neurons evolves with biologically plausible adjustment of the synaptic efficacies. One recent biologically inspired population approach is the attenuated learning procedure by Urbanczik and Senn (2009) that was originally defined for stochastic neurons with reinforcement learning strategies. Here, a neuronal population bases its decision on majority voting and the intensity of learning depends on the reliability of the whole population.

Specifically, neurons that have generated an erroneous response update their synaptic efficacies on every individual error with a learning rate η. This learning rate is maximal if also the population decision is wrong for a given pattern, and minimal if no other neuron in the population misclassified the pattern. In between, learning strength decreases with the number of correctly responding neurons. This algorithm complies with the constraints on a biologically plausible population learning mechanism. However, since all erroneously responding neurons are updated with a common learning rate and the population decision relies on majority voting based on the individual decisions, the algorithm's ability to induce specialization among the neurons is limited. Due to its biological plausibility, we will utilize the attenuated learning in a comparative way throughout this thesis and explain the exact learning rate attenuation procedure in more detail in the methods section.

1.3 EVIDENCE FOR SELECTIVITY IN THE BRAIN

Evidence for highly selective cells is found throughout the brain. Most research on feature selectivity has been conducted in the visual system, partly because in vision, the notion of an 'object' or a 'feature' is the most concrete. We hence focus on the visual system and only briefly review findings from auditory areas.

1.3.1 *Visual object and face detection*

Information about detecting objects in a visual scene, like for example spotting our friend in a bar, is processed in the ventral visual pathway of the mammalian brain. This pathway is mainly involved in answering the 'what?' (is present in the visual scene) question irrespective of variations in the objects' appearance, like different viewing points or spatial positions, and includes a number of differently specialized brain areas. In contrast, the dorsal pathway encodes the 'where?' (is object is present in the visual scene) question that is of minor relevance for this literature review on visual object recognition.

Whereas simple visual features such as orientations are well represented in earlier stages of the ventral visual pathway (orientation selective cells in V1, see also Chapter 12 in Purves et al. 2008), more complex object classes can be distinguished based on neuronal responses in later stages, suggesting a hierarchical representation (Cichy et al. 2014). For object recognition, several studies have highlighted the major role of the inferior temporal cortex (*ITC*, DiCarlo et al. 2012, Kourtzi and DiCarlo 2006). First experimental evidence for highly specific neurons that only respond to a small set of relatively complex shape stimuli in the ITC goes back to the 1970s and was discovered 'by chance':

> 'One day [...] having failed to drive a unit with any light stimulus, we waved a hand at the stimulus screen and elicited a very vigorous response from the previously unresponsive neuron. We then spent the next 12 hr testing various paper cutouts in an attempt to find the trigger feature for

> this unit. When the entire set of stimuli used were ranked according to the strength of the response that they produced, we could not find a simple physical dimension that correlated with this rank order. However, the rank order did correlate with similarity (for us) to the shadow of a monkey hand.' (C. Gross et al. 1972, page 103-104).

Subsequent recordings demonstrated that a large fraction of neurons in monkey ITC fires maximally to complex object (categories) such as brushes (C. G. Gross et al. 1977) or hands (Desimone et al. 1984) but to a much lesser degree to simpler stimuli like gratings or object components (also defined as the *critical feature* by Kobatake and Tanaka (1994)).

Face recognition as a well described sub-system

From the set of highly specific neurons in the ITC, a relatively large subpopulation was found to elicit maximal responses exclusively to human and monkey faces (Desimone et al. 1984, Kobatake and Tanaka 1994). In addition, fRMI studies suggest that separate regions in the temporal lobe are dedicated to face detection in the human and monkey brain (Tsao et al. 2003). By using a visual stimulus set consisting of cartoon faces, Freiwald et al. (2009) investigated the response tuning of single neurons in these face regions on different facial features, such as inter-eye-distance or mouth size. They found that the firing rates of individual neurons varied systematically dependent on usually two to four facial features and remained unaffected by others.

Due to its relatively localized processing area and high concentration of selective cells, the face detection system is well suited for studying general principles of visual recognition (for a review on findings, see also Tsao and Livingstone 2008). For example, by presenting face images from different angles, Freiwald and Tsao (2010) uncovered hierarchically evolving view-tolerance. While neurons in the middle face patches (middle lateral and middle fundus, *ML* and *MF*) showed view-specific responses, two potentially connected downstream areas contained neurons that responded with similar intensity to mirrored faces (anterior lateral patch, *AL*) and view-invariant cells in the most anterior region (anterior medial, *AM*). Also in the human brain, evidence for neurons that respond similarly to mirror-symmetric faces was provided by subsequent functional magnetic resonance imaging (*fMRI*) studies (Kietzmann et al. 2012).

Representation of visual objects in human higher brain areas

Strong projections from the ventral visual pathway to the higher association areas in the temporal lobe suggest an involvement of those areas in object recognition. While recording from intracranial depth electrodes in epilepsy patients, Kreiman et al. (2000) found a significant number of neurons to be category-selective. Those neurons responded specifically to one out of nine visual object categories including for example household objects, faces of unknown actors, animals and pictures as well as drawings from known actors. Within each stimulus category, most neurons showed relatively flat tuning, indicating a class-specific representation. Much narrower categories that

specifically triggered neuronal responses were reported by Quian Quiroga et al. (2005). They found a large fraction of neurons that were selectively activated by individual objects or persons, highly invariant to position and view changes. Their responses were very sparse, each neuron was excited by only 2.8% of the images on average. Interestingly, some neurons did not only fire significantly for images of a specific famous persons such as the actress Halle Berry, but also to drawings of her and even her written name. This neuron has been interpreted as incorporating the 'concept' of Halle Berry. However, this interpretation as 'concept cells', sometimes also refered to as 'grandmother cells', is highly debatted (Bowers 2009, Roy 2012, Roy 2013, for a critical command see also Quian Quiroga and Kreiman 2010). Many other examples for selective cells were found in the human temporal lobe (Quian Quiroga et al. 2007, Ison et al. 2011), suggesting increasing selectivity in higher brain areas (Ison et al. 2011, Mormann et al. 2008). However, a recent study (Valdez et al. 2015) reports less sparse responses (neurons responding to more than 20% of the images). Valdez et al. (2015) hypothesize that the finding of the grandmother cell-like neurons in previous studies are due to learning effects that result from frequent presentations of the same images.

Visual object recognition is plastic: repetition sharpens selectivity

As described above, neurons in the IT cortex of monkeys have been shown to selectively respond to visual objects. However, the previously described studies did not investigate if this tuning changes over time. One study that suggests a possible answer to this questions was performed by Sigala and Logothetis (2002). In a categorization task, monkeys were trained to discriminate visual images of faces that differed in their eye height, eye separation, nose length and mouth height. However, the label of these input faces depended only on two of these face features ('diagnostic' features), whereas the other two were only varied for distraction. After the monkeys learned to distinguish both classes, the authors recorded from neurons in the anterior ITC. Out of the set of active neurons, 70% were selective to the diagnostic features. In agreement with the previous study, Freedman et al. (2006) reported that ITC neurons were more selectively tuned to a set of visual images of cat and dog morphs that had been presented in a training paradigm than to rotated versions of the same images that did not appear during the training phase. Together, these studies indicate that training shapes feature selectivity.

How is learning controlled in the ITC during training? Do neurons in the ITC receive feedback signals from a teacher instance to change their firing behavior? One possible candidate for the source of top-down feedback is the prefrontal cortex (PFC), a downstream area of the ventral visual pathway (for evidence of possible top-down connectivity, see Miller and D'Esposito 2005). Freedman et al. (2003) investigated neuronal responses in the ITC and PFC simultaneously in an object recognition task. Visual stimuli consisted of morphed images between six prototype dogs and cats. Monkeys were trained to decide if two sequentially shown morphs belong to the same category (both cats/ both dogs) or to different categories (one cat, one dog).

After training, most neurons from the PFC showed clearly distinct responses to both classes, whereas early responses from the ITC cells depended more strongly on the actually viewed shape than on its class affiliation. However, after some time delay, information about the input category was also present in ITC cell responses. These late signals might originate from top-down feedback from the PFC neurons, that kept their category-specificity throughout the whole stimulus trial. Citing Freedman et al. (2003): 'the PFC seems more "behavioral", whereas the ITC seems more "visual"'. Consistent with the behavioral role of the PFC, category-selective responses of PFC neurons were enhanced when images were explored in a task-context compared to passive viewing (McKee et al. 2014). This task-involvement was reflected in ITC neuron responses only to a much weaker degree and with long latencies. These findings also support the idea of a top-down feedback signal from the behaviorally more involved PFC.

1.3.2 *Objects and selectivity in the auditory system*

Even though coming up with a clear definition is not as straightforward as in the visual system, we have an intuitive understanding of what an auditory object is. We automatically assign a song of a bird, squealing tires or voices to the presence of a bird, a car and a person, respectively. The auditory environment in which we need to detect the object can be highly complex with different distracting noise sources, similar to a cluttered visual image. Indeed, a number of studies indicate that also neuronal processing of auditory information is similar to that of visual information, including the existence of a ventral 'what' and a dorsal 'where' pathway (for a summary see King and Nelken 2009 and Bizley and Y. Cohen 2013). Based on a set of different monkey communication calls presented from different spatial orientations, Tian et al. (2001) revealed two distinct neuronal populations in the primary auditory cortex (A1), that showed selective firing either to the type of the monkey call or to the spatial orientation, but responses that were invariant to the respective other feature. Subsequent studies found neurons with high response selectivity also in different other regions of the primary auditory cortex (Kusmierek et al. 2012, Tsunada et al. 2011, Myers et al. 2009). Even a 'voice region' analogous to the visual 'face region' has been identified recently (Petkov et al. 2008).

Analogous to the findings in the visual system, the left superior temporal gyrus in a human fMRI study showed graded responses to morphings between spoken syllables (Myers et al. 2009). This indicates a tuning to the features that generated the categories, instead of a discrete response to behaviorally meaningful categories (syllables), similarly to visual ITC neurons. Also electrode recordings from homologue processing areas in monkeys found subsets of neurons with graded responses to similar syllable morphs (Tsunada et al. 2011). The majority of neurons, however, generated spike responses that contained category-specific information when the monkey was trained to distinguish both prototype syllables. However, the predicted class from the population's responses did not always coincide with the behavioral decision of the monkey. Again, a possible explanation would be that the superior temporal regions act in a

more 'auditory' way, and a downstream area, such as the prefrontal cortex, takes the 'behavioral' role (see also Ley et al. 2014). Indeed, neurons in the PFC showed response selectively to higher order categorization of rhesus monkey food calls. Neuronal firing in the PFC could reliably separate 'grunts' that indicate low quality food from two acoustically different 'harmonic arches' and 'warbles' calls that both represent high-quality food (Gifford et al. 2005). These qualitatively similar responses to two physically different stimuli that belong to the same behavioral category are consistent with the role of the PFC as planning and decision stage (see also chapter 26 of Purves et al. 2008).

1.4 STRUCTURE OF THE THESIS

As shown in the previous sections, a vast number of studies indicate that selectivity or specialization is a wide-spread phenomenon in several higher brain areas. Consistent with theoretical investigations in machine learning, specialized learners within a population facilitate solutions to complex classification problems. We assume that also neuronal specialization in sensory processing arises from the need of high classification performance on sensory detection tasks. The aim of this work is to propose a simple population learning method that facilitates the emergence of specialization during learning within a biologically plausible neuronal ensemble. Our attempts to achieve this goal are structured in this thesis as follows:

To model a biologically plausible neuronal population that induces specialization, several components are needed. We build the ensemble mainly on a biologically inspired neuron model, the *tempotron* (Gütig and Sompolinsky 2006). The notations and learning rules underlying the tempotron and two other neuron models are defined in the following methods Chapter 2. Additionally, we describe population approaches for comparison purposes and introduce the general classification paradigm that mimics the sensory detection task.

Chapter 3 is dedicated to the proposed population learning, the *Tagging* algorithm. It is based on a simple population decision (directly obtained from the number of active neurons) and selectively induces learning on error trials only to a subset of neurons. The 'tagging' or selection of the neuronal subset that undergoes learning is determined by locally available individual internal states in combination with the global population feedback, thereby fulfilling the requirements of a biologically plausible population.

Is this weakly selective training sufficient to induce specialization within a neuronal population? Chapter 4 investigates this question in great detail on a classification problem with poisson input spike patterns. To fully control the classification paradigm, we use specifically tailored target classes that are composed of distinct subclasses (such as the two different monkey calls both indicating the presence of high-quality food,

see Gifford et al. 2005). On a wide range of input scenarios, we test the Tagging algorithm's capability to induce specialization within a population of tempotron neurons. We find that specialization emerges not only along the tailored subclasses, but also within individual subclasses by means of spreading decisions in time.

To extend the biological validity of our research, we evaluate the Tagging algorithm on a real-world spoken digit discrimination task in Chapter 5. Here, input patterns consist of acoustic waveforms that are transfered to spike trains using a simplified model of auditory neurons. Also for this more realistic setting, the Tagging algorithm shows tendencies for selective responses within the target digit class that can be attributed to the gender or identity of the speaker.

The Tagging algorithm is not restricted to a specific neuron model. Chapters 6 and 7 are dedicated to illustrate the flexibility of the proposed learning algorithm by applying Tagging to the classical perceptron model and a neuron model with stochastically fluctuating firing threshold, respectively.

We summarize all results obtained with the Tagging algorithm in the concluding Chapter 8 and discuss their implication for further research goals also in the context of the previously described biological findings.

2

METHODS

The main goal of this thesis is to introduce a biologically plausible population learning mechanism that improves classification performance on complex target detection problems by means of induced specialization among the neurons in the population. The algorithm that we will propose in Chapter 3 is generally applicable to different neuron models. Here, we describe three different models that we will base the population on. The neuron models differ in their biological plausibility but also in their way to generate an output spike. For a binary classification problem, the spike or no-spike response of the neuron provides the binary decision on an input pattern. Different synaptic learning rules are used to modify synaptic weights and correct the response behavior of the neuron in case of an erroneous decision.
The second section describes a set of naive as well as state-of-the-art population mechanisms that we will use for comparison with the proposed algorithm throughout the thesis. Population algorithms are evaluated on binary classification tasks where patterns from a behaviorally relevant target class need to be discriminated from patterns from a broad null class. To model structure in the target class, we generate input spike patterns as poisson spike trains that are either noisy copies of discrete templates or determined by a continuously varying parameter (Section 2.3). Additionally, we use spike trains transformed from auditory signals in a speech detection study in a later chapter. Apart from the misclassification error as a measure for the classification performance, we will use three different measures of specialization to analyze the population's qualitative behavior, introduced in the last Section 2.4.

2.1 NEURON MODELS

The first attempts to describe the information transmission of nerve cells with mathematical equations date back to the 1940s (McCulloch and Pitts 1943). The McCulloch-Pitts neuron models the generation of a spike by simply thresholding a weighted sum of input scalar values. A more realistic model that integrates input spike that arrive from different synapses at different points in time is the leaky *integrate-and-fire* model. Here, a time-dependent voltage trace can be explicitly calculated and spikes are generated if this voltage exceeds a firing threshold. Besides the deterministic spiking event whenever this threshold is crossed, also stochastic variants of this model exists.
Since learning is an important issue in this thesis, we are not only interested in a realistic description of the information integration and spike generation process, but also in mechanisms to change the response of these artificial neurons in a supervised learning task. Several learning mechanisms have been developed to modify the synaptic input weights to guide the spiking response of the neurons to a desired direction. The simple *perceptron* learning rule that was developed by Rosenblatt in the sixties (Rosenblatt 1958, Rosenblatt 1962) just adds a certain fraction of the target pattern coordinates to

the synaptic weights of a McCulloch-Pitts neuron. Our main results with be based on the more realistic *tempotron* learning rule that can be applied to the integrate-and-fire neuron (Gütig and Sompolinsky 2006). Additionally, we will describe a reinforcement learning approach for stochastically firing neurons.

2.1.1 *Modeling Neuronal Information Processing*

The McCulloch-Pitts Neuron

The McCulloch-Pitts neuron is one of the simplest models of neuronal information processing (McCulloch and Pitts 1943, a more condensed summary can be found in section 1.1 of Hertz et al. 1991). It receives inputs through a number of n synapses in the form of a simple real value per input synapse. These inputs are multiplied by the weights of the corresponding synapses, summed across all afferents and furthermore compared to a firing threshold ϑ to generate the output of the neuron. Specifically, with synaptic efficacies w_i ($i = 1 \dots n$) for the n input synapses, the binary response o of the McCulloch-Pitts neuron is given by

$$o = \operatorname{sgn}(\boldsymbol{w} \cdot \boldsymbol{x} - \vartheta)$$

In this equation, $\boldsymbol{w} \cdot \boldsymbol{x}$ denotes the scalar product of the vectors $\boldsymbol{x}$ and $\boldsymbol{w}$ and sgn the sign function. It is defined as 1 if the the argument is non-negative, and -1 otherwise. The input pattern $\boldsymbol{x}$ describes a vector of scalar input values, one for each input afferent, and hence can be interpreted for example as the input firing rate to the processing neurons (even though theoretically also negative values are allowed).

The firing threshold ϑ is usually implicitly encoded in an additional weight w_0 that receives a constant 'input' $x_0 = c$ for all patterns. This notation allows for the simpler representation of

$$o = \operatorname{sgn}(\boldsymbol{w} \cdot \boldsymbol{x})$$

with $\boldsymbol{w} = (w_0, w_1, \dots, w_n)$ and $\boldsymbol{x} = (c, x_1, \dots, x_n)$ that we will use in the following.

The Integrate-and-Fire Neuron

A biological neuron receives information from input synapses not in the form of simple numbers, but as a temporal sequence of action potentials or spikes. A more realistic model that allows for a temporally varying input is the integrate-and-fire neuron (see section 5.4 of Dayan and Abbott 2001). In the basic variant of this model, the voltage $V(t)$ of a neuron changes dependent on the current voltage and a time-dependent input multiplied by the membrane resistance of the neuron:

$$\tau_m \frac{dV}{dt} = V_{rest} - V(t) + R_m I_e(t) \tag{2.1}$$

Here, τ_m denotes the membrane time constant of the neuron, V_{rest} the resting potential, R_m the membrane resistance and I_e the external input current. This input current can correspond to an explicit constant or time-varying electrical stimulation, but also

to spiking inputs from presynaptic neurons. For this second assumption that we will consider throughout the thesis, one common approach is to model the current that is induced by each input spike by an exponentially decaying kernel. Specifically, with a kernel proportional to $\exp(-t/\tau_s)$ for $t \geq 0$, the influence of each input spike decays exponentially with a time constant τ_s that should be smaller than the membrane integration constant τ_m. One advantage of an exponentially decaying kernel is that the differential equation (2.1) can be solved analytically. With $X_i = \{t_{i,1}, \ldots, t_{i,nt_i}\}$ denoting the input spike train to synapse i as the set of nt_i (sorted) input spike times, an explicit representation of the voltage $V(t)$ as a function of the current time t is given by:

$$V(t) = \sum_{i=1}^{n} w_i \sum_{t_i \in X_i} K(t - t_i) + V_{rest} \tag{2.2}$$

As for the McCulloch-Pitts neuron, n denotes the number of input synapses and w_i the corresponding efficacies for each synapse. The kernel function K results from the exponential decay of an input spike and is evaluated relative to the input spike time t_i of the spike occurring at input synapse i. Specifically, the solution of the differential equation yields

$$K(t) = V_0 \left(\exp(-t/\tau_m) - \exp(-t/\tau_s)\right) \tag{2.3}$$

for $t \geq 0$ and $K(t) = 0$ otherwise. Herein, for an accordingly chosen membrane resistance, $V_0 = \left(\exp\left(\frac{\log(\tau_m/\tau_s)}{1-\tau_m/\tau_s}\right) - \exp\left(\frac{\log(\tau_m/\tau_s)}{\tau_s/\tau_m - 1}\right)\right)^{-1}$ is a normalization factor that scales the amplitude of the kernel to 1. With this normalization, the synaptic efficacies directly translate to the strength of the postsynaptic potential (*PSP*). The time constants τ_m and τ_s regulate the speed of the exponential increase and decay, respectively. Reasonable parameters that will be used in most of the following simulations are $\tau_m = 10$ms and $\tau_s = 2.5$ms, yielding $V_0 \approx 2.12$.

Whenever the so integrated membrane potential exceeds a certain firing threshold ϑ for time t, an output spike is generated (in accordance with Gütig and Sompolinsky 2006, we will use $\vartheta = 1$ throughout the thesis). All input events that arrive after this threshold crossing are ignored so that the voltage softly decreases down to the resting potential V_{rest}.

The Integrate-and-Fire Neuron with Stochastic Threshold

In its above definition, the integrate-and-fire neuron fires a spike in a deterministic manner exactly -and exclusively- at that time point where the membrane potential crosses the firing threshold. Additional spikes are unlikely since we shunt down all further input spikes after the generation of a single output spike. Here, two different modifications are possible: (1) We allow for firing of additional spikes, for example by resetting the voltage immediately after the output spike and proceed with the integration of further input spikes from the resting potential. (2) We include noise in the model by making the spike generation process stochastic, with a firing probability that depends on the current voltage.
A neuron model that incorporated both ideas is the escape noise neuron that is de-

fined in chapter 5.3 of Gerstner and Kistler (2002). For the description of the model, we follow the Supplementary information of Urbanczik and Senn (2009) and use similar notations.

The voltage $V(t)$ at time t is defined similarly as in equation (2.2) with an additional reset term. Specifically, for every spike s in the output spike train Y, denoted in analogy to X_i as the set of sorted output spikes $Y = \{s_1, \ldots, s_{nt}\}$, we get:

$$V(t) = \sum_{i=1}^{n} w_i \sum_{t_i \in X_i} \epsilon(t - t_i) + V_{Rest} - \sum_{s \in Y} \kappa(t - s) \tag{2.4}$$

To be consistent with the study of Urbanczik and Senn (2009), the double exponential kernel $\epsilon(t)$ is used without standardized amplitude and directly defined as

$$\epsilon(t) = \frac{1}{\tau_m - \tau_s} \left(\exp(-t/\tau_m) - \exp(-t/\tau_s)\right)$$

with membrane constants τ_m and τ_s. The reset kernel $\kappa(t) = \frac{1}{\tau_m} e^{-t/\tau_m} + C_0$ is normalized by $C_0 = (1 + 1/\tau_m)$ to yield a maximal reset of 1 for $t = 0$[1]. The kernels are causal, i.e. both functions are defined only for non-negative time values t, and zero otherwise.

The major difference to the original integrate and fire neuron model is that spike generation is defined in a probabilistic way and depends nonlinearly on the membrane potential. For each point in time t the firing intensity is given by $\phi(V(t)) = \kappa e^{\beta V(t)}$ with fixed parameters $\kappa > 0$ and $\beta > 0$. In the limit of very large β, spike generation will become deterministic with a firing threshold $\vartheta = 0$. The resting potential V_{rest} is negative ($V_{rest} = -1$) to ensure low firing rates at rest. In accordance with Urbanczik and Senn (2009), we will use $\kappa = 0.01$ and $\beta = 5$ for the simulations with the escape noise neuron in Chapter 7.

2.1.2 *Synaptic Learning Rules*

All the previously described neuron models can be idealized as binary classifiers. If the neuron fires a spike (or probably also more in case of the integrate-and-fire neuron with reset), its binary output is defined as $\hat{\ell} = o = 1$ and the corresponding input pattern is assigned to the target class. A silent response ($o = 0$), on the other hand, is regarded as a voting for the irrelevant null class ($\hat{\ell} = 0$). What kind of response is generated for a given input pattern depends strongly on the synaptic weights. Hence, training a neuron to fire only for behaviorally relevant patterns is equivalent to modifying the synaptic weights in the desired direction.

1 This definition slightly deviates from the one stated in the Supplementary information of Urbanczik and Senn (2009) where they did not list an additional summand C_0. However, the actual implementation indeed included the normalization term C_0 (personal communication).

How exactly the weights are modified depends on the specific synaptic learning rule. We will describe three different rules in this section that we will use to train the previously introduced neuron models. Regardless of the specific learning rule, the general learning process works according to the following scheme:

0 Initialize synaptic weights $\boldsymbol{w}$ of the neuron. Choose a scalar learning rate η that determines the speed of learning.

1 Present an input pattern x to the neuron, generate a binary output o based on the current weights $\boldsymbol{w}$.

2 Compare the decision $\hat{\ell} = o$ with the pattern label ℓ. If they agree, go back to 1 wit a new input pattern. Otherwise go to 3.

3 Perform a weight update step according to the neuron-specific synaptic learning rule: Generate weight changes $\Delta\boldsymbol{w}$ and obtain new weights $\boldsymbol{w} = \boldsymbol{w} + \eta\Delta\boldsymbol{w}$. Then go back to 1 with a new input pattern.

Iterate this scheme until all patterns are classified correctly or a number of maximal iterations is reached.

The Perceptron Learning Rule

One of the most famous learning rules to modify synaptic weights of the McCulloch-Pitts neuron is the perceptron learning rule (Rosenblatt 1962). Its purpose is to adjust the synaptic weights after repeated presentations of vectorial input patterns $\boldsymbol{x}$ such that the weighted sum of inputs and weights exceeds the firing threshold if and only if the pattern belongs to the positive target class ($\ell = 1$). Specifically, we demand $\boldsymbol{w} \cdot \boldsymbol{x}$ to have a positive sign for patterns from the target class, and a negative sign otherwise. Whereas throughout the thesis we will denote null class affiliation by $\ell = 0$, for the recapitulation of the original perceptron rule here we follow the notation of Hertz et al. (1991) with $\ell \in \{-1, 1\}$ to offer a direct adaptation of the common learning rule. Whenever the McCulloch-Pitts neuron generates an erroneous response to an input pattern $\boldsymbol{x}$, the perceptron learning rule adjusts each individual weight w_i by

$$\Delta w_i = \ell x_i$$

This learning rule guarantees that the field of the McCulloch-Pitts neuron with the same pattern $\boldsymbol{x}$ after learning will have increased if $\ell = 1$ since $(\boldsymbol{w} + \eta\Delta\boldsymbol{w}) \cdot \boldsymbol{x} = \boldsymbol{w} \cdot \boldsymbol{x} + \eta||\boldsymbol{x}||^2 > \boldsymbol{w} \cdot \boldsymbol{x}$ and vice versa. As described above, the learnable threshold ϑ is encoded in an additional weight w_0 that receives a constant input x_0 for all patterns. With this notation, the threshold can be updated in the same way as the remaining synaptic efficacies. If the McCulloch-Pitts neuron is trained according to this perceptron learning rule, one usually refers to the whole instance as the *perceptron* model.

If a given binary classification task is solvable, it is proven that the perceptron learning rule will converge to a solution within a finite number of steps (for a proof, see for example Dayan and Abbott 2001, Hertz et al. 1991). This is the case for every

problem that is *linearly separable* in the sense that the n-dimensional input hyperspace can be separated by an $n-1$ dimensional hyperplane into two half-spaces with all observations from the null class in the one half-space and all observations from the target class in the other half-space. Whereas in higher dimensions many classification problems are indeed linearly separable, for two dimensions the classical XOR-problem is a common non-separable example with an empty solution space for the perceptron (see Section 6.2).

The Tempotron Learning Rule

The mean firing rate of a neuron is the feature in an output spike train that gains the most attention by researchers and is probably also used by the majority of sensory neurons to encode information. However, several studies in the last decades have shown strong evidence that information about different sensory cues is carried by the precise timing of spikes (Johansson and Birznieks 2004, Singer 1999, Gollisch and Meister 2008). Taking temporal structure into account offers a lot more flexibility in storing different sensory cues (Thorpe et al. 2001). However, it had been not clear for many years how a neuron could utilize temporal information from incoming spike trains. One learning rule that actually is able to decode information from spatiotemporal patterns, is the tempotron learning rule (Gütig and Sompolinsky 2006) that is introduced for deterministic integrate-and-fire neurons. We will base the major part of the simulations on this biologically plausible *tempotron* model, that is again defined as the integrate-and-fire neuron in combination with the tempotron learning rule.

When we interpret this integrate-and-fire neuron as a sensory feature detector, we expect it to fire a spike solely in response to behaviorally relevant inputs. In the binary classification problems we consider in this thesis, the binary spike or no-spike response directly defines the decision for an input pattern. Consequently, we obtain a false positive if the neuron fires an output spike for an irrelevant null input spike pattern and a miss if the membrane potential remains subthreshold during the whole presentation time of a target input spike pattern. In both cases, an error can be traced back directly to the maximum voltage $V_{max} = V(t_{max})$ across the stimulus time. One possible cost function that translates this relation into a continuous number is given by

$$E_{\pm} = \pm(\vartheta - V_{max})\Theta\left(\pm(\vartheta - V_{max})\right) \tag{2.5}$$

with the plus and minus sign for target and null patterns, respectively. The function Θ denotes the heavyside function that takes the value of 1 if the argument is nonnegative (and zero otherwise) and thereby guarantees zero cost for correctly classified patterns. The tempotron learning rule minimizes this cost function by using a gradient descent approach, where each weight w_i is updated by the negative gradient of E with respect to w_i. With the dependencies from formula (2.2) and (2.3), the (negative) derivative of equation (2.5) can be computed under the use of the chain rule as

$$\Delta w_i = -\frac{dE_{\pm}}{dw_i} = \pm \sum_{t_i < t_{max}} K(t_{max} - t_i) \pm \frac{\partial V(t_{max})}{\partial t_{max}} \frac{dt_{max}}{dw_i} = \pm \sum_{t_i < t_{max}} K(t_{max} - t_i)$$

for error trials. Here, the second component of the sum vanishes since t_{max} is defined as the time point at which the maximum of the membrane potential was reached. Hence, the slope of the voltage trace is zero at that point.

Based on this learning rule, the tempotron can not only classify spatiotemporal spike patterns that contain information in their spike rates, but can also reliably distinguish patterns that implemented pure latency or synchrony coding (Gütig and Sompolinsky 2006). Its usefulness in modeling complex behavior of retinal ganglion cells (Gütig et al. 2013) has been demonstrated as well its good performance in speech processing with a slightly more complicated conductance-based neuron model (Gütig and Sompolinsky 2009).

Eligibility-trace based Reinforcement Learning

For the stochastically spiking escape noise neuron, a learning rule has been developed that is based on a *reinforcement* signal that rewards correct decisions of the neuron and punishes erroneous decisions. Reinforcement learning rules in contrast to supervised signals are often not directed. In the learning rule suggested by Xie and Seung (2004) and Pfister et al. (2006) for example, learning aims at reducing the probability of observing exactly the same output spike train for the same input pattern again if the reward is negative. According to Pfister et al. (2006), the log-likelihood for a specific output spike train Y up to time t (denoted by Y_t) is given by

$$L(Y_t|X) = \sum_{s \in Y_t} \log \phi(V(s)) - \int_0^t \phi(V(s))ds.$$

Here, the variable X denotes the total input spike train as a combination of all previously afferent input spike trains X_i.

The derivatives of this log-likelihood with respect to the synaptic weights w_i will provide the basic quantity for the weight updates:

$$\frac{d}{dt}\frac{\partial}{\partial w_i}L(Y_t|X) = \beta PSP_i(t)\left(\sum_{s \in Y_t}\delta(t-s) - \phi(V(t))\right) \tag{2.6}$$

with $PSP_i(t) = \sum_{s \in X_i} \epsilon(t-s)$ being the contributions of the postsynaptic potentials of synapse i and $\delta(t)$ Dirac's delta function. A direct integration over the stimulus time would provide the direct update of the individual weights. However, since the exact stimulus time is not known, Urbanczik and Senn (2009) suggest to use a low-pass filtered version of this time derivative instead:

$$\tau_M \frac{dE_i}{dt} = -E_i(t) + \frac{d}{dt}\frac{\partial}{\partial w_i}L(Y_t|X). \tag{2.7}$$

The solution of this ordinary differential equation $E_i(t)$ is called the eligibility trace of synapse i and its negative values at time T will be used as the weight updates for each synapse:

$$\Delta w_i = -E_i(T)$$

For explicit implementations and problems arising from specific versions, see Section 7.2.1.

In order to stay in a comparable range to the original model from Urbanczik and Senn (2009), we use the same parameters for the simulations in this thesis: The membrane constants were set to $\tau_m = 10$ ms and $\tau_s = 1.4$ ms. The resting membrane potential is $V_{rest} = -1$, the parameters for the firing intensity $k = 0.01$ and $\beta = 5$. Low-pass filtering is performed on the same time scale as the stimulus length $T = \tau_M = 500$ ms.

2.2 POPULATION APPROACHES FOR COMPARISON

The purpose of this thesis is to propose a biologically plausible but still effective population mechanism that utilizes structure in the input data to achieve specialization within the population and thereby improve classification problems on feature detection tasks. While we will dedicate the whole Chapter 3 to the proposed algorithm, here we would like to briefly describe a few population mechanisms that we will use for comparison. These approaches range from very simple unselective training as lower performance limit to state-of-the-art methods and a direct benchmark training as upper limit.

For a population size of $m = 1$ and according parameters, all these population methods reduce to the previously described training of a single neuron. Since the single neuron as well as the population of individual neurons can be regarded as binary classifiers, we will also depict the classification performance of a single neuron in most of the simulations. Intuitively, we expect populations to outperform a single neuron, so that improving the single neuron error should be regarded as a minimal requirement for a reasonable population learning mechanism.

For all binary classification problems that we will investigate in the following studies, we assume to have a population of arbitrary neurons with binary output $o_j \in \{0, 1\}$ for neuron j[2]. Consequently, the null class is encoded by $\ell = 0$, target patterns are labeled as $\ell = 1$.

2 The perceptron outputs of $\{-1, 1\}$ are transformed to $\{0, 1\}$ accordingly

2.2.1 *Trainall - Unselective Training based on Population Voting*

As also for the population algorithm we will propose in the next chapter, for the unselective training we assume an equally weighted population decision of the form

$$\hat{\ell} = \Theta\left(\sum_{j=1}^{n} o_j - d\right)$$

with the heavyside function Θ. The parameter d is defined as the population *decision threshold*. In the specific case of $d = m/2$, the decision corresponds to the common majority voting of population members.

Generally, the population causes a classification error if $\hat{\ell}$ does not agree with the label ℓ for a specific input pattern. In this case, the naive way to change the population's response is to train all neurons in the population that contributed to the erroneous decision. The erroneous decision is easily defined for binary neurons and a binary population response - if the pattern belongs to the target class and the population response is zero, all silent neurons behave erroneously and hence their weights are updated according to the neuron-inherent synaptic learning rule. Similarly, for false positive population responses, all spiking neurons undergo an LTD step. In the following, we denote this kind of naive population learning as *trainall* since in case of a population error we unselectively update the weights of all erroneously behaving neurons.

2.2.2 *Mixture of Experts Training (for Perceptrons)*

In 1991, Jacobs and colleagues (Jacobs et al. 1991) presented a new approach to combine a set of classifiers in a parallel competing way. The idea is a similar one that also motivates this thesis: If the classification problem is too difficult to be solved by one single neuron, we divide the feature space into a fixed number of regions and train a single expert on each. With this, the neurons decompose the overall problem into simpler subproblems and provide a modular solution. This idea is inspired by the mathematical theorem of total probability where the overall probability $P(A)$ is decomposed into a sum of probabilities of the sub events $P(B_j)$ times the conditional probabilities $P(A|B_j)$. Jacobs et al. (1991) transfered this mathematical law to the following network architecture (see also Figure 2.1): In addition to m expert neurons with n inputs (we consider McCulloch-Pitts neurons in the following) they include a gating neuron instance of the same size that also receives the same input. Each of the m gating neuron assigns a weight g_j to the outputs of the corresponding jth expert such that the current input pattern is gated to the neuron that is specialized on the corresponding feature space region. With this, the expert neurons represent the conditional probabilities and the gating neurons the probabilities that the pattern belongs to the corresponding subregion. Hence, the total probability that the pattern belongs to the target class is approximated by the summed product of the outputs. Conse-

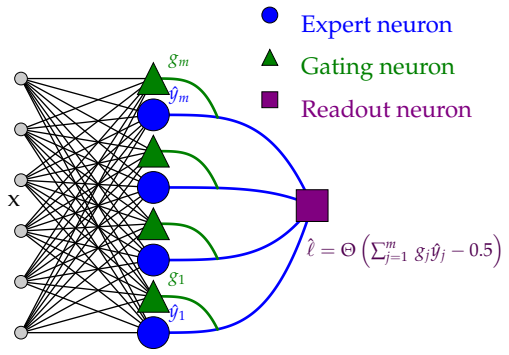

Figure 2.1.: MIXTURE OF EXPERTS ARCHITECTURE. A single layer of gating (green) and expert neurons (blue) receives common input $\boldsymbol{x}$ (gray). Every gating neuron provides a relative weighting g_j to the expert's outputs $\hat{y}_j$, the population output is calculated based on the summed product of g_j and y_j.

quently, the binary readout of the network is given by a thresholded version of this probability:

$$\hat{\ell}(\boldsymbol{x}) = \Theta\left(\sum_{j=1}^{m} g_j(\boldsymbol{x})\hat{y}_j(\boldsymbol{x}) - 0.5\right).$$

Here, $\hat{y}_j$ denotes the output of the jth expert. In case of a binary perceptron classification this should not equal the binary decision itself, but the result of a suitable activation function of the perceptron's field. We will use the sigmoidal function $\hat{y}_j(\boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{w_j}\cdot\boldsymbol{x})}$ as an activation function, with $\boldsymbol{w_j}$ being the weight vector of the jth McCulloch-Pitts neuron, including an offset term for the variable threshold as introduced in Section 2.1.2. Like this, the expert neuron's output agrees with the probability that the pattern belongs to the target class, given that it falls to its specific expert region. To ensure that also the gating neurons' outputs g_j can be regarded as the probability of input $\boldsymbol{x}$ being attributed to expert j, we generate them via a soft-max function based on the gating neuron's fields:

$$g_j = \frac{\exp(\boldsymbol{v_j}\cdot\boldsymbol{x})}{\sum_{k=1}^{m}\exp(\boldsymbol{v_k}\cdot\boldsymbol{x})}$$

Herein, the vector $\boldsymbol{v_j}$ denotes the $n+1$ dimensional weights of the gating network for the expert j.

Generation of such a complex division of labor cannot be achieved by simply training the McCulloch-Pitts neurons unselectively according to the perceptron rule. Rather, we require a learning rule that is explicitly designed for this complex architecture. Specifically, the weights of the experts and the gating neuron are adjusted during learning such that the negative log likelihood function of the mixture model as the objective error function is minimized:

$$E = -ln\sum_{j=1}^{m} g_j(x)\Phi_j(\ell|\boldsymbol{x})$$

where $\Phi_j(\ell|\boldsymbol{x})$ denotes the conditional probability of the label being ℓ given the input pattern $\boldsymbol{x}$, dependent on the individual output of the jth expert (Moerland 1997). For

a binary classification problem with labels $\ell \in \{0, 1\}$, the expert's conditional density is given by the binomial distribution

$$\Phi_j(\ell|\boldsymbol{x}) = (\hat{y}_j(\boldsymbol{x}))^{\ell}(1 - \hat{y}_j(\boldsymbol{x}))^{(1-\ell)}.$$

With this density for the simple perceptron model used here, the partial derivatives of the error function are analytically deducible and hence give rise to the iterative updates of the expert's and the gating neuron's weights:

$$\Delta \boldsymbol{w_j} = \pi_j(\hat{y}_j - \ell)\boldsymbol{x}$$

and

$$\Delta \boldsymbol{v_j} = (g_j - \pi_j)\boldsymbol{x}.$$

Here, π_j is a kind of posterior distribution for the expert j, given by

$$\pi_j = \frac{g_j \hat{y}_j^{\ell}(1 - \hat{y}_j)^{(1-\ell)}}{\sum_{k=1}^{m} g_k \hat{y}_k^{\ell}(1 - \hat{y}_k)^{(1-\ell)}}$$

Consistent with the algorithmic procedure explicitly stated for example by Tang et al. (2002), we modify the weights of gating and expert neurons according to the derived adjustment rules after each input pattern, regardless of whether the binary population response was correct.

This basic variant of the mixture of experts can be expanded to hierarchical structures of expert and gating layers, yielding a powerful method for complex classification problems (Yuksel et al. 2012). However, we are interested in biologically relatively simple networks and hence restrict our investigations to the presented basic variant. Since the gating and expert neurons rely on simple scalar product calculations, a direct translation of the population approach to more complex neuron models such as integrate-and-fire neurons remains challenging. Hence, we will compare the basic mixture of experts approach to the proposed Tagging algorithm only for perceptrons as base learners in Chapter 6.

2.2.3 *Attenuated Learning based on Population Reliability*

Originally defined for neurons with stochastically fluctuating threshold (Pfister et al. 2006, see Section 2.1.1), Urbanczik and Senn (2009) introduced an attenuated population training based on the reliability of the population decision. Here, the population decides by use of the classical majority voting, so $\hat{\ell} = \Theta\left(\sum_{j=1}^{m} o_j - m/2\right)$ for the individual spike or no-spike outcomes o_j. To ensure a clear majority, the number of neurons in the population is chosen as an odd number in all simulations where we consider population learning based on majority voting.

The escape noise neurons in the original population from Urbanczik and Senn (2009) are trained by a reinforcement learning approach. Specifically, their individ-

ual weight updates rely on a synapse-specific eligibility trace $E_i(t)$ (see Section 2.1.1) and do not explicitly differ for false positives and misses: $\Delta w_i = -E_i(T)$ for each input synapse $i = 1, \ldots, n$.

In the population, each neuron j receives an individual reward signal r_j. If both, the individual and the population decision based on majority voting, are incorrect (yielding specifically $r_j = -1$), the neuron updates its weights according to the original learning rule as in the independent neuron case. However, for a correct population decision, the weight updates are attenuated as:

$$\Delta\tilde{w}_{j,i} = a(r_j - 1)E_{j,i}(T)$$

Here, the factor a implements a weighting based on the number of correctly behaving neurons. It is defined as $a = \exp(-P^2/m)$ with P being the number of correct neurons, if the population decision is correct (and $a = 1$ otherwise as stated above). With this weighting, the learning gets less strong the more neurons are already correct on that trial. Consequently, attenuated training based on the number of correct neurons provides some security-margin of a correct population decision for repeated presentations of similar input patterns.

Even though defined for reinforcement learning based neurons with stochastically fluctuating threshold, we can principally also apply the concept of attenuated learning to deterministically spiking neurons in a supervised learning framework. Substituting $-E_{j,i}(T)$ by the general individual weight update $\Delta w_{j,i}$ yields

$$\Delta\tilde{w}_{j,i} = ae_j\Delta w_{j,i}$$

with e_j as individual error for neuron j in the supervised learning notation, corresponding to $e_j = 1 - r_j$. We will use this population learning adaptation for tempotrons in the main Chapter 4 and investigate the original variant in Chapter 7.

2.2.4 *Benchmark: Direct Training*

For some of the following simulation studies, we assume that the behaviorally relevant target class is composed of a discrete number of subclasses. In real classification problems, the subclass identity of a sensory input is not resolved by the labels and hence not accessible to the neurons. In this thesis we use the knowledge of the subclass identity only to evaluate the neurons' ability to specialize. However, in order to relate the classification performance of the Tagging algorithm and the previous population approaches to a benchmark, we can explicitly guide the training process of the neurons in the population to the intuitively best matching. For this so called *direct training* we assign certain subclasses to specific neurons before initializing the learning process and thereby explicitly induce specialization right from the start. The population decision as well as the learning in case of population false positives remains the same as for the Trainall algorithm. Specifically, we predict the target class

if at least d neurons fire and update weights of all erroneously firing neurons at false positives. However, each time the population misses a specific target, only the neurons that are responsible for the corresponding subclass according to this matching undergo learning. For the sake of simplicity, we shift the specialist subclasses by one for each neuron. So specifically, for a strategy S, we assign all patterns from subclass $j, \ldots, j + S$ to neuron j; i.e. whenever a pattern from the set $j, \ldots, j + S$ is missed by the population, we train neuron j on that pattern (unless it has already fired a spike).

2.3 INPUT PATTERNS

In order to evaluate the ability of the proposed population algorithm to uncover structure in the data to achieve higher classification performance, we need to generate sensory input patterns with known ground truth. In many applications, spike trains are modeled as poisson processes, for simplicity ignoring for example the refractory period after output spikes (Rieke et al. 1999). We will also follow this approach and base the input patterns on poisson spike trains that originate either from discrete subclasses or continuously morphed templates to model classification problems with discrete and continuous structure, respectively. Additionally to the definition of these poisson input patterns, in this section we will describe a mechanism to transform recorded auditory speech signals into spike trains based on basic properties of neurons in auditory processing stages required for spoken digit recognition in Chapter 5.

2.3.1 *Discrete Embedded Poisson Patterns*

To mimic sensory detection tasks where the target class is composed of different substructures - like two acoustically different vocalizations of 'harmonic arches' and 'warbles' that both signal the discovery of rare, high-quality food (Gifford et al. 2005)- we model sensory inputs as embedded poisson spike patterns that are noisy realizations of a discrete set of poisson templates.

Individual spike patterns were generated from a $T = 0.5$ s long poisson process with rate 2 Hz for each input synapse independently. Specifically, for each afferent, we drew a poisson number of spikes with expectation value 1 and distributed occurrence times for each realized spike uniformly within the 0.5 s time window. For a null trial, these presynaptic activity patterns were drawn randomly and independently for each presented pattern. For the target class, however, we based the presented patterns on k poisson templates that we had generated at the beginning and held fixed throughout the simulation. Each time a target pattern was presented to the neuronal population, one of these templates was chosen with equal probability. To control the difficulty of the problem, we added three sources of noise. Firstly, we temporally embedded the pattern, regardless of target or null snippet, in a 1.5 s poisson pattern background of the same input firing rate (2 Hz) at uniformly distributed random starting times. Furthermore, we induced noise by adding Gaussian temporal jitter with zero mean and

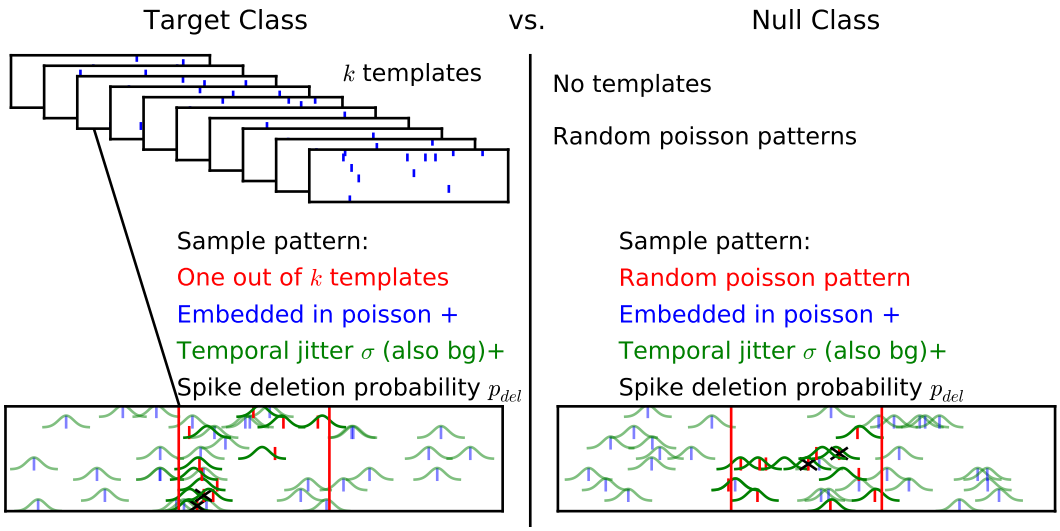

Figure 2.2.: DISCRETE EMBEDDED POISSON PATTERN CLASSIFICATION. Each input pattern from a target class is generated from one of k poisson pattern templates, whereas patterns from the null class are drawn without any template structure. Regardless of its label, the poisson snippet is embedded in random poisson background and spikes are shifted with Gaussian noise of standard deviation σ. Additionally, within the $T = 500ms$ long snippet, spikes are deleted with spike deletion probability p.

standard deviation σ to the input spikes. Additionally, we deleted input spikes with a probability p_{del}, modeling the loss of spikes due to diverging neurotransmitters in the synaptic cleft. Whereas the temporal jitter also affected the background spikes, we decided to apply the deletion probability only to spikes of the 0.5 s snippet to allow for different deletion probabilities in different templates arriving at the same synapse. In order to ensure the same statistics of target and null patterns, these noise operations were applied equally to both classes of spike trains. The procedure is visualized in Figure 2.2. Unless stated otherwise, null and target patterns were drawn with the same probability 0.5.

2.3.2 *Continuous Poisson Patterns*

For the continuous input pattern classification, we constructed spike patterns of approximately the same statistics as in the discrete embedded poisson pattern setting to ensure comparability. However, patterns within the target class depended on a continuous parameter with a procedure already established in Gütig (2013) and Gütig (2014). To construct the target class, we drew a poisson distributed number (with expectation λ) of spike pattern templates for each input synapse together with a uniform random continuous template location between zero and one. Each template was again a poisson pattern of length $T = 0.5$ s with spiking rate $r_\alpha = 2$ Hz. Input spike patterns from the target class were constructed as a warped version between two adjacent spike pattern templates depending on a warping value α ranging from zero to one (with circular boundary conditions). Specifically, every spike time in one template

was associated to a specific spike time in the template with adjacent location (or both adjacent templates if the number of templates is larger than two). If the number of spikes agreed, a direct pairing existed and every intermediate realization was a linear interpolation of the spike times of the paired template times. Otherwise, phantom spike times were drawn for the template with less spikes to allow for a one-to-one mapping. A temporally interpolated spike that only existed in one of the templates was added to the input pattern if the α value was close to the location of the pattern that actually contained the additional spike.
With the above construction, the expected number of templates per synapse λ is negatively correlated to the similarity of close values - if many templates exist, the warping between them has to be very fast and hence even close α values are associated with very distinct spike patterns. We used an expectation value of $\lambda = 15$ for the number of poisson templates. In analogy to the discrete patterns in the previous section, the null class consisted of random poisson spike patterns with the same statistics without any internal template structure. Again, regardless of the class label, the individual patterns were embedded in a background poisson process of the same input rate and additional noise was induced by added Gaussian temporal jitter of standard deviation σ and spike deletion probability p_{del}.

2.3.3 *Auditory Front End*

The speech processing analysis in Chapter 5 is based on the TI46 data base, available via `http://www.ldc.upenn.edu/`. This data set consists of wav files of isolated digits performed by 16 speakers (8 male, 8 female), each speaking 26 utterances of the ten digits. In order to transform the sound signals to input spike trains, we utilized the auditory front end introduced by Gütig and Sompolinsky (2009). Herein, spike patterns were generated by offset- and onset encoders that detected the occurrences of elementary spectrotemporal events from the normalized sound signals. For this purpose, the input layer to the tempotrons formed a 2-dimensional tonotopy-intensity map. Each of its afferents generates spikes by performing either an onset or an offset threshold operation on the power of the acoustic signal in a given frequency band. Whereas an onset afferent elicits a spike whenever the log single power crossed its specific threshold level from below, offset afferents encode the occurrences of downward crossings. Each onset or offset afferent was assigned one of the $N_f = 16$ different center frequencies and one of 15 different intensity thresholds, that were set relative to the maximum signal over time to $\theta_j = j/16$, $j = 1, \ldots, 15$. Additionally, one input afferent for each center frequency was assigned to encode the timing when the maximum log power was reached (corresponding to $j = 16$ for on- and offset detectors), yielding 496 ($(15 + 15 + 1)$ thresholds $\cdot$ 16 frequencies) afferents in total. For a more detailed description of the spike generating process and further investigations on individual neuron performance based on this auditory front end we refer to Gütig and Sompolinsky (2009).

2.4 MEASURES OF SPECIALIZATION

In order to quantify the observed specialization within the neuronal populations, we utilize three different measures. While the commonly used correlation coefficient captures the similarity in the firing behavior of pairs of neurons, it does not include knowledge about the internal structure of the stimulus. The for neuronal populations established redundancy measure on the other hand provides insights about the information distribution within a population of the specific substructure but is harder to interpret. Since both previous methods do not exactly capture our view on specialization, we introduce a new measure based on the Kullback-Leibler divergence (*KL divergence*) for individual neurons. This measure quantifies how well the underlying stimulus structure determines the firing behavior of the neuron regardless of the population.

2.4.1 *Measuring Pairwise Diversity: Pearson's Correlation Coefficient*

One of the most common measures of diversity between two classifiers is the pearson correlation coefficient. Since responses of binary classifiers only take two different outcomes (correct or wrong), the correlation between the responses of two learners can be simply represented by

$$\rho = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$$

where a, b, c and d are given as in Table 2.1 (Shipp and Kuncheva 2002).

	Neuron 2 correct	Neuron 2 wrong
Neuron 1 correct	a	b
Neuron 1 wrong	c	d
Total	a+b+c+d=1	

Table 2.1.: POSSIBLE RESPONSES FOR PAIRS OF NEURONAL CLASSIFIERS. All values denote fractions and sum up to one.

For the spoken (and handwritten) digit detection, we use exactly this simple measure on the target and null class separately to quantify the diversity of the response of each pair of neurons. As a measure for the whole population, we simply depict the average over all pairwise correlations.

When one is interested in measuring only the trial-to-trial variability for a pair of neurons given different tuning properties to different stimulus categories, one should consider the use of the noise correlation instead of the classical correlation coefficient. As defined by Montijn et al. (2014), noise correlation is calculated as the average pearson correlation coefficient over different presentations of each individual stimulus category. We use the explicit notion of the noise correlation to compute the trial-to-trial variability in a complex classification problem with discrete composed target classes (Figure 4.9). Here, depending on their specialization, neurons can be tuned to totally

different target subclasses or the exact same subclasses. Neurons with similar tuning would naturally show highly positively correlated responses and neurons with antagonistic tuning negative correlated responses, but these realizations would only reflect the tuning properties. To avoid masking of the true trial-to-trial correlation, the use of noise correlations is necessary.

2.4.2 *Measuring Specialization within a Population: Redundancy*

One commonly used measure to quantify how a population of neurons transmits information about a given stimulus (structure) is the redundancy measure. For two neurons already stated by Rieke et al. (1999), the redundancy measure is defined as

$$R = I(X_1; S) + I(X_2; S) - I(X_1, X_2; S)$$

where $I(X_1, X_2; S)$ denotes the Shannon information about the stimulus when both neurons' responses are observed simultaneously. Similarly, $I(X_i; S)$, $i = 1, 2$ represents the information provided by knowing only responses from neuron i. For two general random variables X_i and S the Shannon information (or also mutual information) is defined as

$$I(X_i; S) = \sum_{x_i} \sum_{s} p(x_i, s) \cdot \log_2 \frac{p(x_i, s)}{p(x_i) \cdot p(s)}$$

with the sum of the individual probabilities p over all possible realizations of X_i and S.

Two neurons that provide totally independent information about the stimulus have a redundancy of exactly zero. If $R > 0$, the sum of individual information is larger than the combined information, signifying redundant responses (with maximum of $R = I(X_1; S) = I(X_2; S)$ for identical neurons). Synergy, however, is reflected in negative values of R. When divided by $I(X_1, X_2; S)$, this *normalized* redundancy operates within the range of -1 (maximal synergy) and $+1$ (maximal redundancy).

The redundancy measure can be extended for populations of $m > 2$ neurons to

$$R = \sum_{i=1}^{m} I(X_i; S) - I(X_1, \ldots, X_m; S)$$

as used by Schneidman et al. (2003) and Chechik et al. (2001). It is important to note, however, that this measure does not tell us if possible synergy arises from pairs of neurons or higher order combinations of those. The inverse of this redundancy, the *WholeMinusSum* synergy (Griffith and Koch 2014) hence provides only a lower bound on the synergy existent in a neuronal population. Defining a normalization factor of this extended redundancy measure (that we will call *SumMinusWhole redundancy* in accordance with the nomination of its inverse) is not as straightforward as for two neurons. Whereas Chechik et al. (2001) and Marre et al. (2015) use modified versions to obtain values that are easier to interpret, we stick to the original definition due to

the lack of agreement on normalization constants. The following relation holds regardless of the use of a normalization factor: The smaller the realization of the redundancy measure R is, the less redundant the responses within the population are.

2.4.3 *Measuring Subclass Dependency: Kullback-Leibler Divergence*

We define specialization as the degree to which the given substructure determines the response behavior of the neurons. Since both previously described measures do not explicitly capture this view of specialization, we here propose a new measure to quantify the response dependency of an individual neuron on the subclass structure. We start by looking at specialization for a two subclass problem.

For a given binary classification problem with the inputs belonging to class $i = 1$ or $i = 0$, we define the neuron with binary output o as perfectly specialized on the problem, if the following conditions hold:

$$P_1^*(o = 1|i = 1) = 1$$

and

$$P_0^*(o = 0|i = 0) = 1$$

so the neuron fires if and only if observations from class 1 are presented. In our understanding of specialization, every deviation of the firing probability from each of the two conditional probability densities $p_1^*(o|i = 1)$, $p_0^*(o|i = 0)$ reduces the specialization of the neuron on that given problem by the same amount (regardless of the marginal probabilities of occurrences of each input class). We use the Kullback-Leibler (*KL*) divergence between the empirically obtained and the optimal density to measure this deviation. For general probability densities p and p^*, the KL divergence quantifies the information loss when p is used to approximate p^* and can be calculated as

$$D_{KL}(p^*||p) = \sum_{x \in X} p^*(x) \log \frac{p^*(x)}{p(x)}$$

for every possible event x. Hence the proposed measure for specialization of a neurons firing to a specific binary problem is defined as the sum of the KL divergence of the measured probabilities from this optimal specialization for both input conditions:

$$S_{A,I} = D_{KL}(p_1^*(o|i = 1)||p_1(o|i = 1)) + D_{KL}(p_0^*(o|i = 0)||p_0(o|i = 0))$$

This measure can be simplified with the above definition of $p_1^*(o|i = 1)$ and $p_0^*(o|i = 0)$ to

$$S_{A,I} = -\log P_0(o = 0|i = 0) - \log P_1(o = 1|i = 1)$$

Measuring the specialization of a neuron on the more than two-element template-substructure is a generalization of this measure. We define the specialization $S_{A,T}$ as

the optimal two-class specialization of all potential binary partitions in the template space:

$$S_{A,T} = \min_{I \in BP(T)} S_{A,I},$$

whereas $BP(T)$ denotes all binary partitions of the template space (with $\#BP(T) = k^2$ since each pairing is included twice with shifted labeling). The above definition with the equally weighted contributions of both conditional distributions regardless of the input statistics ensures that a neuron that fires only for one template is regarded as specialized as a neuron firing for two templates with exactly the same probability as neuron one (f.i. rows in a firing matrix $[1,0,0,0]$ and $[1,1,0,0]$ would give the same specialization measure). Hence, a neuron with Kullback-Leibler divergence of zero is in our understanding perfectly specialized on the given substructure. The smaller the KL divergence, the more specialized a neuron is.
In practice, if each template occurs equally often, one does not have to scan through all binary partitions but only to $k-1$ by sorting all conditional firing probabilities $p(o=1|t=\sigma(1)) > \ldots > p(o=1|t=\sigma(k))$. Starting with the binary problem to distinguish $t=\sigma(1)$ from the remaining templates, in each new partition $h+1$ it suffices to successively add $t=\sigma(h+1)$ to the target class of the previous set.

3

THE TAGGING ALGORITHM

We hypothesize that specialization within populations of neurons evolves when initial structural differences are amplified by competition between different neurons. One model that implements this kind of learning is the mixture of experts model as introduced in Section 2.2.2. Here, small initial pattern-preferences of the expert neurons are augmented by providing more responsibility to neurons with higher performance on the corresponding input pattern. The complex expert- and gating neuron structure, however, relates to a complicated neural network equivalent with context-dependent inhibitory connections ('sigma-pi units') (Jacobs 1999) and its implementation is hence biologically challenging. At the other extreme, population approaches like majority voting in combination with unselectively training of all neurons that misclassified a pattern allow for a straightforward biological implementation, but are limited in inducing specialization within the population.

In this chapter, we propose a population approach with an intermediate complexity between mixture of experts and unselective training. The network consists of a layer of structurally equal processing neurons with a common input layer. Population decisions are based on the number of active neurons. We achieve specialization within the population by weakly selective training that relies on a comparison of internal states of the neurons on a given input pattern, following a gradient-like approach on the population error. Since comparisons of internal states among a population are biologically implausible, the final (local) Tagging algorithm only uses locally available variables. Here, exclusively those neurons undergo learning whose individual internal states exceed a plasticity induction threshold.

3.1 INDUCING SPECIALIZATION VIA WEAKLY SELECTIVE TRAINING

Population Architecture

Similar to a number of previously described neuronal ensemble methods (Jacobs et al. 1991, Urbanczik and Senn 2009), the proposed model relies on a simple two layer neural network that consists of a number of processing neurons and a final binary decision stage. All of the m processing neurons in the network receive common input and send each a binary spike or no-spike response to an abstract readout neuron. The readout neuron integrates all spiking responses over a long time window and thresholds the number of firing neurons against a population decision criterion d (for a scheme of the population see Figure 3.1). Specifically, if the number of active neurons is larger or equal to d, the corresponding input pattern is assigned to the

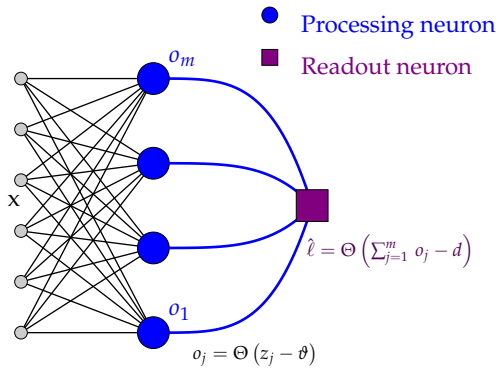

Figure 3.1.: TAGGING ARCHITECTURE. A single layer of processing neurons (blue) receives common input x (black) and sends their individual binary spike or no-spike responses o_j to the abstract readout neuron (purple). The population label is determined by the number of active neurons, thresholded against a decision criterion d.

target class and otherwise to the null class. The population decision $\hat{\ell}$ as the output of the population is hence defined as

$$\hat{\ell} = \Theta\left(\sum_{j=1}^{m} o_j - d\right)$$

with o_j denoting the binary output of the jth processing neuron. Following standard literature (e.g. Hertz et al. 1991), Θ denotes the heavyside function that yields $\Theta(x) = 1$ if the argument x is nonnegative, and zero otherwise. For $d = (m+1)/2$ and an odd number of population members, this readout corresponds to the common majority voting that is used in many ensemble approaches. However, with this additional decision threshold parameter, the present model gains more flexibility. Specifically, for small values of d, an effective solution to the classification problem is provided by neurons that are highly specialized and respond selectively to only small subsets within the target class. On the other hand, larger values of d enable solutions that rely on the coactivation of different neurons. These combined firing solutions can for example result from a specialization on the null class but also from a representation of a target stimulus that is composed of different target features. This second strategy would be consistent with the 'recognition-by-components' theory that postulates that visual objects are identified if a certain number of its components is present in the image (Biederman 1987).

Learning within the Neural Population

The above strategies of specialization within the target or null class facilitate solutions to accordingly structured classification problems. However, it is not obvious how neurons within a population find a synaptic configuration that realizes this solution during a supervised learning task.

Consistent with the idea of the mixture of experts model, we aim to train the neurons in such a way that small initial preferences for specific subsets of input patterns are enhanced and hence specialization is facilitated. By training all neurons in an unselective manner when a population error occurs (*trainall*, see Section 2.2.1), this can be achieved only to a limited degree. Here, all neurons learn the same patterns even

though for a correct detection of a target only a subset of neurons needs to fire for that pattern if d is smaller than the population size. Instead of this unselective training, we propose a population approach that selectively trains only subsets of neurons on each misclassified input pattern. The idea is that neurons that are close to the correct decision should undergo learning, whereas we leave alone neurons whose response was 'very wrong'. Like this, this second kind of neurons can focus on patterns with different statistics instead and initial pattern preferences are expected to be enhanced.

To determine the set of trained neurons, we assume that every neuron fires a spike if an internal state variable z exceeds a firing threshold ϑ. For biologically plausible neuron models, this variable usually agrees with the maximal membrane potential over the time course of the stimulus (V_{max}), but also other internal states as for example the field of the simple perceptron model are possible (as will be shown in Chapter 6). Regardless of its actual representation, we assume that this internal variable z_j depends on synaptic weights w_{ij} for every input synapse i and every neuron j. As defined above, at least d active neurons are needed to evoke a population response in favor of a target pattern. Hence, when sorting all internal states among the population members, the population decision is determined by the d-largest value $z_{(d)}$. With this, we can consider the following population error function

$$E_{\pm} = \pm(\vartheta - z_{(d)})\Theta\left(\pm(\vartheta - z_{(d)})\right).$$

Plus and minus signs refer to errors for target and null patterns, respectively. This the error function is only positive for a misclassified observation. Since the first term measures the distance of the d largest internal state from firing threshold, the error increases the more the d largest internal state deviates from the firing threshold.

In case of a population error, the negative gradient of this error function with respect to the i,j-th weight is given by

$$-\frac{dE}{dw_{ij}} = \pm\frac{dz_{(d)}}{dw_{ij}} = \pm\sum_{k=1}^{m}\frac{\partial z_{(d)}}{\partial z_k}\frac{dz_k}{\partial w_{ij}}.$$

For $k \neq (d)$, small changes of z_k do not affect $z_{(d)}$ as long as they do not exceed $|z_k - z_{(d)}|$. Hence, all summands except for $k = (d)$ vanish and the derivative is given by

$$-\frac{dE_{\pm}}{dw_{ij}} = \pm\frac{\partial z_{(d)}}{\partial w_{ij}}.$$

As a result, the cost function only depends on the derivative of the single neuron with the d largest internal state, since the above expression is zero for $j \neq (d)$. Therefore, to reduce the population error, on a given error trial we select only the neuron with the d-th largest internal state and update its weights according to a neuron-specific learning rule. We expect that this selective training enhances initial preferences in a positive feedback manner: If we assume for example $d = 1$ and a population miss, all neurons have remained silent. The neuron with the d-th largest internal state in this

case is the one that was closest to firing a response. If its weights are increased on that specific trial, it is very likely that, when a similar target pattern is presented to the population, again the same neuron shows the highest internal state. Consequently, after repeated presentation of similar patterns where always this neuron is selected for learning, it starts firing for this subset of target patterns. All remaining neurons, whose responses were far away from the firing threshold, do not undergo any learning steps for this input pattern. Therefore, they are not influenced by the same input subset, but can focus on target patterns with different statistics instead.
Since we assign a binary learning 'tag' to the neuron that is globally selected as responsible among the set of neurons, we call this algorithm *global Tagging* in the following.

As noted above, the presented learning approach enhances initial preferences. However, if a neuron initially does not favor any target pattern over the other population members, its weights remain small during the learning process. Neurons that fall entirely silent are useless in this specific neuronal population as they do not contribute to the classification problem. Assuming that this neuron operates mainly in this population, the presence of permanently silent neurons would be inefficient also from the biological perspective. Indeed, several experimental studies show that neurons maintain a certain firing rate by different kinds of metaplasticity as for example synaptic scaling (Turrigiano 2008, Burrone et al. 2002). In the proposed algorithm, we reactivate chronically silent neurons by a preferential training on missed target patterns. Specifically, if a neuron has not fired a single spike within the last $silent_{lim}$ patterns (in the following simulations, $silent_{lim} = 1000$), it is added to a set of silent neurons. For any missed target pattern, we select the neuron with the largest internal state z among the set of silent neurons to be trained on that target pattern. Only if this set is empty (as in the majority of trials), the global Tagging rule is applied. As soon as a silent neuron fires a spike, it is excluded from the set of silent neurons and its counter is reset.

3.2 THE (LOCAL) TAGGING ALGORITHM

In the global Tagging training that we derived by a gradient-like minimization of the population error in the previous section, we select exactly one neuron that undergoes learning at each error trial. The selection of the responsible neuron is based on a comparison of an internal variable z among all population members. In a biologically plausible neuron model, this variable corresponds to the maximum membrane potential V_{max} over the time course of the stimulus. We will use $z = V_{max}$ for the following argumentation, even though other internal variables are possible. A neuron in the brain does not have access to membrane traces of the neighboring neurons to perform interneural comparisons of the maximum voltage. Additionally, the global superthreshold maximum voltage, that would be used to determine the tagged neuron in case of a population false positive, is only a hypothetical value and never realized. Hence, to meet our goals of a biologically plausible population learning mechanism that agrees with our definition in Section 1.1, we require essential modifications to the above selection procedure of trained neurons.

For false positives we modified the algorithm in the way that we simply adjust the weights of all erroneously firing neurons. With this unselective training on false positive patterns, we need to facilitate specialization within the population by a selection process of trained neurons following population misses. Here, the subthreshold membrane potential is a locally available measure accessible by each individual neuron. Instead of comparing these values among different population members, we threshold each individual V_{max} against a neuron-specific learning threshold v_{train} such that all neurons whose maximum membrane potential exceeds this threshold undergo a learning step for the current missed input pattern. With this modification, the idea of enhancing initial preferences still holds: Neurons that were close to firing and hence close to changing the population response with minimal effort initiate learning to potentiate their responses. Since the selection process of tagged neurons with this modification depends only on a locally available training threshold[1], we refer to this biological plausible algorithm as the (local) Tagging algorithm. We show in Section 4.1 that a fixed value of 0.85ϑ common to all neurons yields clearly specialized firing behavior in a classification task where the target class is composed of discrete subclasses. However, to avoid tuning of this parameter for different classification paradigms, we suggest to adjust the learning threshold during the training process.

Adjusting the Training Threshold during Learning

In the neuron models we consider, LTD and LTP steps are assumed to have equal strength. With this assumption, on the individual neuron basis, learning has converged only if the number of LTD steps agrees with the number of LTP steps. Furthermore, on the population level, the ratio of misses and false positives should match the occurrence probability of target patterns compared to null patterns in a steady state. Consequently, for equally sized input pattern classes, false positives FP and misses MS should be minimized equally in the decision taking population. Following these considerations, we developed an adjustment rule for the learning threshold based on the factor of the above two equilibria. The variable $\chi = \log\left(\frac{FP}{MS}\,\frac{LTP_j}{LTD_j}\right)$ detects all deviations from the combined target equilibrium. Positive values indicate an imbalance in the direction of too many LTP steps, whereas negative values point at the presence of too many LTD steps of the corresponding neuron j. The learning threshold $v_{train,j}$ has a direct influence on the probability to induce an LTP learning step; if χ is skewed towards the positive direction, this shift can be corrected by an increase of $v_{train,j}$, raising the hurdle for LTPs. Similarly, reducing $v_{train,j}$ increases the probability of an LTP step and thereby shifts the variable χ towards the positive direction. Hence, one possibility to maintain a balanced learning is to adjust the learning threshold as

$$v_{train,j} = v_{train,j} + a_\chi \chi \tag{3.1}$$

with a_χ being a suitable scaling factor. In practical simulations, one could evaluate this ratio over a fixed number of presented patterns and adjust the learning threshold accordingly. However, when one of the required four components is (close to) zero,

1 We will use the notations *training threshold* and *learning threshold* equivalently throughout the thesis.

the dynamics become instable. Hence, we modify the learning threshold after each learning epoch (of usually $p = 1000$) patterns only if each component counts at least $c_{min} = 5$. For learning iterations where at least one counter does not yield this minimal number of events, we extend the monitoring window over an additional learning epoch and proceed accordingly until all values reach c_{min}.
Unless otherwise stated, we refer to the (local) Tagging population algorithm as this version where the training thresholds are adjusted based on the target equilibrium of global error rates and local learning counters.

The flexible learning threshold aims at balancing learning dynamics in the population. However, as noted above, to provide a reliable estimate of the statistics, every component needs to appear a certain number of times. Calculation of the target ratio is hence specifically impaired in situations where neurons turn silent. Those neurons have not been recruited for LTP learning for longer time intervals, hindering evaluation of the fraction χ. Therefore, the current algorithm also requires additional mechanisms to reactivate silent neurons. While direct preferred training as used for the global tagging is excluded due to biological implausibility, options involve for example direct scaling of (excitatory) synapses (Turrigiano 2008) or metaplasticity with respect to the learning threshold (Cooper and Bear 2012). As we have an adjusting learning threshold at hand in the proposed Tagging variant, we include the metaplasticity directly in this adjustment procedure. Specifically, we decrease the learning threshold by a constant amount a_{decr} after each input pattern presentation where the corresponding neuron was detected as silent.
In contrast, for the rarely used local Tagging with fixed learning thresholds, we scale all input synapses by a constant factor of $a_{scaling} = 1.00001$. Even though several synapses may be inhibitory, this scaling increases the variance of the membrane trace and hence effectively lowers the firing threshold.
In analogy to the global Tagging, silent neurons are defined as those that have not fired any spike for at least the previous $silent_{lim} = 1000$ patterns.

In the following chapters we investigate to what extent the selective training mechanisms are able to induce specialization within a neuronal population. Whereas our main findings within the Chapters 4 to 5 are based on integrate-and-fire neurons with the tempotron learning rule (see Section 2.1.2), we also demonstrate the applicability of the (gradient-like global) Tagging algorithm to the analytically more tractable perceptron model in Chapter 6 and a neuron model with stochastically fluctuating firing threshold in Chapter 7.

Possible Biological Implementations

Even though the local Tagging algorithm complies with our defined constraints of biological plausibility, it is still a challenge how this population learning might be implemented in the brain. As this thesis focuses on the algorithmic constraints of a plausible population learning mechanism, we only briefly review some neurobiologically inspired ideas and concepts here that might account for a biological implementation.

The idea of a sliding learning threshold can be related to the Bienenstock-Cooper-Munro (*BCM*) theory of synaptic plasticity that goes back to the 1980s (Bienenstock et al. 1982). Here, it is postulated that coincident pre- and postsynaptic activity induces Hebbian-like LTP learning only if the postsynaptic activity exceeds a threshold, and LTD otherwise. Interestingly, this threshold is not fixed but depends on the previous firing history of the postsynaptic cell. Experimental evidence supports the existence of such threshold-induced metaplasticity mechanisms in visual cortex and hippocampus (Abraham et al. 2001, Cooper and Bear 2012) and suggests the inclusion of intracellulary as well as also intercellulary pathways as possible mechanisms for adjustment of the sliding threshold (Hulme et al. 2013). These findings are in line with the proposed adjustment rule based on local plasticity history combined with population error feedback, even though the notion of our training threshold deviates from the BCM theory in terms of its direct relation to supervised learning and its lack to induce LTD.

Also consistent with the BCM theory, neurons have developed chemical mechanisms to maintain a certain firing rate. Returning to a homeostatic firing rate can be realized on many different levels. In the neurobiological literature, direct scaling of (excitatory) synapses (Turrigiano 2008) or metaplasticity with respect to the learning threshold (Cooper and Bear 2012) are discussed as potential mechanisms. The proposed procedure to lower the learning threshold to reactivate silent neurons is consistent with the BCM theory. Citing Cooper and Bear (2012): '[...] weakly active neurons seek to 'turn up the volume' on synaptic input by reducing this threshold'.

4

UNCOVERING STRUCTURE BY SPECIALIZATION

In the previous chapter we introduced the Tagging algorithm, a simple population learning scheme that is based on weakly selective training. In a supervised learning task, following each population error trial only a subset of neurons in the processing layer undergoes learning. Whereas in the global variant of this algorithm, we assign a learning 'tag' to only one neuron whose identity is determined based on an interneural comparison of internal states, the local Tagging algorithm selects a subset of neurons in a biologically plausible manner. Here, whenever the population misses a target pattern, all neurons whose locally available internal variable exceeds a neuron-specific learning threshold, perform an LTP step. Learning as a consequence of erroneously detected null patterns, however, is simply induced for all firing neurons.

Is this weakly selective training sufficient to induce specialization within the population? We study this question in great detail in this chapter for a population of tempotron neurons. The tempotron is based on an integrate-and-fire neuron that integrates spike patterns to generate a time-dependent voltage trace and fires a spike if the maximum of this voltage exceeds a firing threshold ϑ. Learning is induced by the tempotron learning rule that modifies synaptic weights according to their contribution to the maximal membrane potential V_{max} (see Section 2.1.2). To determine the subset of neurons that undergo learning within the Tagging algorithm, we utilize this maximum membrane potential V_{max} as internal state z that is related either to values of other neurons in the population (global Tagging) or to a neuron-specific training threshold v_{train} (local Tagging).
To investigate if specialization emerges when the population is trained with the Tagging algorithms, we consider a specifically tailored classification problem where we know the ground truth. For this, we generate input spike patterns as poisson processes with the target class being composed of different discrete subclasses. The main focus of the following investigation is if the proposed algorithm achieves accurate performance on the binary classification task by uncovering this hidden subclass structure that is not resolved by the binary label of the input patterns.

4.1 UNCOVERING HIDDEN STRUCTURE IN DISCRETE CLASSES

In a study by Gifford et al. (2005), the authors found that neurons in the monkey prefrontal cortex (PFC) could reliably distinguish conspecific calls denoting high quality food from those that denoted low-quality food, even though the high quality-food calls 'harmonic arches' and 'warbles' showed acoustically different structures. This problem can be regarded as binary (high-quality vs low-quality food), with the target class being composed of different discrete subclasses (harmonic arches and warbles). We model this kind of composite binary classification problem by template-based pois-

son input patterns. Specifically, every input pattern from the target class is realized by sampling one out of k poisson pattern templates. Patterns from the irrelevant null class, however, are chosen randomly without any template structure. To control the difficulty of the classification problem, we embedded the poisson patterns in poisson background noise and included two additional noise sources: Individual spikes are deleted independently with deletion probability p_{del} and shifted in time with Gaussian noise with standard deviation σ. The task of the neuronal population is to distinguish target patterns from null patterns, irrespectively of the underlying subclass identity of the corresponding input pattern. However, given the discrete structure within the target class, we hypothesize that classification performance is improved when neurons in the processing layer of the population utilize this structure to generate a population decision. Therefore, we expect to obtain a reflection of this structure in the firing responses of the individual processing neurons after training with the Tagging algorithm.

4.1.1 *One-to-one Specialization*

In the above example, separating high-quality from low-quality food calls could be implemented by the following simple architecture in agreement with the Tagging population: Two processing neurons project to a simplified PFC neuron that assigns a pattern to the high-quality food class if at least one neuron fires (corresponding to $d = 1$). If one processing neuron learned to be activated solely by the presence 'warbles' and the other neuron by 'harmonic arches', the complex target detection task would be divided into two simpler subproblems, facilitating higher population accuracy than when the problem would be tackled as a whole.

In this section we investigate if this one-to-one specialization emerges automatically within a neuronal population. To this end, we consider a complex target class that is - in contrast to the food-call example with two subclasses - composed of $k = 9$ discrete subclasses. Populations of $m = 9$ neurons were trained on the binary supervised classification problem to distinguish this complex target class from a broad null class without internal structure. As in the above example, input patterns were assigned to the behaviorally relevant target class if at least one neuron within the population was active ($d = 1$). We adjusted the difficulty of the classification problem by shifting spikes in each input pattern with Gaussian noise of standard deviation $\sigma = 0.07$s and by deleting them with probability $p_{del} = 0.2$. In each of $lmax = 10000$ learning epochs, we presented $p = 1000$ random patterns from the null or target class with the same probability and updated the weights of the neurons following each population error trial according to the tempotron learning rule.
When we applied this weight update procedure unselectively to every erroneously behaving neuron in the population (called *trainall* in the following), the population classification error saturated at about 22% - which was only slightly superior to a single neuron tackling the same binary classification problem (yielding an error of 24%,

see Figure 4.1, panel (e)). This significant but only minor improvement suggests a lack of emerged specialization within the population if the hypothesis of a major benefit by division of labor is correct. Investigating the firing probabilities for every neuron to every target subclass indeed revealed unselective, diffuse firing with no clear indication of specialization (Figure 4.1, panel (a)).

However, when we trained a population of neurons with the same architecture and population decision, but selectively only performed LTP steps on neurons whose membrane potential exceeded a neuron-specific training threshold as suggested in the Tagging algorithm, we indeed obtained specialization. When we arranged the two dimensional firing probabilities of every neuron and target subclass according to the maximally preferred subclass, the resulting *firing matrix* revealed a clear diagonal structure (Figure 4.1, panel (b)), indicating a one-to-one mapping of neurons and subclasses. These results were also consistent across a large number of simulations, quantified by the sum of the off-diagonal entries as deviations from the diagonal structure. The histograms of these off-diagonal sums were clearly distinct for populations trained with the Tagging and the trainall procedure (Figure 4.1, panel (c) and (d)). This finding emphasizes the role of weakly selective training: Even though the neurons did not receive any direct information about the subclass identity of individual input patterns, specialization emerged since different initial preferences between the neurons were enhanced. The input connections of those neurons that were already close to generating a spiking response to a target pattern were strengthened, whereas other neurons with overall low membrane traces were not distracted by this specific pattern and could focus on learning different aspects of the task.

Consistent with the hypothesis of specialization as a means for accurately solving classification problems, the clear one-to-one mapping that we obtained with the Tagging algorithm is accompanied by a significantly improved classification performance (Figure 4.1, panel (e)). With 1.6% final classification error, the local Tagging algorithm strongly outperformed the unselective trainall procedure as well as a single neuron. In fact, it even performed slightly better than the gradient-like global variant with 1.9% final population error, that utilized a biologically implausible comparison of maximal membrane potentials within the processing neurons to select the single tagged neuron. To relate the results to a benchmark how well a perfectly specialized population performs on this binary classification task, we also trained a population where information about the subclasses was directly provided to the individual neurons during learning. Specifically, whenever patterns from subclass j were missed by the population, we strengthened weights of solely neuron j (*direct training*). With this biologically implausible direct training, the error was reduced to 1.5%, which showed to be a significant improvement over the Tagging results on a two-sample t-test (99% confidence level). However, compared to the impressive error improvement with Tagging over the unselective trainall population, this difference appears of minor relevance.

A recent biologically inspired population approach (Urbanczik and Senn 2009) utilized the number of correctly behaving neurons as a population reliability measure

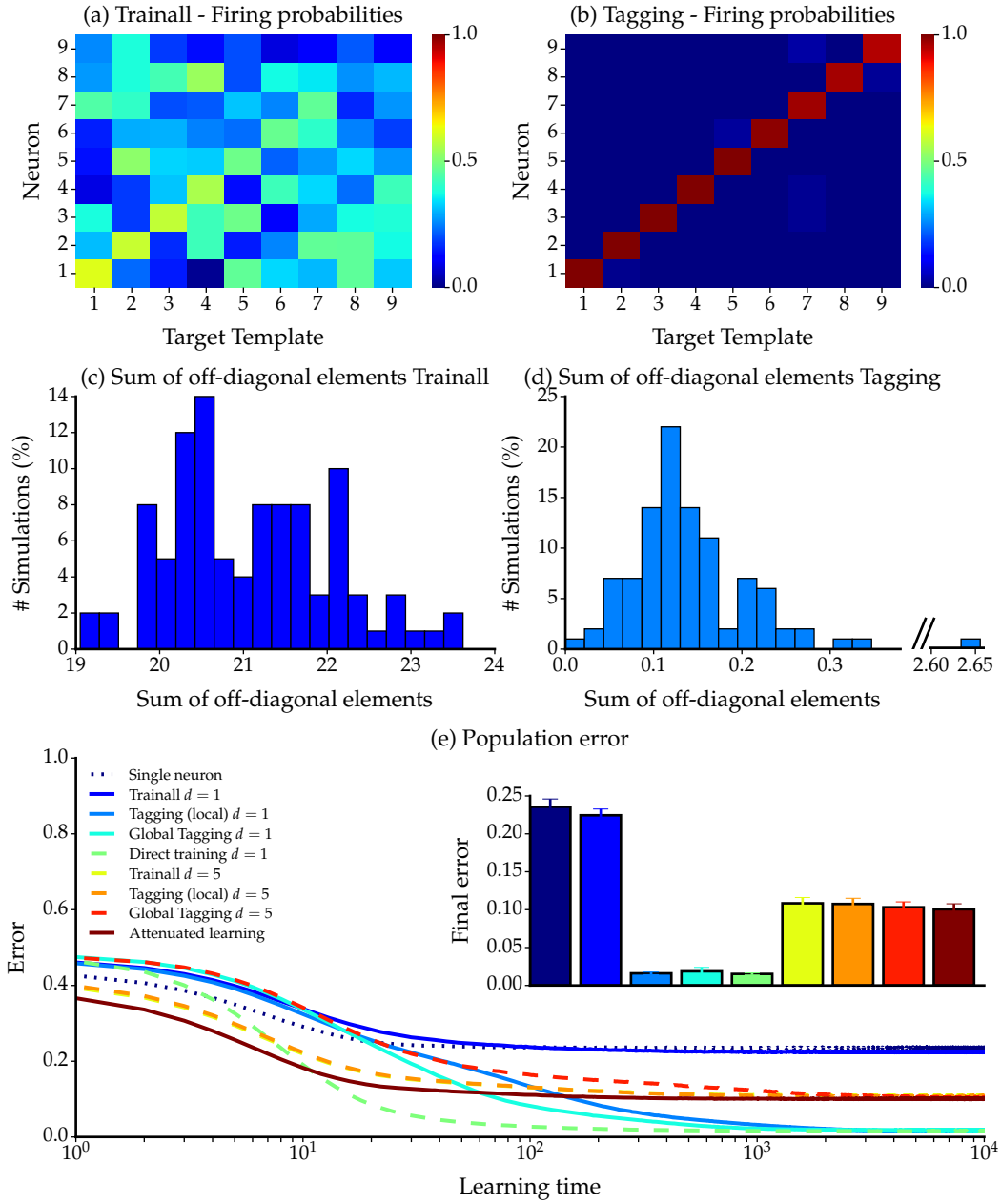

Figure 4.1.: SPECIALIZATION ON SUB-CLASSES. A population of $m = 9$ neurons learns to distinguish $k = 9$ subclasses from poissonian noise. (a) Firing probabilities of each neuron (row) to each target subclass (column) (*firing matrix*) of a typical population trained via unselective trainall training. Neurons fire diffusely for different subclasses. (b) Firing matrix for a population trained with the (local) Tagging algorithm. Every neuron specializes to fire for a single template. Firing matrices are sorted to highlight the diagonal structure. (c) and (d) Histogram of the sum of off-diagonal elements of the sorted firing matrices for all 100 simulations with trainall (c) and Tagging (d). The distributions are clearly distinct, with Tagging results being very close to the diagonal structure. Note the single outlier for Tagging. (e) Population classification error over time for a single neuron and populations of $m = 9$ neurons trained on different population approaches and decision thresholds. Inset panel: Final classification errors for all training variants, averaged over all 100 simulations. Error bars denote the standard deviation (convention throughout the thesis). Except for the trainall and local tagging populations with $d = 5$, all differences are pairwise significant on a 99% confidence level, obtained with a t-test for independent samples.

to attenuate learning. Since its population decision is based on majority voting, we also trained populations of neurons with $d = 5$ for the global and local Tagging algorithms as well as the unselective trainall learning to enable a fair comparison with the attenuated population learning. Consistent with the intuition that the classification paradigm favors specialized solutions, all populations based on majority voting perform significantly worse than the specialist solutions with Tagging and $d = 1$. Error rates ranged from 10% to 11% as opposed to the previously mentioned $1.5\% - 1.9\%$ for selective training with $d = 1$ (Figure 4.1, panel (e)). Within the group of majority voting populations, the attenuated learning by Urbanczik and Senn (2009) yielded best results, followed by global and local Tagging. Again, the unselective training was inferior to the more sophisticated learning methods, albeit not as profound as for $d = 1$. All these performance differences, however, were negligible compared to the clear performance drop of majority voting based mechanisms compared to selective training with $d = 1$.

These results indicate that for majority voting based decisions, neuronal populations do not benefit from the proposed weakly selective training over attenuated training. However, the findings also show that the Tagging algorithm is much more flexible compared to the attenuated learning algorithm from Urbanczik and Senn (2009) and classification performance can be strongly improved in situations where specialist solutions are beneficial.

4.1.2 *Influence of the Adaptive Learning Threshold*

The key component of the Tagging algorithm is that on miss trials only those inactive neurons undergo learning whose maximum membrane potential exceeds a neuron-specific training threshold. The idea is that crossing the training threshold acts as in indicator for how close the neuron was to generating the correct firing response. However, how to actually set this training threshold is not obvious. In order to avoid problem-specific parameter tuning, we suggested heuristic adjustment during learning that is based on a target equilibrium between global population error components and local learning counters (see Section 3.2). If an individual neuron performs too many LTP steps in relation to the number of misses in the population, its individual threshold should be increased to avoid further learning and vice versa. The previous simulations on the classification problem with $m = 9$ neurons distinguishing $k = 9$ target subclasses have been performed with a local Tagging algorithm that implements this kind of learning threshold adaption, with individual thresholds starting from a common value of $v_{train} = 0.8$. All simulated populations yielded low classification errors and clearly specialized neuronal firing responses. Would the results still look qualitatively the same if we had initiated the thresholds with for example $v_{train} = 0.5$ or even kept them fixed throughout the whole simulation?

To investigate how sensitive the population behavior is to the actual choices of different (fixed) learning thresholds, we returned to the previously described classification problem. Firstly, we trained different Tagging populations of $m = 9$ neurons with 20 different fixed, common training thresholds ranging from 0.05 to 0.95. The population error rates that we obtained for 100 different simulations as well as the qualitative behavior of the population strongly depended on v_{train}. If v_{train} was too small (like $v_{train} = 0.3$) in almost all misses all neurons underwent learning, which made the whole learning process equivalent to the trainall procedure (panel (a), Figure 4.2, left inset firing matrix). Too large learning thresholds on the other hand slowed down the learning process since only very few misses led to a learning update of a neuron (right inset matrix). In order to allow for reasonable learning to some degree, additional mechanisms to reactivate silent neurons like synaptic scaling were necessary. However, also with these mechanisms, after $lmax = 10000$ learning epochs we still obtained miss rates close to one for $v_{train} = 0.95$ and did not extend the learning time (yellow curve in panel (a)). Only a fixed threshold of $v_{train} = 0.85$ produced specialization in most simulations. The final population error revealed a clear minimum obtained with this well-chosen learning threshold (see dark blue curve in panel (a)).

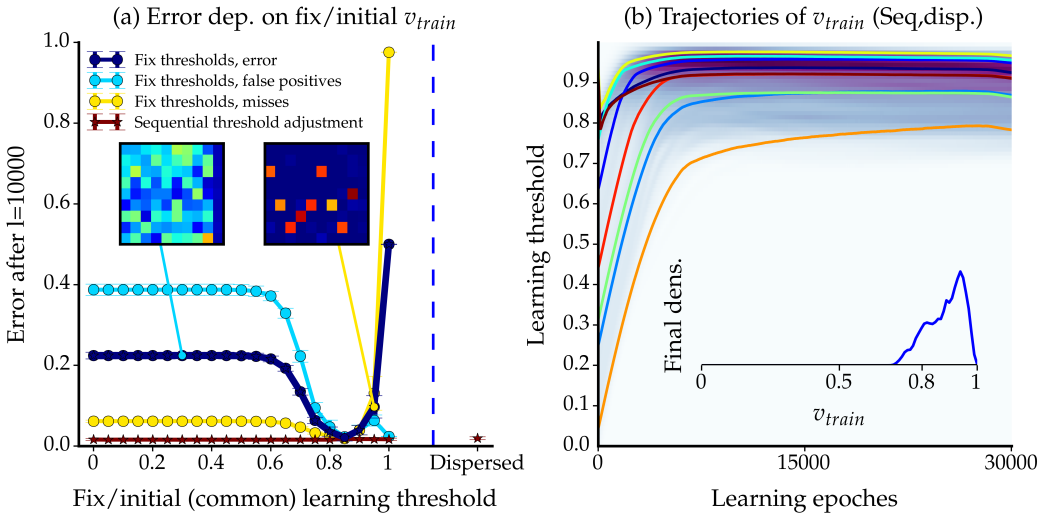

Figure 4.2.: INFLUENCE OF THE LEARNING THRESHOLD. (a) Average population error for local Tagging with fixed thresholds and sequential learning threshold adjustment (dark blue and red curve, respectively). For fix small thresholds, all neurons update their weights on miss trials, yielding a large number of false positives (light blue curve). Too high learning thresholds hinder miss correction totally, on the extreme yielding a miss rate close to one (yellow curve). Only intermediate values of about $v_{train} = 0.85$ evoke well-behaved learning with similar error as for local Tagging in panel (a) of Figure 4.1. By adjusting the learning threshold sequentially, all simulations approach the low error solution, regardless of the initial threshold. Also dispersed initial thresholds within the population do not cause learning problems. (b) Evolution of the learning threshold via sequential adjustment. Colored lines show training threshold evolution for $m = 9$ neurons of a representative Tagging population that was initialized with random different threshold values. The background coloring depicts the density estimated over all neurons from the 100 simulations. Regardless of the starting values, all training thresholds evolve to higher values, with more than 90% of the final values between $v_{train} = 0.76$ and $v_{train} = 0.98$ (inset panel).

In contrast to these results obtained with common fixed learning thresholds, populations that incorporated the automatic adjustment of the learning threshold based on the above explained equilibrium yielded specialized solutions regardless of the initial starting threshold. A uniformly low final classification error accompanied this specialization (Figure 4.2, red curve in panel (a)). Even when the individual neurons within a population were initialized with dispersed randomly chosen learning thresholds, the classification error remained at the same low level. We monitored the individual training thresholds for each population and neuron in this dispersed setting over time and found that regardless of the starting value, 90% of all training thresholds converged to final values within the interval $[0.76, 0.98]$ (Figure 4.2, panel (b)). Due to these very similar results for all tested common or dispersed initial training thresholds, we omitted parallel runs with several starting values in the following simulations. Instead, we initialized all neurons with $v_{start} = 0.8$ and applied the self-adjustment rule for all tempotron-based populations throughout the thesis. Furthermore, all following simulations are run with the biologically plausible local Tagging variant unless stated otherwise.

If synaptic weights of individual neurons are small and hence the membrane trace stays below the learning threshold for all patterns, neurons might fall entirely silent during learning. Here, additional mechanisms are needed to reintroduce these neurons to the learning process. Whereas for the simulations with fixed training thresholds we relied on the common concept of scaling all synapses by a constant factor larger than one, we chose to implement reactivation mechanisms for variable training thresholds also on the basis of v_{train}. Specifically, we decreased the corresponding training threshold by a fixed value every time a silent neuron is detected. To evaluate the influence of this algorithmic component, we monitored the fraction of patterns for which a neuron is defined as silent in the previous simulations (Figure 4.3). Consistent with the idea of the learning threshold, small values of v_{train} led to weight updates for almost all non-firing neurons and thereby hindered that neurons turned silent during the simulation. Therefore, for local Tagging with fixed learning thresholds (red line) the fraction of silent pattern was zero for small thresholds and increased for larger values of v_{train}. For variable learning thresholds, however, the adjustment counteracted this effect and hence lowered the need for an additional re-activation component. In rare cases, when the required statistic to adjust the threshold was unavailable due to too small learning counters, this re-activation was still required. However, even for larger starting thresholds this situation occurred in less than 0.5% of all presented patterns. Based on these low numbers we would conclude that the additional re-activation procedure does not play a major role during learning and does not interfere with the basic algorithmic concepts.

4.2 INTERNAL SPECIALIZATION

In the previous section we showed that the Tagging algorithm is capable of recovering subclass structure and hence improves classification performance in situations where

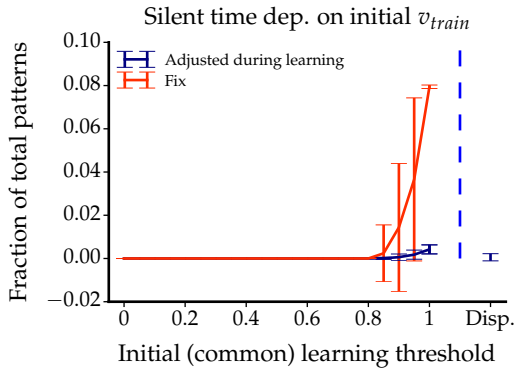

Figure 4.3.: ROLE OF SILENCE. Average fraction of input patterns for which a neuron was detected as silent during Tagging learning. For fixed learning threshold neurons turned silent due to too few LTP steps (red curve). When the learning threshold is adjusted during learning, the adjustment counteracts the silencing (blue curve).

the number of neurons and subclasses agree. However, how does this algorithm perform in more general settings of unequal population- and target class size? For a surplus on sensory classes, a beneficial specialization would be obtained when the number of subclasses are roughly equally split among the neurons in the population (like each neuron firing for two subclasses, etc). However, the inverse situation of an exceeding number of neurons seems structurally more interesting. Here, if target templates differ in their difficulty, one possible solution would be that additional neurons focus on the more complicated subclass and hence share the difficulty across the population. Do the neurons in the Tagging population indeed follow this strategy? And if so, how do they manage to share the difficulty of a single subclass that does not contain any explicit substructure?

4.2.1 *Specialization within a Subclass*

To investigate the question if difficulty is shared equally within the population, we built up a (local) Tagging network with $m = 5$ neurons and $k = 4$ templates, where patterns from one subclass were disturbed by a higher input spike deletion probability ($p_{del} = 0.5$ compared to $p_{del} = 0.2$ for the remaining four subclasses and the null class, with $\sigma = 0.07$s temporal jitter for all patterns). The above considerations suggest that within such a framework, classification performance would be improved if the more difficult subclass was represented by two neurons instead of one. Indeed, in 99 of the 100 simulations, three out of the five neurons specialized to fire for one of the simpler subclasses as in the previous diagonal case, whereas the two remaining neurons together detected the more difficult template (Figure 4.4). Since the first three subclasses were disturbed by a smaller spike deletion probability, the corresponding neurons detected these patterns with more than 90% accuracy. For the more difficult subclass, however, the firing probabilities ranged from 43% to 88% for the two specialist neurons in all simulations (representative example firing matrix in Figure 4.4, panel (a)), reflecting the higher noise regime for this subclass.

How is the specialization of two neurons on the same subclass realized? If responses of the two neurons learning the same target pattern were highly correlated, an improvement of the classification performance would be possible only by a small amount. However, the final error obtained with these populations was significantly

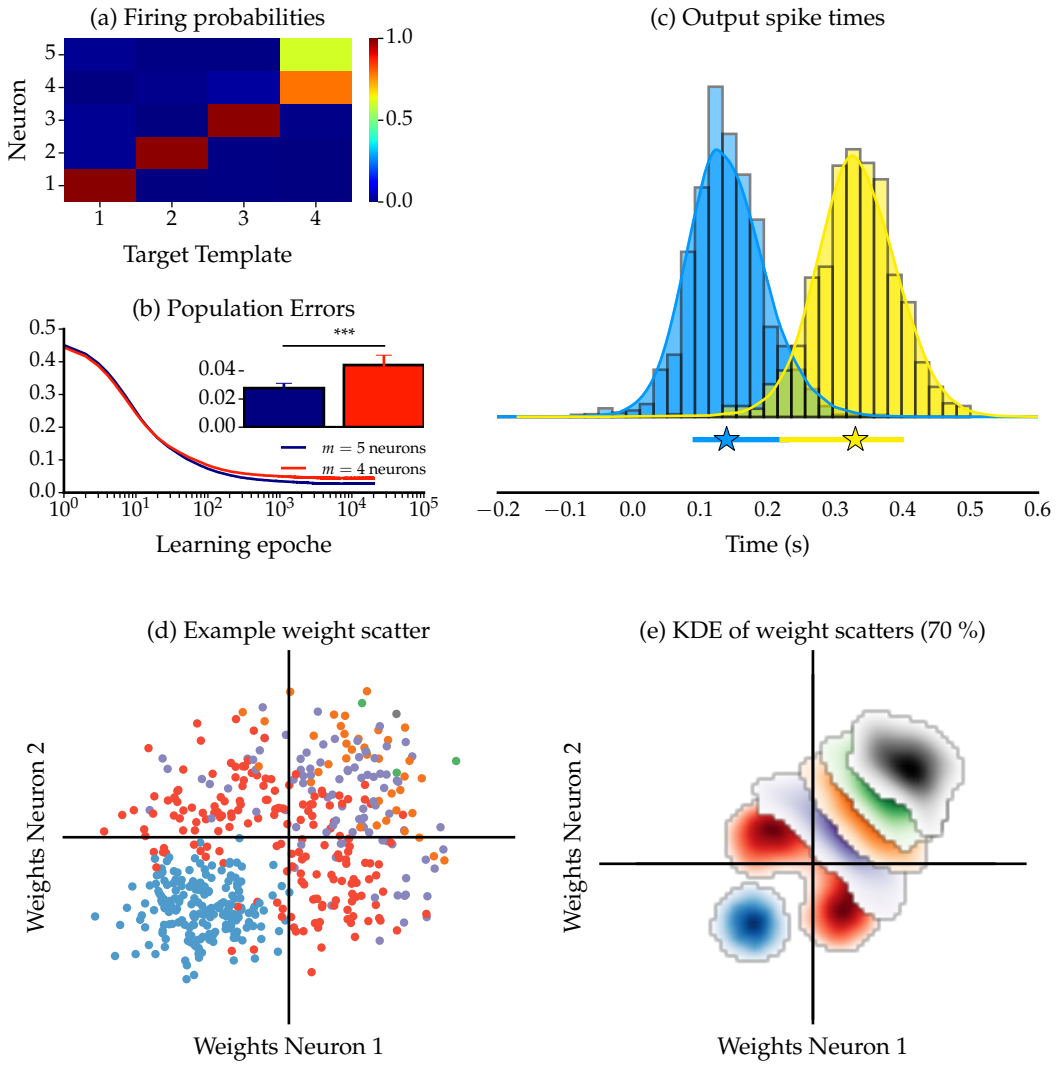

Figure 4.4.: INTERNAL SPECIALIZATION ON A DIFFICULT SUBCLASS. A Tagging population of $m = 5$ neurons learns to distinguish $k = 4$ subclasses from noise where the fourth subclass suffers from higher spike deletion probability. (a) Sorted firing matrix of a representative Tagging simulation. Three neurons show a one-to-one specialization on the templates, whereas the two remaining neurons shared the more difficult template with firing probabilities between 60% and 80%. (b) Error of the Tagging population with $m = 5$ neurons compared to one with only $m = 4$ neurons. The additional neuron provides significant performance improvement. (c) Output spike time histograms for the representative example of panel (a). Spike timings overlap only slightly, neurons specialize on temporally distinct features of the input pattern. Bars on the bottom show the minimum, mean (star) and maximum of the average latencies of first and second specialist among all simulations. (d) Synaptic weight scatter plot for all weights of the firstly firing specialist neuron (x-axis) and secondly firing specialist neuron (y-axis) of the representative example from (a). Colors depict the number of input spikes for the target template on the corresponding synapse (in the order blue (0 spikes), red, purple, orange, green. The more input spikes a synapse has, the larger its weight for both neurons. Synaptic weights within the same input-spike-group are negatively correlated. (e) Two-dimensional kernel density estimators of the synaptic weight pairings for all 100 well-behaving simulations, evaluated separately for each number of input spikes. Within the borders of the individual blobs lie 70% of all corresponding synaptic pairings. Color code is the same as in (d) with an additional cluster in black for five or more spikes (not present in the representative example in (d)).

reduced compared to that of populations with only $m = 4$ neurons, suggesting a non-trivial specialization within the template (panel (c)). Indeed, when looking at the output spike times of the two specialist neurons, the corresponding histograms suggest a separation in time. Here, one neuron fired at the beginning of the stimulus presentation and the other at the end (panel (c)).

This finding indicated that, in fact, the two neurons responded to different features within the target poisson template. This strategy offers a more robust firing for behaviorally relevant patterns: If the preferred feature of one neuron is present in the corresponding pattern, this neuron evokes a response and thereby accounts for a correct pattern assignment by the population. The target is directly identified even though the second neuron potentially fails to fire for this pattern due to noise in the structure of the second feature. We get an idea of how this focus on different features was implemented in the neuronal population by looking at the synaptic weights after learning. All input synapses that contained exactly one spike in the target template had negatively correlated weights for both neurons: The neuron that fired early in time had positive weights for all synapses where the input spike within the target template arrived early, and smaller or negative weights for synapses with late arriving input spikes in the target patterns. For the later responding neuron, this relation was inverted, yielding negatively correlated weights between both neurons (Figure 4.4, illustrated in red in panel (c) and (d)). In general, the more spikes an input synapse contains for the target template, the more clearly firing on this channel indicates the presence of a target pattern. Hence, both neurons developed positive weights for these synapses with many input target spikes. Similarly, weights for input channels without any spike in the target template were clearly negative for both neurons (illustrated in blue). If the neurons received an input spike through one of these synapses, this spike could only originate from either the poisson background noise in a target pattern or from the feature or the background of a null pattern, hinting with higher probability to the presence of a null pattern. Taken together, when grouping input synapses according to the number of input spikes in the single target template, we found positive correlations between both neuron's weights across synapse groups and negative correlations within synapse groups. These findings can be interpreted as a compromise between rate coding (used for rough identification of the target) and temporal coding (used for specialization within the target).

4.2.2 *Consulting more than one Neuron*

The finding of sharing one template by spreading decisions in time raised the question of how many neurons could temporally specialize on one 500ms template, implicitly asking for the minimally possible width of the spike eliciting time window. To investigate this problem, we considered a target class that was composed of only a single noisy subclass. Specifically, we chose to delete spikes with a high input spike deletion probability of $p_{del} = 0.8$. Together with the poissonian background noise, this remained the only noise source as we did not include temporal jitter to minimize

interference with the temporal input structure. We trained neuronal Tagging populations with different population size, ranging from $m = 1$ to $m = 20$, to detect this noisy subclass in order to evaluate the degree of temporal specialization depending on the size of the population.

The final population errors after training of each population showed a clear dependency on the number of neurons. Whereas a single neuron obtained a classification error of about 25% in this difficult detection task, the error decreased monotonically with m, saturating at about 4.6% for a population of 20 neurons. This error improvement suggests a benefit from larger populations which might be realized by means of temporal specialization within the subclass. Indeed, output spike time histograms of the participating neurons spanned the whole stimulus time window and overlaped only slightly for moderately sized populations (exemplarily shown for $m = 8$ in panel (c)). Firing time windows narrowed down when comparing $m = 8$ neurons to output histograms of $m = 3$ neurons (panel (b)). To obtain a quantitative measure, we defined the width of a tuning curve as the 90% coverage range of individual neuron densities resulting from a kernel density estimator. We calculated the densities based on the sorted latencies, pooled over all simulations (resulting densities for $m = 8$ neurons depicted in the inset panel of panel (d)). This measure revealed a fast drop until a population size of about $m = 5$ neurons and showed saturation for larger populations. Despite the saturation, the firing window width decreased monotonically, indicating a uniform partition of the temporal space for all investigated population sizes that was accompanied by a reduction of the population error for larger m.

One parameter that has been disregarded so far is the decision threshold d that determines the number of necessarily firing neurons for predicting the presence of a target. In the scenario above, increasing the number of neurons m led to a reduction of the error to some degree, but saturation took place before perfect classification was achieved (minimal error 4.6%). Can an increase of the number of required neurons for a target decision further reduce the population error? If d is fix and the population size m grows to infinity, the false positive rate of individual neurons must approach zero to ensure a population false positive rate below one. It is unclear whether in such a situation the large false positive rate could be compensated by the hit rate of the population. Hence, we hypothesize that for large m we also need to increase the decision threshold to yield high classification performance of the population.

For the same scenario of the single difficult subclass and populations of $m = 20$ neurons, we varied the decision threshold d from 1 to m. Consistent with the hypothesis of suboptimal classification with $d = 1$ for large populations, we indeed found a reduction of the error to 2.2% (Figure 4.6) for an optimal decision threshold of $d = 3$. In this population, erroneous firing of up to two individual neurons would be tolerated by the population decision. This spike tolerance in turn allowed for broader and hence overlapping output spike time histograms (Figure 4.6, panel (b)).

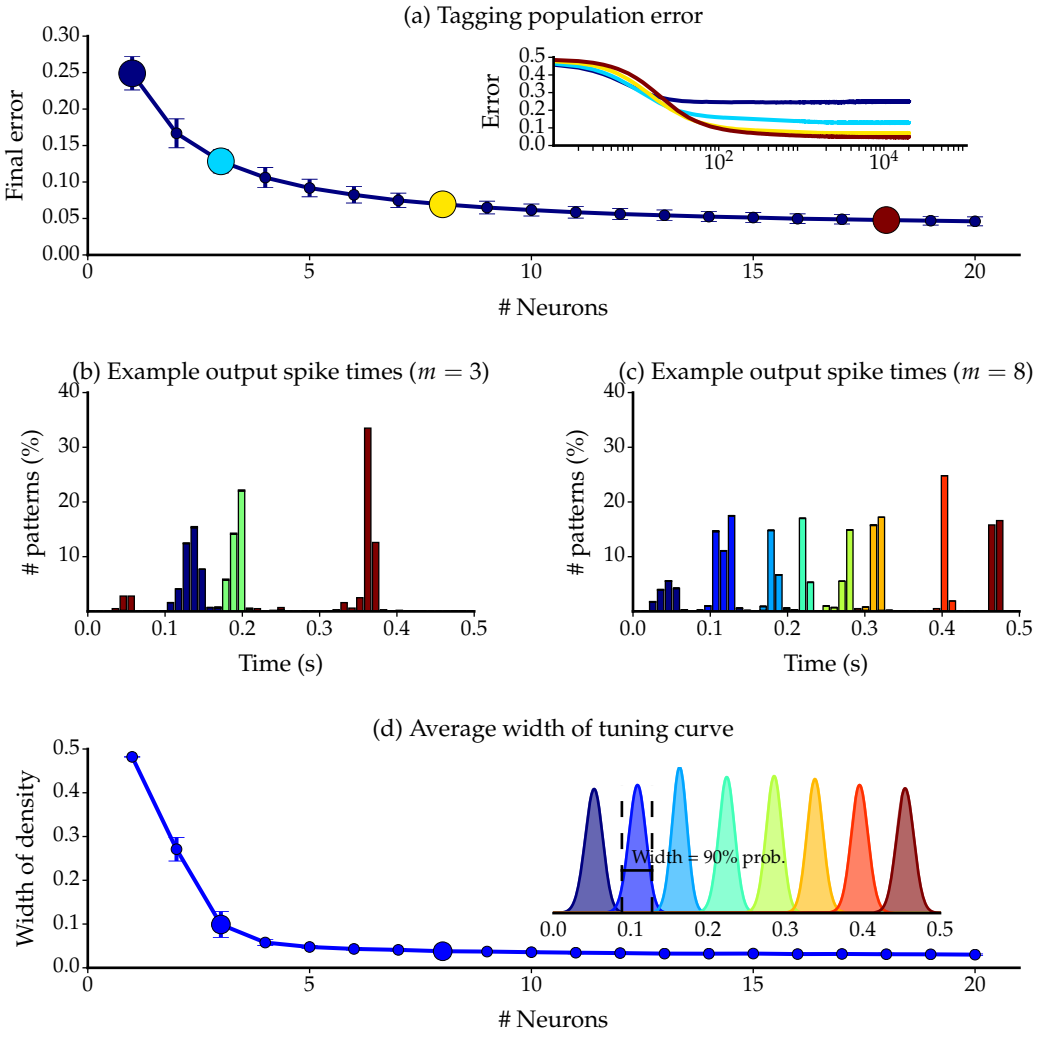

Figure 4.5.: LIMITS OF TEMPORAL SPECIALIZATION. Tagging populations with varying number of neurons learned to distinguish one single noisy subclass from poissonian noise. (a) Final population error (fraction of 1) after $lmax = 20000$ learning epochs for varying number of neurons. Error bars denote mean and standard deviation across all 100 simulations. The final error seems to saturate before reaching perfect classification. Individual error traces are shown in the inset panel for $m = 1, 3, 8, 18$ neurons. (b) and (c) Output spike time histograms for a representative Tagging population with $m = 3$ neurons and $m = 8$ neurons, respectively. (d) Width of the spike eliciting time window as 90% coverage range of output spike densities (see inset panel) depending on the number of neurons, averaged over all m neurons. Inset panel: Density plots of output spike timings for $m = 8$ neurons for all simulations, the jth density calculated by pooling all latencies from all simulations' neurons that yielded the $j$ths earliest mean output latency among the population.

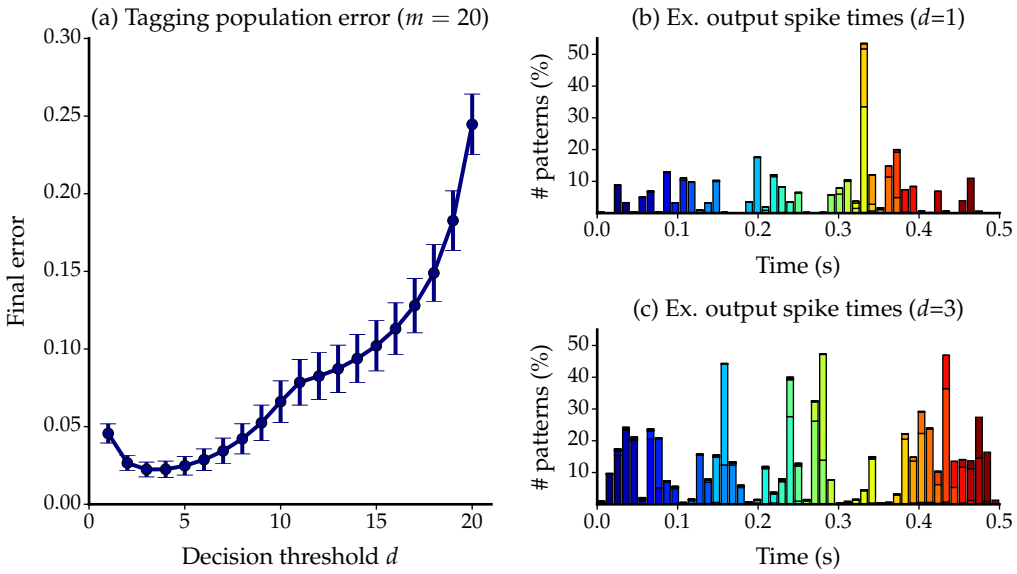

Figure 4.6.: CLASSIFICATION IMPROVEMENT WHEN CONSULTING MORE NEURONS. A Tagging population of $m = 20$ neurons learned to distinguish a single noisy target pattern from poissonian noise for different decision thresholds d. (a) Final population error after $lmax = 20000$ learning iterations. The optimal error of 2.2% was obtained with $d = 3$. (b) Output spike time histograms for a representative Tagging population with the optimal $d = 3$ span the whole time interval and overlap to a certain amount to ensure simultaneous firing of at least 3 neurons per target presentation.

In general, the optimal decision threshold most likely does not only depend on the number of neurons and subclasses, but also on the diversity of the individual subclasses. Hence, providing a general rule how to choose d is not straightforward. We will discuss the use of a theoretical error expression as a predictor for the expected error to *a priori* set the decision threshold in the following section.

4.3 PREDICTING THE TAGGING POPULATION ERROR

One question that is also related to the optimal choice of the decision threshold d is if the Tagging population error can be predicted based on knowledge of the individual neurons' properties. More specifically, this general question can be subdivided into two parts: (1) Under which assumptions can we derive an explicit representation of the theoretical population error? And (2) Does an empirical Tagging population fulfill these assumptions?

4.3.1 *Theoretical Population Error*

In order to address the first question, we start by deriving a theoretical representation of the population error for two situations: First, assuming that all individual firing properties of each neuron and each subclass are known, and second that neurons in the ensemble optimally share the subclasses in a homogeneous way with common

hit rates and miss rates. In both situations, all neurons are assumed to fire in an uncorrelated manner.

Error Calculation for known Firing Matrices

We investigate the error for a Tagging ensemble of m neurons and a decision threshold of d for a stimulus setup where the target class is composed of k templates. Let x denote the input spike pattern and ℓ the corresponding label as well as $\tilde{t}$ the template identity if $\ell = 1$. We assign a false positive rate f_l for each neuron $l = 1, \ldots, m$ that applies for presented null patterns. Furthermore, each template $j = 1, \ldots, k$ is identified with a hit probability $h_{l,j}$ for neuron l. With these notations, the overall error E for the tagging algorithm, is composed of the population false positive rate F and miss rate M:

$$\begin{aligned} E &= F + M \\ &= p_0 P(\hat{f}(x) = 1 | \ell = 0) + \sum_{j=1}^{k} p_j P(\hat{f}(x) = 0 | \ell = 1, \tilde{t} = j) \end{aligned}$$

with p_0 being the probability that a pattern of the null class is shown, $p_j, j = 1, \ldots, k$ the probability for patterns of the different target templates.

For the derivation of the overall false positive rate, all $m!$ possible permutations of the neurons need to be considered since each neuron has its own false positive rate that needs to be taken into account for calculating the firing probability for a null pattern. For this, let $\tau_z(q)$ denote the qth element of the zth permutation of the vector containing all neuron indices. We get:

$$\begin{aligned} F &= p_0 P(\hat{f}(x) = 1 | \ell = 0) \\ &= p_0 P(\#Neurons fire \geq d | \ell = 0) \\ &= 1 - P(\#Neurons fire < d | \ell = 0) \\ &= p_0 \left(1 - \sum_{r=0}^{d-1} P(\#Neurons fire = r | \ell = 0) \right) \\ &= p_0 \left(1 - \sum_{r=0}^{d-1} \sum_{z=1}^{m!} \prod_{q=1}^{r} f_{\tau_z(q)} \prod_{p=r+1}^{m} (1 - f_{\tau_z(p)}) \frac{1}{r!(m-r)!} \right) \end{aligned}$$

Similar calculations are necessary for the explicit representation of the miss rate.

$$\begin{aligned} M &= \sum_{j=1}^{n} p_j P(\hat{f}(x) = 0 | \ell = 1, \tilde{t} = j) \\ &= \sum_{j=1}^{n} p_j P(\#Neurons fire < d | \ell = 1, \tilde{t} = j) \\ &= \sum_{j=1}^{n} p_j \sum_{r=0}^{d-1} P(\#Neurons fire = r | \ell = 1, \tilde{t} = j) \\ &= \sum_{j=1}^{n} p_j \sum_{r=0}^{d-1} \sum_{z=1}^{m!} \prod_{q=1}^{r} (h_{\tau_z(q)j}) \prod_{p=r+1}^{m} (1 - h_{\tau_z(p)j}) \frac{1}{r!(m-r)!} \end{aligned}$$

This complicated representation of the error allows for reproducing empirically observed errors and the investigation of very specific cases. However, optimization of tagging-specific parameters is practically impossible under this general assumptions, since the entire behavior of the individual neurons needs to be known. Therefore, we additionally derived an error for a more convenient setting where the strategy of the neurons is assumed to be equal and symmetrically distributed among the templates. The simplified error term that will be derived in the following subsection agrees with the calculations here by setting the corresponding parameters appropriately.

Error Calculation for known Strategies

In contrast to the previous investigations, we now consider the specific situation where each neuron specializes to learn exactly $\tilde{S}$ templates of the target group. We assume that for this $\tilde{S}$ and the given setting of m and k a symmetrical strategy exists such that each template is represented by the same number of specialists, i.e. $S = m/k \cdot \tilde{S}$ is an integer. Furthermore, each neuron has the same individual error rate $e = f + (1 - h)$ where h denotes the hit rate for a learned template and f the false positive rate. Importantly, in this scenario the false positive rate does not only apply to patterns of the null class but also to target patterns from a different template than the one that the neuron is specialized to. In analogy to the general case, the overall error of the ensemble is composed of

$$\begin{aligned} E &= F + M \\ &= p_0 P(\hat{f}(x) = 1 | \ell = 0) + \sum_{j=1}^{k} p_j P(\hat{f}(x) = 0 | \ell = 1, \tilde{t} = j) \end{aligned}$$

With the above assumptions, the overall false positive rate can be derived via the binomial distribution as

$$\begin{aligned} F &= p_0(1 - P(\#N < d|\ell = 0)) \\ &= p_0\left(1 - \sum_{r=0}^{d-1} P(\#N = r|\ell = 0)\right) \\ &= p_0\left(1 - \sum_{r=0}^{d-1} f^r(1-f)^{m-r}\binom{m}{r}\right), \end{aligned}$$

since the probability to fire for a null pattern is the same for each neuron, namely the false positive rate f. For the miss rate, the situation is slightly more complex. Here, we have to keep in mind that for each template a set of S specialists and a set of $m - S$ non-specialized neurons exist. The probability that exactly l neurons fire, is composed of the sum over $q = 1, \ldots, S$ of selecting one specialist, multiplied by the corresponding firing rates and the converse probability to fire for the remaining $m - r$ neurons. The probability of drawing exactly q success objects in a sample of r without replacement from a population of m with S successes in total, is given by the hypergeometric distribution $H(m, S, r)$:

$$P(x = q|m, S, r) = \frac{\binom{S}{q}\binom{m-S}{r-q}}{\binom{m}{r}}$$

Multiplied with the number of possibilities to draw r of m elements, we get

$$\begin{aligned} M &= \sum_{j=1}^{n} p_j P(\#N < d|\ell = 1, \tilde{t} = j) \\ &= \sum_{j=1}^{n} p_j \sum_{r=0}^{d-1} P(\#N = r|\ell = 1, \tilde{t} = j) \\ &= \sum_{j=1}^{n} p_j \sum_{r=0}^{d-1} \sum_{q=0}^{r} P(x = q|m, S, r) h^q f^{r-q}(1-h)^{S-q}(1-f)^{m-S-(r-q)}\binom{m}{r} \\ &= \sum_{j=1}^{n} p_j \sum_{r=0}^{d-1} \sum_{q=0}^{r} \frac{\binom{S}{q}\binom{m-S}{r-q}}{\binom{m}{q}} h^q f^{r-q}(1-h)^{S-q}(1-f)^{m-S-(r-q)}\binom{m}{r} \\ &= \sum_{j=1}^{n} p_j \sum_{r=0}^{d-1} \sum_{q=0}^{r} \binom{S}{q}\binom{m-S}{r-q} h^r f^{l-q}(1-h)^{S-q}(1-f)^{m-S-(r-q)} \end{aligned}$$

In the simple case of having the same probability for each template and letting a null pattern and a target pattern occur equally often, we arrive at the simplified expression:

$$E = \frac{1}{2}\left(1 - \sum_{r=0}^{d-1} f^r (1-f)^{m-r} \binom{m}{r}\right) \tag{4.1}$$

$$+\frac{1}{2}\sum_{r=0}^{d-1}\sum_{q=0}^{r} \binom{S}{q}\binom{m-S}{r-q} h^q f^{r-q}(1-h)^{S-q}(1-f)^{m-S-(r-q)} \tag{4.2}$$

4.3.2 *Theoretical Error Space for a Single Target Class*

In the simple case of a target class that is represented by only a single template, the set of possible strategies by each neuron reduces to $\tilde{S} = 1$, so $S = m$. Hence at least in this specific situation, the error in formula (4.2) simplifies to the usual binomial error

$$E = \frac{1}{2}\left(1 - \sum_{r=0}^{d-1} f^r (1-f)^{m-r} \binom{m}{r}\right) + \frac{1}{2}\sum_{r=0}^{d-1} \binom{m}{r} h^r (1-h)^{m-r} \tag{4.3}$$

and the error space can be fully explored for different values of f, h and d.

We evaluated this error space for an uncorrelated homogeneous population of $m = 10$ neurons, as a function of the individual false positive and miss rate. If the miss rate ($ms = 1 - h$) and the false positive rate are zero, the error is zero regardless of d. At the other extreme, if the individual error is 0.5, with either $f = 1$ or $ms = 1$, the population error is 0.5 as well, again regardless of d. Within this triangle, the population error varies smoothly and highly depends on the decision threshold d. Figure 4.7 shows the theoretical population error color-coded in the f-ms-plane for the optimal d. The black contours depict the regions where the optimal d changes to the next higher one. For the outer most left slice $d = 1$ is optimal, for the outer right $d = 10$ is. In any of these segments, the error decreases to the lower left corner. As observed in the previous simulation studies, for noisy patterns an empirical Tagging population never reaches an exact error of zero but saturates at finite error rates that depend on problem-specific parameters. To investigate the evolution of neuron-specific error components in an empirical population during learning, we trained a Tagging population with $m = 10$ neurons to distinguish a single-template target class from null patterns with noise parameters $\sigma = 0$ and $p_{del} = 0.8$ for decision thresholds ranging from $d = 1$ to $d = 10$.

Also visualized in Figure 4.7, the colored dots denote the final empirical false positive/miss rate pairs averaged over the population members, with the black trajectories monitoring the development of the population components for a single seed. Almost all simulation's trajectories ended in the region where the decision threshold that was used for training was optimal (except for the largest ds). Reaching the d-optimal segment mostly already happened within the first learning epoch (trajectories start at

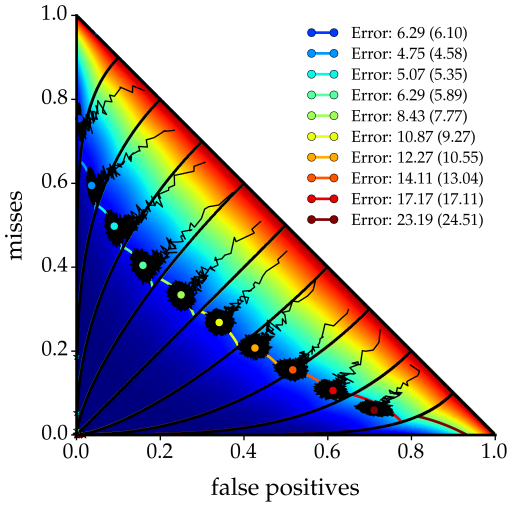

Figure 4.7.: THEORETICAL ERROR SPACE FOR A SINGLE SUBCLASS. The triangle color codes the optimal theoretical population error based on formula (4.3) for a population of $m = 10$ neurons with homogeneous individual false positives and misses on the x- and y-axis. Optimal decision thresholds d depend on the ratios of both error components, regions with the same optimal threshold are separated by black lines. The optimal d is close to 1 for low false positive rates (outermost left panel) and close to m for low miss rates (outermost right panel). Empirically measured population average values of neuron-specific error components are plotted as colored dots for different decision thresholds. The corresponding population error is depicted in the legend together with the theoretically optimal error for these *f-ms* constellations in parentheses.

false positives and misses obtained within the first learning epoch of $p = 1000$ patterns). After reaching the corresponding segment, the Tagging population improved its error smoothly by changing the individual error components along the gradient until saturation was reached. These findings indicate a good learning behavior of the Tagging population independent of the decision threshold, in the sense that all populations found individual *f-ms* ratios that corresponded to a steep gradient of the population error for the trained threshold. The corresponding empirically measured population errors (shown in the legend of Figure 4.7) revealed $d = 2$ as optimal choice for this specific classification problem. The theoretical error (corresponding to the color code of the error plane) for the empirically measured false positives and miss rates were similar and agreed with the empirical population on the optimal choice of d (numbers in parenthesis of the legend). Here, it should be noted that the two errors are not necessarily equal, since the theoretical error was obtained by averaging the individual *f-ms*-characteristics of the $m = 10$ neurons in the population. Furthermore, calculation of the theoretical error underlied the assumption of independence of the population components. However, since most empirical error rates did not qualitatively deviate from the theoretical error, the independence assumption was not necessarily rejected. For further considerations of these independence assumptions, see the following sections where we investigated empirical and theoretical population errors for a more complex scenario.

4.3.3 *Single Neuron Error Curves*

Motivated by the previous results that indicated indeed an uncorrelated firing at least for single target templates, we were interested if the theoretical population error in formula (4.2) can actually predict the Tagging population error in arbitrary situations. This would especially allow to select the optimal decision threshold d before setting up the Tagging population just based on individual neuron properties. The single neu-

ron error components strongly depend on the difficulty of the classification problem. Furthermore, for a given noise regime, this error should be correlated with the number of templates a neuron has to learn. Hence, for any given classification problem, in order to design the optimal population and determine its error, we need to know the individual neuron performance based on different numbers of templates.

Aiming at predicting the error of an exemplary problem of $m = 20$ neurons that need to distinguish $k = 5$ subclasses from null patterns, we trained individual neurons with 100 different starting seeds for different templates where we raised the number of templates to be learned from one to five. Spikes of each pattern were shifted with $\sigma = 100ms$ and deleted with $p_{del} = 0.4$. As already shown in Figure 4.7, optimal false positive and miss rates may be severely imbalanced depending on the decision threshold in a (hypothetical) Tagging population. Hence, we needed to provide options of such imbalanced error components when training the individual neurons. In order to model these different focuses on either false positive or miss rate, we varied the probability of a target pattern from 1% to 99%. This procedure was based on the idea that for only rarely occuring target features minimizing the firing for null patterns generally has a stronger beneficial effect on the error than maximizing the detection rate of rare targets, and vice versa. Indeed, higher target probabilities yielded larger empirical false positive rates but smaller miss rates and the other way around for smaller target probabilities (first panel of Figure 4.8). The individual neuron error curve described a convex function. Consequently, unbalanced error components in general lead to a larger individual error when $e(f)$ treats both components equally as $e(f) = 1/2f + 1/2ms$ (second panel).
For the empirical measured individual *f-ms* pairings, observations from one starting seed appeared to lie on a curve. Hence, for each seed we linearly interpolated the curves on an equally spaced grid and afterwards averaged them over 100 seeds to obtain the mean individual error curves (solid lines). These mean curves for each number of templates were used to calculate the theoretical error of tagging populations in the following.

4.3.4 *Tradeoff: Correlations ↔ Individual Accuracy*

Based on the previously measured individual neuron curves, a direct comparison of the empirical Tagging error and the theoretical error according to formula (4.2) could yield insights into how well the Tagging error could be predicted by the individual neuron properties.

For the setting of $m = 20$ neurons and $k = 5$ subclasses (with same seeds as in the individual neuron investigations), we trained Tagging populations with different decision thresholds and compared the resulting population error to the theoretical one (first panel of Figure 4.9).

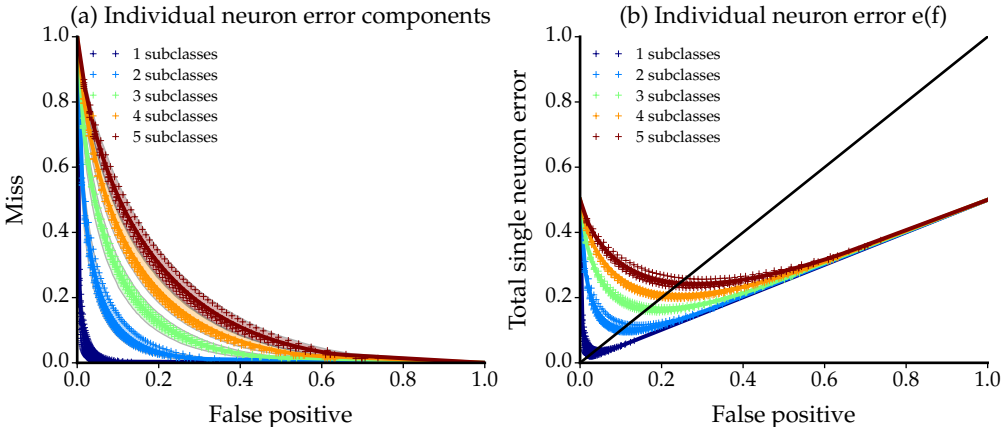

Figure 4.8.: SINGLE NEURON ERROR CURVES. (a) Empirically measures single neuron error components for different number of subclasses in the target class (color-coded). Each dot represents a single simulation obtained for different target probabilities and starting seeds (three shown). Training on higher target probabilities usually yielded values on the lower right corner and vice versa. Solid lines represent the interpolated curve after binning the empirical values on a grid, averaged over 100 seeds, envelopes mark the 5% and 95% quantiles. (b) Same empirical values, represented as the total single neuron error when both components are treated equally: $e(f) = 1/2ms + 1/2f$. The black line indicates the main diagonal. The minimal error was achieved for $ms = f$.

Surprisingly, the empirical Tagging population error curve deviated strongly from the theoretical error for almost all decision thresholds. Why did the Tagging population perform significantly worse than a theoretically optimal population? The obtained large deviations must be traced back to violations of one of the three major assumptions that underlied the theoretical error formula: Either a) the population was not homogeneous, b) the neurons were correlated or c) the individual error curves did not apply for this population. For some simulations, inhomogeneity could be part of the problem, but also for purely symmetrical solutions this error difference did not vanish. Hence, we tested the validity of the two remaining assumptions by calculating the individual neuron error components and the pairwise noise correlations (defined as by Montijn et al. (2014), see Section 2.4). The noise correlations between any pair of neurons in the Tagging populations were small. However, the individual neuron errors clearly exceeded the ones obtained when training individual neurons outside of the populations that yielded the individual neuron error curves (exemplarily shown as colored dots in Figure 4.9).

Why did these deviations from the single neuron error curves arise? Did the Tagging population fail to find the optimal solution? Or was the empirically optimal solution actually different from the theoretical one based on the single neuron error curves? To investigate these questions, we took the same sets of neurons and trained them separately, solely on predefined specific target patterns based on the theoretically optimal strategy. As this procedure agreed with the one utilized for generating the single neuron error curves, it automatically ensured a performance of these individually trained neurons close to the average curves. For an equally good representation of all subclasses, we assigned target subclasses to the individual neurons

by simply shifting the responsibilities by one template for each pattern. Specifically, neuron j was trained to distinguish targets composed of templates $j, \ldots, j + \tilde{S}$ (with circular boundary conditions) from the null class. Here, $\tilde{S}$ denotes the theoretically optimal strategy for the given d obtained with error formula (4.2) (for a detailed description of the training process, see appendix A.2). After separate training of the individual neurons, we combined all neurons in a population and measured the resulting population error with the respective decision threshold. The population error based on individual training turned out to be very similar to the Tagging error - and thereby also significantly higher than theoretically possible (Figure 4.9, panel (a), light gray curve). This finding ruled out the answer that the Tagging algorithm just failed to find the better solution with lower individual errors and, instead, indicated that there is no such better solution.

By construction, the separately trained population was homogeneous and the neuron specific errors were in the same range as the average neuron curves underlying the theoretical error curve. Hence, the large deviations found for these simulations could only be explained by high correlations between neurons. Indeed, the pairwise correlations showed to be considerably higher than for the Tagging learning (Figure 4.9, panel (b), average values depicted as dark gray dots). Correlations showed to be specifically high between neurons that favored the same subclasses (light gray dots indicate pairwise correlations, separated by the number of common subclasses of the respective two neurons).

The results from Tagging and separate training together suggest that neuronal populations face a tradeoff: Either the neurons show correlated responses but have low individual error rates (combined solution) or the neurons decorrelated their responses but have higher individual error rates (Tagging solution). In fact, for general regression ensemble methods that base their decision on a weighted average of the individual outputs, the ensemble generalization error can be shown to be decomposed into two parts. The weighted average of the individual generalization errors on the one hand, and the weighted average of the ambiguities that captures the diversity of the ensemble members one the other (Krogh and Vedelsby 1995), both components facing a tradeoff (Chandra et al. 2006). Even though the decision of the Tagging population does not exactly match a weighted average, the previous findings indeed support a decomposition of the population error into both components, with the Tagging algorithm focusing on minimizing the ambiguity summand, the separate training (by construction) the individual error component.

4.4 SPECIALIZATION ON CONTINUOUS FEATURES

So far we have assumed that the target class consists of discrete subclasses, as the different vocalizations indicating detection of high quality food (Gifford et al. 2005). However, many sensory objects vary on a continuous scale - one may think of differ-

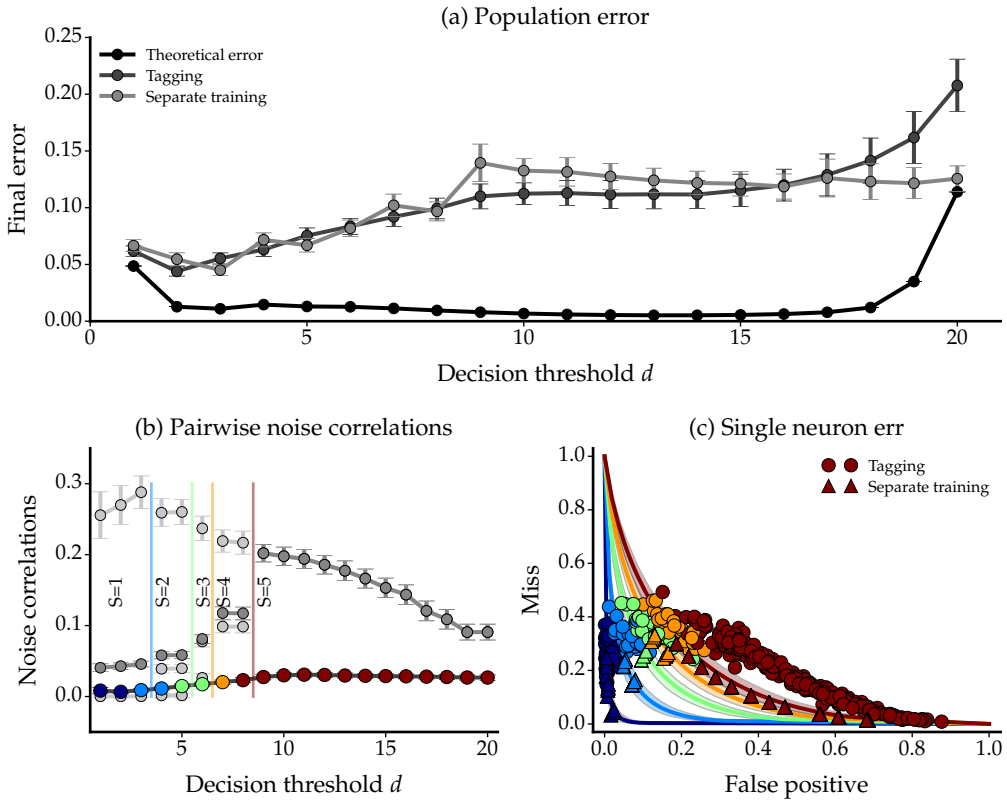

Figure 4.9.: TRADEOFF: INDIVIDUAL ACCURACY ↔ PAIRWISE CORRELATIONS. For a setting with $m = 20$ and $k = 5$, Tagging populations as well as separate neurons were trained with different decision thresholds from $d = 1$ to $d = 20$. (a) Population error for empirical and theoretical populations for different decision thresholds. Empirical Tagging populations (middle gray) yielded similar errors as the separately trained neurons (light gray). Both performances were clearly inferior to the optimal theoretical error (black) according to formula (4.2) with single neuron errors obtained from the empirical single neuron error curve averages of Figure 4.8. (b) Average noise correlations between any pair of neurons within the Tagging population (colored dots) and between any pair of separately trained neurons (gray colored dots) for different decision thresholds. For separate training, dark gray dots denote average values between all pairs, whereas light gray dots show calculations among subsets of pairs that contain a fixed number of common subclasses (as defined by the strategy): Highest values always denote pairs with highest number of common subclasses for the given strategy, while lowest values denote pairs of neurons with no common favored subclass. Colored lines in the plot separate regions of optimal strategies. The same color coding in the Tagging dots denotes the most frequently used strategy within all simulations (see also Section A.2). Even though the strategies used for Tagging and separate training were similar, separately trained neurons show a clearly higher pairwise correlations, even when averaged over all (including non-connected) neuron pairs. (c) Empirical individual neuron error components for Tagging populations (dots) and separately trained neurons (triangles) for one randomly chosen simulation of each decision threshold. Values are color-coded according to the firing strategy used by that specific neuron in the population. Whereas separately trained neurons by definition realized error components close to the average single neuron error curves, the components for the Tagging populations usually exceeded the curves by cost of a higher individual error.

ent handwritings that all denote the same written word, the same sentence spoken by different voices or dialects etc. Also, smoothly varying positions of a visual object can be considered as continuous subclasses as well as different views of a face in visual face detection tasks. In the following simulation study, we investigate if the Tagging algorithm is also capable of specializing on continuous ranges of target stimuli.

For this purpose, we constructed presynaptic poisson patterns of approximately the same statistics as in the discrete setting to ensure comparability, but 'warped' between a discrete set of different template activity patterns in the target class to generate continuously varying stimulus realizations (see Section 2.3.2). This warping was realized by shifting spike times from one template continuously in time until they matched corresponding spike times from the adjacent template. The number of templates hereby negatively correlated with the similarity of two 'nearby' target input patterns. A circular, continuous value $\alpha \in [0, 1]$ modeled the morphing between the different templates and hence determined the actual representation of an input spike pattern. Again, the task of the neuronal population was to distinguish these continuously varying spike patterns of the target class from poisson patterns from the null class that did not rely on any structure but had the same statistics.

Is the Tagging algorithm also able to uncover structure within this continuous framework? To investigate this question, we followed a similar strategy as for the discrete subclasses and trained a Tagging population of five neurons to distinguish continuous features from the null class with a decision threshold of $d = 1$. The results were again interpreted in comparison with a single neuron, the naive unselective training approach and the attenuated learning procedure. The error curves showed an improved classification performance for the Tagging learning rule, similar as for the discrete patterns (Figure 4.10, panel (a)). Inferior classification performance by the attenuated learning approach indicated a benefit of specialization with small decision thresholds. Indeed, Tagging populations achieved specialization by uniformly covering the total range of the continuous parameter by the five neurons. The firing rates formed clearly separated tuning curves for all population members whose bell-like shape reminded of empirically recorded tuning curves for example in orientation selective cells of the primary visual cortex (Henry et al. 1974, panel (b)).

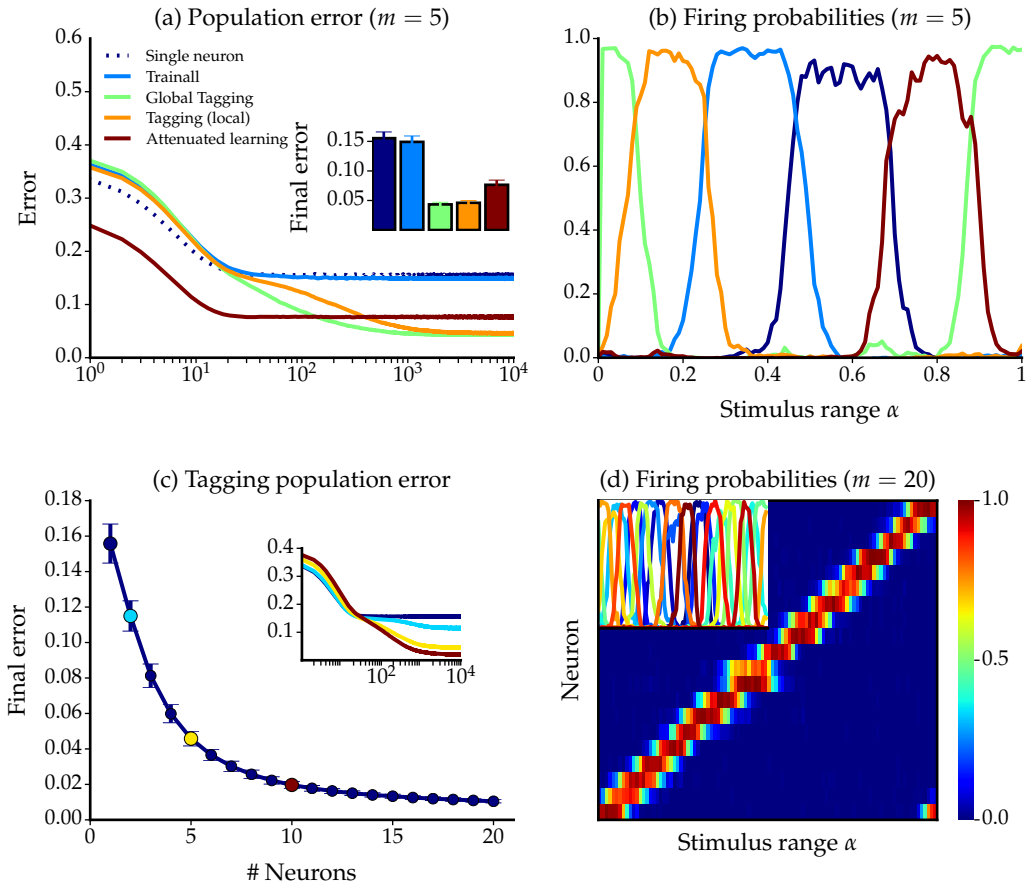

Figure 4.10.: SPECIALIZATION ON CONTINUOUS FEATURES. (a) Error curves for a single neuron and populations of $m = 5$ neurons on a binary classification problem with the target class consisting of continuously interpolated spike patterns between about 15 spike templates. Both Tagging variants (with $d = 1$) performed similarly well and yielded results superior to all remaining approaches, as also visible in the bar plot of final errors in the inset panel. (b) Firing probabilities of a representative (local) Tagging population for features on the continuous cyclic α range, each color representing an individual neuron. All patterns from the target class were represented with more than 70% firing probability by at least one neuron. (c) The final error after $lmax = 10000$ learning epoches decreased for larger Tagging populations (with $d = 1$). Inset panels show error trajectories for $m = 1, 2, 5, 10$. (d) Firing probabilities for $m = 20$ neurons in a Tagging population. The two-dimensional representation similar to panel (b) that is depicted in the inset panel does not provide much information due to the large number of bell-shaped curves. Hence, we additionally presented the firing probabilities as heat matrix in the main panel. Firing probabilities are color coded as for the discrete setting in the previous chapters, with neurons on the y-axis and the stimulus range α on the x-axis.

In analogy to the discrete single subclasses, we tested for saturation in terms of specialization on α ranges. For this, we increased the number of neurons in the local Tagging population. Consistent with the findings for the discrete input classes (Section 4.2), the error decreased for larger populations until saturation was achieved (Figure 4.10, panel (c)). Similarly to the temporal specialization obtained for discrete templates, the tuning curves reduced their width for more neurons and uniformly spanned the whole stimulus range. This monotonic decline in combination with the monotonically decreasing population error indicated a beneficial partition of the classification problem (Figure 4.11, tuning curve width is defined similarly to the temporal separation study as the fraction of α values that covered 90% of the firing probability of a neuron).

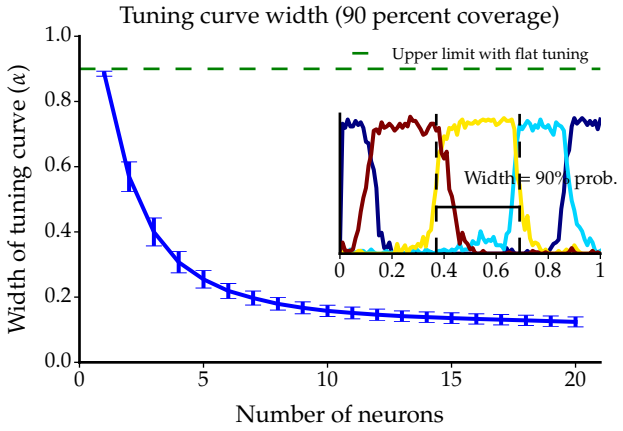

Figure 4.11.: TUNING CURVE WIDTH FOR CONTINUOUS FEATURES. Width of the tuning curves dependent on the number of neurons in the population for the continuous classification problem of Figure 4.10. The width of a tuning curve was defined as the fraction of α-grid values that covered 90% of the firing for the corresponding neuron, as visualized in the inset panel for an exemplary population of $m = 4$ neurons. A flat tuning with unselective firing for all parameter values would yield a tuning width of 0.90 (dashed green line).

5 SPEECH RECOGNITION

While we showed in the previous chapter that the Tagging algorithm induces specialization on artificial poisson input patterns and thereby improves classification performance, a generalization of the algorithm's capability to uncover structure in more realistic data remains questionable. In the following chapter we evaluate its performance on spoken digit recognition. We choose speech because it is rich in structure and hence might provide different cues of possible specialization. Specifically, we are interested in the hypothesis that identifying specific target digits might be facilitated by specialization on speaker-dependent properties, such as gender.

5.1 VARIABILITY OF SPEECH AND AUTOMATIC SPEECH RECOGNITION

When we talk to an unknown person on the phone, usually a rough picture forms in our head that automatically assigns a gender, an approximate age and sometimes even a likable or dislikable appearance to that unknown person. Even though this picture might change during the conversation due to the discussed content, much of this information is already embedded in the voice of the speaker. This content-independent paralinguistic or *voice information* is one reason why automatic speech recognition is still a complicated issue for machine learners. Two acoustic waveforms of the same word may have significantly different physical properties when spoken by different individuals. Large variability is ascribed especially to the gender of the speaker. Female voices have shown to evoke a higher mean fundamental frequency, perceived as pitch, in different languages (Pepiot 2013, Hillenbrand et al. 1994). The same studies report also fundamental gender-specific differences in the third and the fourth formants of vowels. Cross-gender differences appear to be language-dependent and hence need to be at least partly socially constructed and cannot be totally attributed to anatomical differences for example of the vocal folds in men and women (Pepiot 2013). Regardless of its origin, these known cross-gender differences and the accompanying high variability within the same spoken words lead to relatively bad speech recognition performance for machine learning approaches. Hence, many automatic speech recognition systems evaluate their classification rate on male and female data sets separately (Gütig and Sompolinsky 2009, Gharavian and Ahadi 2007, Mirghafori et al. 1994). On the other hand, having identified speech properties that systematically differ for speaker characteristics as gender, one can utilize these speech properties to infer speaker characteristics from speech recordings, as our brain does automatically for example during phone calls. For example, gender of the speaker can be deduced from the pitch measured with an automatic pitch detection procedure and simple template matching with an accuracy of more than 80% (Kumar et al. 2011). More sophisticated methods such as SVM together with a larger set of about 100 statistical speech features based on formants, pitch, intensity and spectrum yield even classifica-

tion rates of about 95% (Sedaaghi 2009). Gender classification could also be used as a preprocessing step for automatic speech recognition systems to improve performance on speech recognition (Vergin et al. 1996).

Within the last years, automatic speech recognition systems that focus on pure speech content classification have evolved with increasing complexity. Whereas for a long time hidden Markov models with Gaussian mixture models dominated the field (Gales and Young 2008), deep neural networks have entered the competition on the highest word accuracy rates (Mohamed et al. 2012, Hinton, Deng, et al. 2012). These models overcome the problem of speaker-dependent variability by means of a large number of parameters and hidden layers. However, also a simpler, biologically inspired system yields comparable results; Schafer and Jin (2014) trained a set of 1000 integrate-and-fire neurons to fire for a random acoustic feature in a single neuron-specific target word. In a second step, the order of the spiking responses of these auditory feature detectors is decoded by a hidden Markov model or a simple template matching algorithm to predict the spoken digit. The obtained spectro-temporal receptive fields of the neurons are similar to those of neurons in the secondary auditory cortex. Together with the high classification performance with this relatively simple approach these findings suggest that despite the trend to more complex networks in automatic speech recognition, the high performance of the auditory system might be achieved by relatively simple mechanisms.

Clearly, we did not expect to enter this competition and achieve the benchmark performances on a speech recognition task with the simple Tagging algorithm. Our motivation to apply Tagging to speech was a different one that resulted from the variability of speech: If the same spoken word shows fundamental different acoustic properties when spoken by a male or female speaker, do these differences reflect in the response properties of the individual neurons in a Tagging population? Can a population of for example two neurons that is trained to identify the digit 'five' yield improved classification performance compared to a single neuron by specialization on the gender of the speaker? As discussed above, the gender is a profound characteristic of the speaker, but of course not exclusively determining the auditory waveform. Even though the neurons do not have any explicit knowledge about the speaker characteristics during learning, we were interested if we could find any properties in the different neurons' spiking responses that relate to voice information. The TI46 data set that we base the analysis on in the following sections contains spoken words from 16 different speakers, 8 of which are female (see Section 2.3.3). Since to us the gender and the speaker identity is known, we could analyze the response properties to these features after training neurons on spike inputs that were generated from the acoustic wave forms. In a two-dimensional tonotopy-intensity map, input neurons were modeled as onset- and offset decoders of spectrotemporal events. Specifically, every input neuron was assigned a specific frequency band and an offset or onset threshold. An offset or onset decoding neuron elicited a spike whenever the power of the auditory signal in the corresponding frequency band crossed its threshold from above or below, respectively (for details see Section 2.3.3). With these input patterns that mimicked re-

sponses from neurons situated in the inferior colliculus, the Tagging population could be trained to distinguish certain target digits from the remaining data set. Similarly to the poisson pattern simulations, response statistics of the neurons in the population helped to test the hypothesis that Tagging improves classification performance by specialization on speaker-dependent properties.

5.2 DETECTING GROUPS OF DIGITS - EVEN VERSUS ODD

Before devoting our attention to the problem of specialization within single digits, we started by investigating the capability of the Tagging algorithm to uncover known structure for this new input pattern type. For that purpose, we investigated the tailored problem of distinguishing even from odd digits. With that classification problem, the target class was composed of the subclasses $\{0, 2, 4, 6, 8\}$. We hypothesized that training a Tagging population with $m = 5$ neurons and $d = 1$ would reveal this substructure in the neurons' firing responses. To reduce additional structure in the data, the training was performed on the male and female data set separately. We compared the (local) Tagging algorithm with a single neuron as well as with the gradient-like global version of the Tagging algorithm and the trainall training procedure (Figure 5.1). The classification errors[1] for the Tagging algorithms were significantly lower than the ones for the trainall as well as the single neuron for both data sets (Figure 5.1, panel (a)). This performance improvement could be at least partly traced back to specialization on the digit substructure: Even though firing matrices showed less clear structure than for the poisson patterns in Chapter 4, we could visually confirm slightly sparser and more specialized representations for Tagging than for unselective trainall (see the firing probability representations of two exemplary chosen populations in the left panels). To also quantitatively investigate the specialization of the populations, we utilized three measures that address different aspects on interneural relations and intraneural specialization (see Section 2.4). Evaluated for the optimal binary partition within the target substructure, the Kullback Leiber (KL) divergence of the measured firing probabilities from the perfect specialization distribution quantifies the degree by which a given substructure determines the firing behavior of individual neurons. Its population average was lower for neurons in the Tagging population than for those in the trainall algorithm. This agreed with also the visual impression that the Tagging neurons' firing behavior was determined more strongly by the even digit substructure than the trainall neurons' (panel (d)). As a second specialization measure, the Sum-Minus-Whole redundancy relates the information content about the substructure that is present in the whole population to the sum of the neurons' individual information components. Positive values indicate redundant information, negative values synergy between neurons. For the even vs odd spoken digit recognition task, the members of the Tagging population responded less redundant to the whole stimulus substructure (defined by all ten digits, regardless of null or target class) than those of the trainall population (panel (e)). Finally, correlation within a population was measured as the

1 We chose to visualize classification errors relative to single neuron errors in the following, especially since performances varied significantly between different digits and hence a direct representation of the absolute errors would be complicated in the following subsection.

average over all pairwise pearson correlations coefficients. We evaluated it on the target and null class separately to allow for an easier interpretation. On the target class, Tagging populations showed almost vanishing correlation values, supporting the results from Section 4.3 that stated that the Tagging population decorrelates responses between neurons. Also correlations on the null class remained low. The trainall population, however, showed significantly higher correlations on both classes (panels (f) and (g)), also consistent with a more redundant information in the population.

5.3 DETECTING SINGLE DIGITS - VOICE INFORMATION AS SPECIALIZATION CUE?

In order to examine the hypothesis of gender as a potentially interesting substructure, we considered the problem of distinguishing a specific target digit from all nine remaining ones. For each target digit, we trained populations of two neurons (with $d = 1$) on the total data set (male and female voices). Generally, with a population of two neurons the classification performance was improved compared to single neuron performance. Except for the target digits 0 and 2, the Tagging population yielded significantly lower errors than the trainall algorithm (t-test on 99% confidence level, see Figure 5.2).

How was this error improvement realized in the Tagging populations? We examined the specialization of the neuronal populations with respect to the structure of interest, gender. For some target digits, neurons in the population showed different firing rates to the male and female speakers. The most stable and sharp results were found for the populations designed to identify the digit 'seven' (Figure 5.2, panel (f)). Here, one neuron was active at more than 90% of the times a male spoke 'seven' but less than 20% of the times a female pronounced the target word, and the other way around for the second neuron. These results of clear gender-separation for the target digit 'seven' can be regarded as consistent with the findings from Pepiot (2013). Here, the authors reported a notably difference in the second formant between genders especially for open vowels, where both vowels in 'seven' belong to. The clear gender specialization that we observed in the Tagging population was indeed due to the selective training mechanism as training the neurons in an unselective manner yielded much more diffuse firing matrices. Here, much lower error rates for Tagging also reflected the benefit of specialization on gender for this specific digit 'seven'. To generally quantify the degree of specialization in the sense of how far the gender of the speaker determined the firing probability of a neuron, we calculated the KL divergence of the optimal binary partition (whereas here with two subclasses only one binary partition between male and female speakers existed). Small values of this specialization measure for Tagging compared to the trainall algorithm confirmed the empirically observed more pronounced gender specialization for the weakly selective learning approach (panel (e) in Figure 5.2). Consistent with the visual impressions, the KL divergence values showed the highest discrepancies between the two algorithms for the digit 'seven'. For the remaining target digits the differences were smaller if existent.

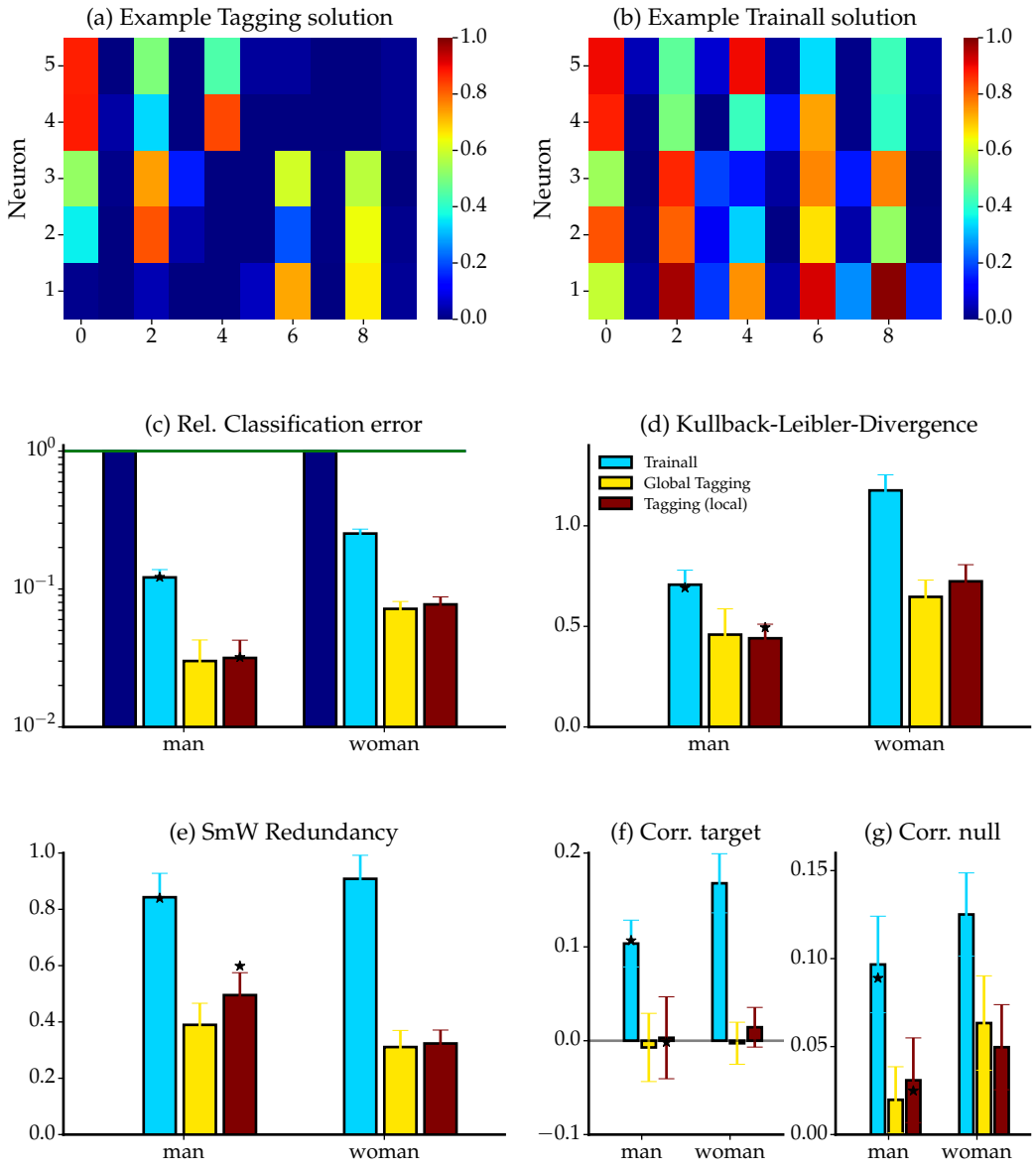

Figure 5.1.: SPOKEN DIGIT DETECTION: EVEN VS. ODD. A single neuron and populations of five neurons each were trained to distinguish even (target class) versus odd (null class) spoken digits with both Tagging variants and trainall. Speech signals from the TI46 data set served as inputs after transformation to spike trains via an acoustic frontend (see Section 2.3.3), classification was performed on the male and female data set separately. (a), (b) Firing matrices of a representative (local) Tagging and trainall solution for the male data set, respectively. Differences between both methods in terms of specialization were not as profound as for the poisson inputs in Section 4.1. (c) Final classification error of the population mechanisms relative to the single neuron error (dark blue bar) for both genders. The stars denote relative errors for both selected examples in (a) and (b). (d)-(g) Measures to characterize the degree of specialization and correlation within the Tagging and trainall populations. Low values of the KL divergence indicate subclass-specific responses of individual neurons, interpreted as the degree of single neuron specialization.

For some simulations on the digit 'five' and 'six', we found a relatively clear representation of male voices by one neuron, but shared responses from both neurons to female voices. We suspected that for these digits some, but not all discriminative information was hidden in the speaker's gender. The 'next higher' structure that we had access to in this data set was the identity of the speaker. Visualizing the firing probabilities on the finer scale of the speakers indeed uncovered preferred firing for different individuals (panel (h), 5.2). Speaker identity preferences should be reflected by a clearly reduced KL divergence of the optimal partition on speaker subclasses (panel (g)) compared to the superclass gender. Indeed, for many target digits we found a significantly smaller KL divergence for Tagging populations on this speaker identity scale and consequently larger KL divergence discrepancies between Tagging and trainall solutions. This suggests that the better classification rates for Tagging for many target digits could be ascribed to speaker identity specialization.

Based on the previous findings on specialization on the finer speaker identity scale, we wondered if enlarging the population size would improve classification performance further. In analogy to the simulations in 4.2, for example for the digit 'six' we hypothesized that in a population with four neurons, the additional neurons specialized on the female voices and hence shared the more difficult subclass on the finer speaker identity scale. Whereas for four neurons the gender specialization was reduced (which agrees with the idea that with only two classes, no four neurons can optimally specialize on them if they are not highly correlated), the speaker identity specialization indeed increased (panel (e) and (g) of Figure 5.3). Also for the digit 'six' we found a cleaner specialization on speaker identities within the female class (panel (h)).

In the classification problems in Section 4.2, the internal specialization was realized by firing at substantially different time points of the stimulus. Also for the speech processing we found output spike time differences that were in the range of the stimulus duration. However, we did not find significantly larger time differences between Tagging neurons and trainall neurons. We would therefore conclude that the different firing times are more a side effect of specialization on different features related to different speaker identities than the key property of specialization itself (panel (i) and (j) in Figures 5.2 and 5.3).

The pairwise pearson correlation values within the target class furthermore supported the idea of subclass specialization in the Tagging population. Highly negative correlations were found for the two digit setting, with the by far clearest results for digit 'seven'. These negative correlations were consistent with the clear gender specialization within this target population. For four neurons, the average pairwise correlations were generally smaller, but still negative (for the majority of target digits). The almost vanishing correlations for the trainall procedure in the target class were accompanied by highly positive correlations in the null class. These findings went hand in hand with a generally higher classification error for the trainall procedure due to the lack of specialization within the target class.

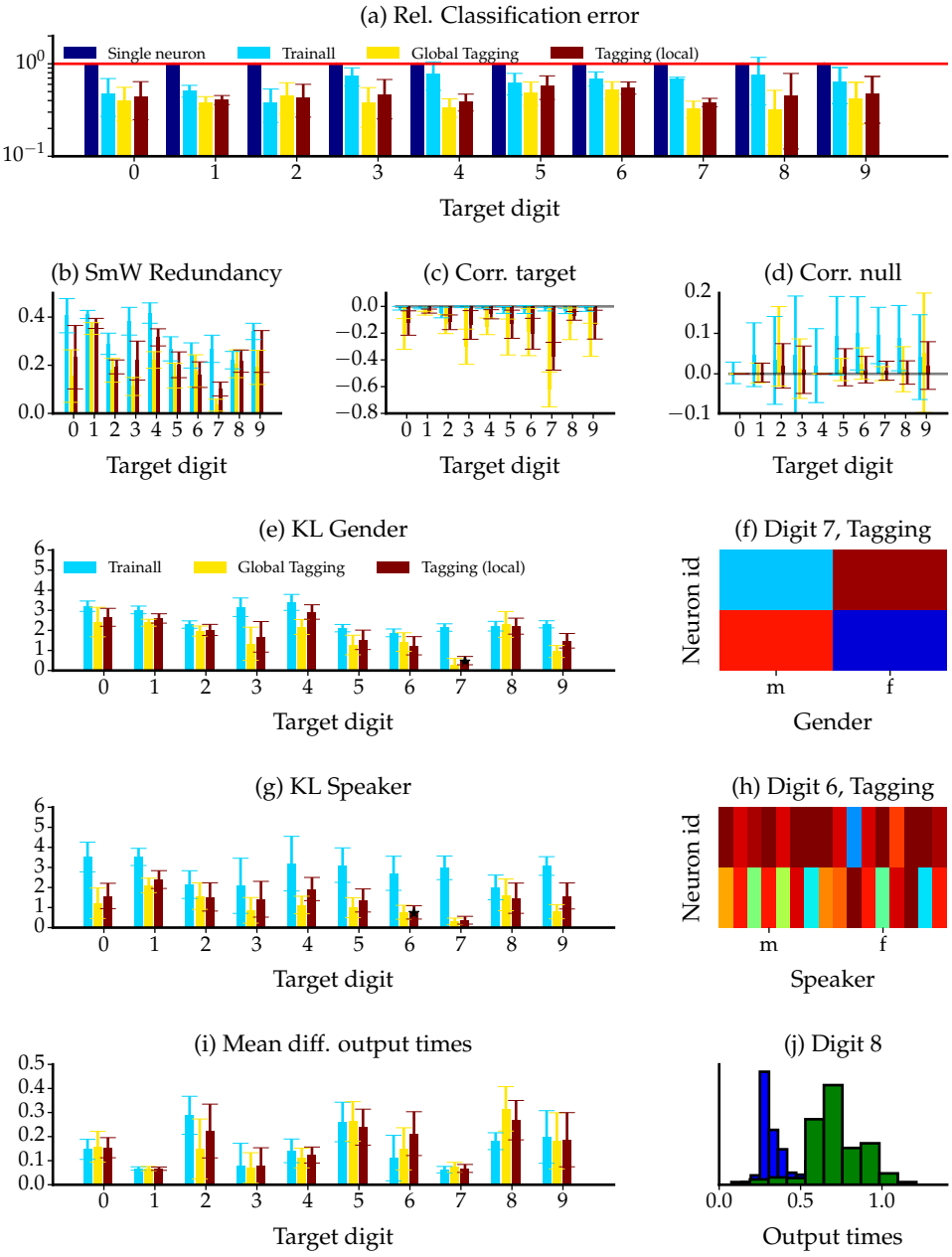
(a) Rel. Classification error
Single neuron
Trainall
Global Tagging
Tagging (local)
Target digit
(b) SmW Redundancy
Target digit
(c) Corr. target
Target digit
(d) Corr. null
Target digit
(e) KL Gender
Trainall
Global Tagging
Tagging (local)
Target digit
(f) Digit 7, Tagging
Neuron id
Gender
(g) KL Speaker
Target digit
(h) Digit 6, Tagging
Neuron id
Speaker
(i) Mean diff. output times
Target digit
(j) Digit 8
Output times

Figure 5.2. ***(preceding page)*:** SPOKEN DIGIT DETECTION: DETECTING SINGLE DIGITS WITH 2 NEURONS. A single neurons and populations of two neurons each were trained to distinguish a specific target digit from all remaining digits with both Tagging variants and trainall ($d = 1$). (a) Classification errors for trainall and Tagging populations relative to the single neuron error for all ten different target detection problems. For many target digits, the Tagging population improved classification not only to a single neuron but also to unselective population training. (b)-(d) Redundancy (measured on subclasses {nullclass, target male, target female}) and correlation measures to characterize the population behavior. Unselectively trained populations in general show to be more redundant and higher correlated on the null class than populations based on weakly selective training. Tagging populations showed negative correlations on the target class, indicating specialization within the target digit. (e) KL divergence for the possible specialization cue 'gender'. Digit seven showed a clear reduction of the KL divergence for the Tagging algorithm compared to trainall, a representative firing matrix (indicated by the star) resolved by the gender of the speaker for digit 'seven' in panel (f). (g) KL divergence for the possible specialization cue 'speaker identity'. Apart from digit 'seven', also especially digit 'six' and 'five' showed clearly reduced realizations for Tagging populations compared to unselectively trained populations. A representative firing matrix for digit 'six' resolved by the speaker identity within the target class is shown in panel (h). (i) Possible specialization by spreading decisions in time was analyzed by means of the average difference of output spike times between both neurons. Largest differences were found for digit 'eight', a representative latency histogram is depicted in panel (j).

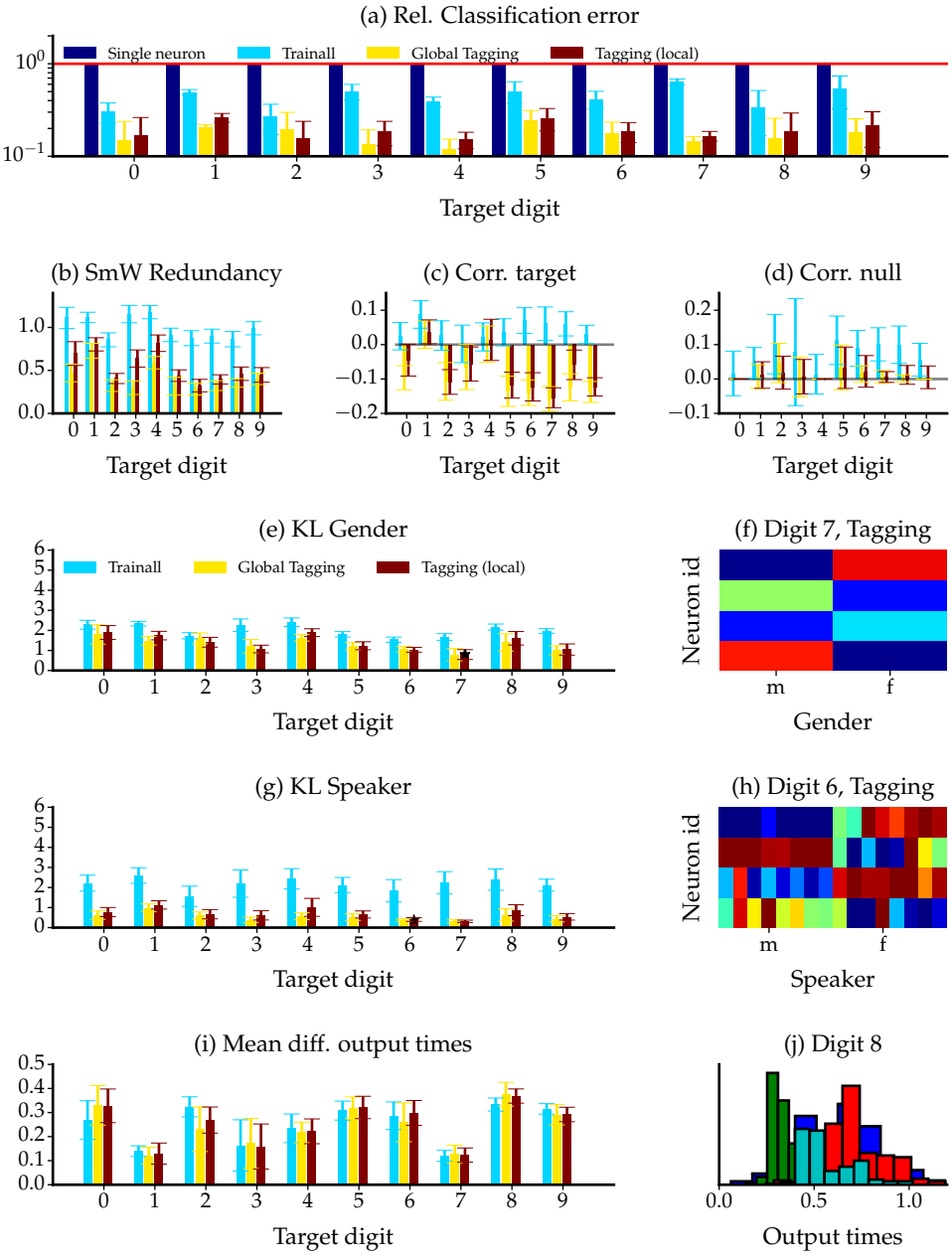
(a) Rel. Classification error
Single neuron
Trainall
Global Tagging
Tagging (local)
Target digit
(b) SmW Redundancy
Target digit
(c) Corr. target
Target digit
(d) Corr. null
Target digit
(e) KL Gender
Trainall
Global Tagging
Tagging (local)
Target digit
(f) Digit 7, Tagging
Neuron id
m
f
Gender
(g) KL Speaker
Target digit
(h) Digit 6, Tagging
Neuron id
m
f
Speaker
(i) Mean diff. output times
Target digit
(j) Digit 8
Output times

Figure 5.3. ***(preceding page)*:** A single neurons and populations of four neurons each were trained to distinguish a specific target digit from all remaining digits with both Tagging variants and trainall ($d = 1$). (a) Classification errors for trainall and Tagging populations relative to the single neuron error for all ten different target detection problems. For many target digits, the Tagging population improved classification not only to a single neuron but also to unselective population training. Generally, performances were superior than those for two neurons (Figure 5.2). (b)-(d) Redundancy and correlation measures to characterize the population behavior. Unselectively trained populations in general show to be more redundant and higher correlated on the null class than the populations trained with weakly selective training. Tagging populations showed negative correlations on the target class, indicating specialization within the target digit. (e) KL divergence for the possible specialization cue 'gender'. Again, digit seven showed a clear reduction of the KL divergence for the Tagging algorithm compared to trainall, a representative firing matrix (indicated by the star) resolved by the gender of the speaker for digit 'seven' in panel (f). The two additional neurons did not seem to contribute much compared to the two neuron example (panel (f) of Figure 5.2). (g) KL divergence for the possible specialization cue 'speaker identity'. All digits showed clearly reduced realizations for Tagging compared to trainall. A representative firing matrix for digit 'six' resolved by the speaker identity within the target class is shown in panel (h), indicating a more diverse structure for female voices. (i) Possible specialization by spreading decisions in time was analyzed by means of the average difference of output spike times between the fastest and slowest neuron. No clear differences were found, a representative latency histogram of digit 'eight' for comparison with the previous figure is depicted in panel (j).

6

EXTENSION TO PERCEPTRONS

Rosenblatt's perceptron was one of the first models of synaptic learning within neurons (Rosenblatt 1958). Even though nowadays more sophisticated neuron models dominate the field of computational neuroscience, the perceptron embodies basic principles of neuronal processing. Additionally, it has the advantage that it is analytically well described. Showing that the Tagging algorithm also induces specialization when it is based on a population of perceptron neurons would highlight the fundamental benefit of weakly selective training. In this chapter, we demonstrate exemplarily for the XOR problem that indeed Tagging breaks the symmetry of also perceptron learners and thereby provides a solution to the non-separable problem that is not solvable by single perceptrons.

Applying Tagging to a population of perceptrons offers the possibility of direct comparison with the basic mixture of experts model. We illustrate the qualitatively different solutions on the XOR problem for both methods and show that their classification performances on the MNIST handwritten digit data set are comparable.

6.1 TAGGING FOR PERCEPTRONS

The Tagging algorithm that we introduced in Chapter 3 is not restricted to a population of tempotrons. In fact, the weakly selective population learning can be applied to any neuron model that utilizes an internal continuous variable to generate the neuron's binary response. For the integrate-and-fire neurons, this variable corresponds to the maximum membrane potential within the stimulus time.
Similarly, the field $h = \boldsymbol{x} \cdot \boldsymbol{w}$ of a McCulloch-Pitts neuron provides an internal measure of the robustness of the spiking response in perceptron learning. If it is close to zero, a small perturbation in the input can change the sign of the field and hence the binary output of the neuron. Direct transfer of the global Tagging rule with the field as internal state yields the (global) Tagging algorithm for perceptrons:

1. For an input pattern $\boldsymbol{x}$, calculate the individual responses $o_j = \Theta(h_j)$ for every perceptron j. The population response is $\hat{\ell} = 1$ if the number of active neurons is larger of equal to the decision threshold d, and $\hat{\ell} = -1$ otherwise[1].

2. If the population output does not coincide with the input label, update the weights of the perceptron with the d-largest field $h_{(d)}$ by the perceptron learning rule $\Delta \boldsymbol{w} = \ell \boldsymbol{x}$[2].

1 We changed the labeling to $\ell \in \{-1, 1\}$ with $\ell = -1$ for null patterns to allow for a direct application of the perceptron learning rule.

2 Here, we use an additional weight w_0 with a fixed input coordinate x_0 to model the decision bias for each individual perceptron as already described in 2.1.

In addition to these basic rules, we reactivate silent neurons by preferential training on miss patterns. Silent neurons are defined in a similar manner to the tempotron learning as those that have not evoked a positive spiking response for any of the previous 1000 patterns.

Discussed in detail in the appendix (Section A.5), the Tagging algorithm suffers convergence problems if L^2 norms[3] of the synaptic weights between neurons in a Tagging population are highly discrepant. These diverging weight norms across neurons are relatively specific to the perceptron learning rule since updates that do not lead to an immediate error correction always reduce this norm (for calculations see Section A.5). For Tagging, if the neurons' weights and hence their fields are not on the same scale, direct comparison between different neuron's fields on error trials gets biased. Hence, we additionally introduce a normalized Tagging variant. Herein, after each learning trial we normalize all weights such that the new weights' L^2 norm is the same for all neurons. For simplicity, we chose the average norm of all perceptrons in the population before normalization as the final norm for all neurons. This procedure keeps the general relations still intact but can be interpreted as an additional reactivation procedure for neurons with almost vanishing weight norms. Since we observed these problems of large discrepancy between weight norms only for low-dimensional input spaces, we used the normalized Tagging variant only in the following Section 6.2, but not for the classification of handwritten digits (Section 6.3).

6.2 SOLVING THE XOR PROBLEM

A classical problem to demonstrate the limitations of the perceptron is the XOR problem. In a two-dimensional square defined by the coordinates

$$(x_1, x_2) \in \{(-1,-1), (-1,1), (1,-1), (1,1)\},$$

the edges are labeled with $\ell = 1$ if one of their coordinates is 1, but not both (hence 'exclusive or'). As easily seen in a graphical representation (see also Figure 6.1), this set of four observations is not linearly separable and hence not solvable for a single-layer perceptron. A solution becomes available if a hidden layer is incorporated into the perceptron. However, already for this relatively simple 2-layer-perceptron, convergence may require remarkably many learning epochs (hundreds of repetitions through the pattern set, according to Hertz et al. (1991), p. 131).

We addressed the XOR problem with two perceptrons operating on the Tagging algorithm. It can be easily verified that a solution is provided for $d = 1$ when one perceptron fires for solely one target corner and the other for the other target. But does the Tagging algorithm find this solution? We could prove that if the two target pattern are orthogonal to each other in the sense that the zeroth bias coordinate is chosen as $x_0 = \sqrt{2}$ (yielding $-1 - 1 + (\sqrt{2})^2 = 0$), the Tagging algorithm converges to a solu-

3 For a vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ the L^2 norm is defined as $||x|| = \sqrt{\sum_{i=1}^{n} x_i^2}$.

tion to this simple XOR problem after a finite number of steps (see Section A.5 in the appendix). We also empirically tested this for different learning rates η and compared the number of learning iterations that it took the algorithm to find a solution between Tagging and the basic state-of-the-art mixture of experts learning (see Section 2.2). The required learning iterations decreased for larger learning rates (shown in panel (c) of Figure 6.1 as median instead of the mean due to clearly non-symmetrical deviations). However, large envelopes (denoting the empirical 20% and 80% quantiles) indicate a non-negligible set of outliers. Indeed, for the original Tagging, 30 of the 1000 simulations did not converge within the evaluated 10000 learning epochs (of the shuffled four patterns each). These simulations suffered from highly discrepant weight norms among the population. Introducing a normalization of the weight norms to the Tagging algorithm indeed led to convergence of all seeds within the simulated time window. Regardless of the actual Tagging variant, the median number of required learning iterations was comparable to that obtained for the basic mixture of experts model. However, the solutions were qualitatively different (panel (a) and (b) show exemplary solutions for the Tagging and ME algorithm, respectively). While a solution for the Tagging algorithm is only possible if one perceptron fires for solely one target pattern each, the mixture of experts model is more flexible due to the additional gating structure. Depending on the relative size of the scalar products with the gating weights, either one or the other expert (green and red solid lines in Figure 6.1, panel (b)) determines the population decision within an input region (color coded here by the intensity of green to red color). In the exemplary solution visualized in panel (b) of Figure 6.1, even though the expert neurons fired also for one null pattern each, this did not affect the population decision since these patterns belonged to the responsible region of the respective other neuron.

While the solution for the original XOR problem was easily found by both algorithms, we raised the difficulty by extending the XOR problem in the two dimensional space. For $m \in N$, we used $2 \cdot m$ patterns with x_1-coordinates $\{0-(m-1)/2, 1-(m-1)/2, \cdots, (m-1)-(m-1)/2\}$ (each for both patterns of a pair) and alternating x_2-coordinates of -0.5 and 0.5. This parametrization was used for convenience to align the pattern center to the origin and ensure a distance of each neighboring points of one. The zeroth coordinate was fixed as $x_0 = -1$ for all patterns. Labels of the patterns were alternating as for the original XOR problem (see also Figure 6.2 for $m = 4$). The so extended XOR problem is solvable for a Tagging population of m neurons with $d = \lceil \frac{m}{2} \rceil$ if the corresponding hyperplanes line up in parallel with a slope of one and a distance of one to each other (see panel (a), Figure 6.2). However, also less obvious solutions exist for other specific m and d constellations, as shown in panel (b) for $m = 4$ and $d = 3$. Similarly as in the original XOR problem, the mixture of experts algorithm found conceptually different solutions with its specific gating structure (panel (c) for $m = 4$). In contrast to the setting with $m = 2$, however, convergence was not guaranteed for the Tagging algorithm. The more pattern pairs we added to the setting, the fewer simulations converged. Here, for more than three template pairs, also the normalized variant failed to obtain a solution within the $lmax = 10000$ learning epochs in the majority of cases (Figure 6.2, panel (d)). However, despite its larger flexibility,

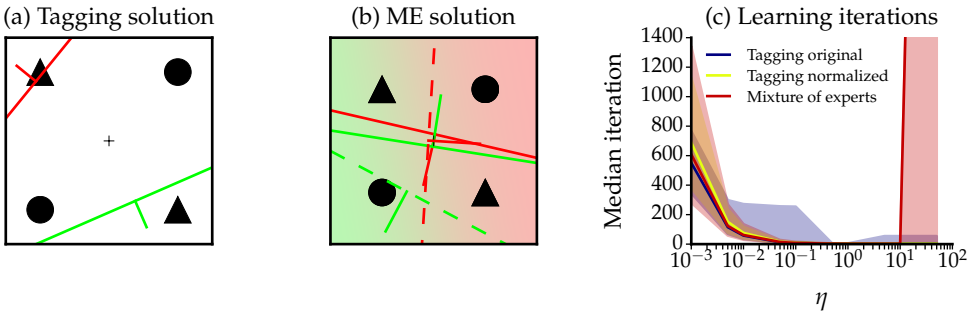

Figure 6.1.: SOLVING THE XOR PROBLEM WITH TAGGING FOR PERCEPTRONS. For the XOR problem, two target patterns with two-dimensional coordinates (1,-1),(-1,1) (triangulars in panels (a) and (b)) should be distinguished from two null patterns with coordinates (-1,-1),(1,1) (circles in panels (a) and (b)). This problem is non-separable and hence not solvable for single perceptrons, but can be solved by Tagging ($d = 1$) and mixture of experts with two (expert) neurons each. (a) Representative solution of a Tagging population. The green and the red line represent the hyperplanes corresponding to $\boldsymbol{w} \cdot \boldsymbol{x} = 0$ that split the hyperspace in firing/non-firing half-spaces. The nose orthogonal to the plane denotes the direction of firing. Both neurons specialized to fire for one target triangle each. (b) Representative solution of a mixture of experts population. The solid lines represent the expert hyperplanes $\boldsymbol{w} \cdot \boldsymbol{x} = 0$ (which with the sigmoidal activation function corresponds to a firing probability of 50%), the dashed lines the corresponding gating hyperplanes $\boldsymbol{v} \cdot \boldsymbol{x} = 0$. Since for the gating neurons, only the relative scalar products are relevant for classification, the background color depicts the responsibilities $g_j = \frac{\exp(\boldsymbol{v_j} \cdot \boldsymbol{x})}{\sum_{l=1}^{m} \exp(\boldsymbol{v_l} \cdot \boldsymbol{x})}$. The combination of gating and expert weights allows for fundamentally different solutions: In this example, both expert neurons fired for one target and one null pattern each, but the gating neurons guided the null patterns to the other neuron, respectively, to avoid misclassification by the population. (c) Number of learning iterations (with all shuffled four patterns each) until perfect classification was achieved, dependent on the learning rate η. Both Tagging variants and the mixture of experts model showed similar median required learning times (solid lines). Colored envelopes denoting the empirical 20% and 80% quantiles, however, showed higher variability for the mixture of experts for small, and for the original Tagging for intermediate learning rates. For too large learning rates, the mixture of experts model failed to provide a solution before weights grew too large.

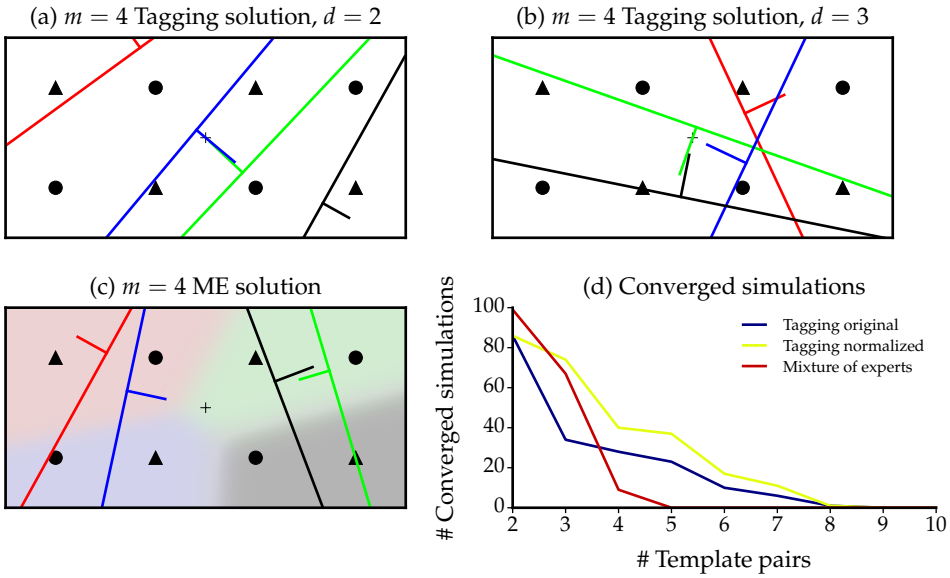

Figure 6.2.: TACKLING THE EXTENDED XOR PROBLEM. For the extended XOR problem, m pairs of alternating target and null patterns were aligned on the x_1-axis (with their coordinates centered around zero, represented by the cross). For m pairs of patterns, we trained m (expert) neurons in Tagging and ME populations with $d = \lceil \frac{m}{2} \rceil$ for Tagging. (a) Representative solution of a Tagging population with $d = 2$ on a four pair problem. Perceptron hyperplanes aligned in parallel, such that a solution was provided by every target pattern being detected by exactly two neurons. (b) Representative solution of a Tagging population with $d = 3$ on the same problem. It is easy to validate that for $d = 3$ a similar solution as in (a) exist with parallel hyperplanes of slope -1. The exemplary solution shown here is not as intuitive but also provided firing for every target pattern by exactly three neurons and for every null pattern by only two neurons. (c) Representative solution of a mixture of experts model. For better visibility, we omitted the gating hyperplanes. The color depth in the background again determines the responsibility of the corresponding expert, again providing a structurally different solution than obtained with Tagging. (d) Number of converged solutions (out of 100) within $lmax = 10000$ learning epochs for all three methods depending on the number of pattern pairs.

the mixture of experts model appeared to have similar difficulties. For more than four pattern pairs it even failed to find a solution for any simulation within 10000 iterations. Hence, judging from these simulations, neither of both models appeared to be clearly superior to the other.

6.3 HANDWRITTEN DIGIT RECOGNITION

A classical data set to evaluate the performance of a machine learning method is the MNIST data set of handwritten digits. Highly sophisticated procedures compete in the complex task to correctly predict the identity of a digit on the basis of a 28×28 pixel gray-scale handwriting image. The MNIST data set is freely available[4] and consists of a training data set of 60000 images as well as 10000 test images from approximately

4 From: http://yann.lecun.com/exdb/mnist/

250 writers with a corresponding labeling. The digits have been size-normalized and centered in a fixed-size image. Correct identification of a pattern among the ten classes of possible digits was achieved with a test error rate of up to 0.56% for support vector machines and even 0.23% for 35 convolutional nets[5]. Even though the Tagging algorithm principally yields binary decisions, we could evaluate the Tagging's multi-class performance by using standard multiclass approaches as one-against-all or one-against-one (Ou and Murphey 2007, see also specifically on handwritten digit problems Milgram et al. 2006). Therein, the decision of several classifier (populations) responsible for different subproblems are combined to yield the final multi-class outcome. However, we were not interested in a direct error comparison with highly complex state-of-the-art machine learning procedures. Our focus was again rather on the possibility of specialization within the target class and an accompanying classification improvement. Therefore, we sticked to binary classification problems and proceeded in a similar manner to the auditory speech processing analysis (Chapter 5).

We started with the again relatively arbitrary classification problem of distinguishing even from odd (handwritten) digits. Similarly to the spoken digits, we expected to gain improved classification performance by specialization on the individual digits in the target class. We trained five perceptrons on the training data set and evaluated the test error after $lmax = 10000$ iterations through the shuffled training data set. The unselective training based on the naive trainall procedure did not yield any classification improvement over a single neuron for neither $d = 1$ nor the majority voting with $d = 3$[6]. However, the misclassification error was reduced to 40% of the single neuron error for Tagging with $d = 17$[7]. For this odd against even classification problem, the mixture of experts algorithm showed to be slightly superior with a significant performance improvement on the 99% confidence level (Figure 6.3, panel (c)). Typical firing matrices indicated specialization within the subclass structure in Tagging populations (panel (a)). The trainall approach unexpectedly also showed selective firing for specific digits - however, to the same digit 'six' for most neurons (panel (b)). This mirrored especially in a high redundancy[8] - and probably the unnaturally high classification error (panels (c)-(e)).

As already suspected by the qualitatively different solutions on the XOR problem, the mixture of experts model followed a different approach than the Tagging algorithm. Here, high classification performance was achieved almost exclusively by the gating neurons; representative firing matrices for expert neurons in ME populations showed constant firing for either all or no digits, with firing probabilities of 100% for

5 Classification performance results are summarized on http://yann.lecun.com/exdb/mnist/

6 But at least for the majority voting we did not detect a significant impairment either, as confirmed by the two-sample t-test.

7 As for the spoken digit recognition, we chose to represent the relative error due to relatively small overall error rates and better comparison among differently difficult digits in the following simulation.

8 As for the even-against-odd classification task in Section 5.2, redundancy was measured on the substructure of all digits $\{0, \ldots, 9\}$. However, in contrast to the spoken digit recognition, we also needed to use this representation of individual digits for calculations of the redundancy in the individual digit recognition task due to the lack of additional substructure information.

active neurons. This extreme firing origined from very high or very low expert weights that drove the activation functions to saturation. This strategy precluded a reasonable calculation of the specialization measures on the basis of the experts structure. Hence, specialization measures are only depicted for Tagging and trainall results, confirming the sharper specialization for selective training in Tagging (panel (d) to (g)).

For the spoken digit recognition, we hypothesized that the gender of the speaker served as specialization cue that helped to improve classification performance. This assumption was based on several acoustical studies that suggested systematical spectral differences between male and female voices (see Section 5.1). In fact, gender differences also seem to exist in handwriting styles and recent studies aim at predicting the gender or other writer-specific properties with automatic machine learning approaches. For example Tomai et al. (2004) deduced the writer's gender from isolated letters with accuracy up to 70%, more advanced studies dealt with connected texts (Maadeed and Hassaine 2014) and online and offline features (Sesa-Nogueras et al. 2015). However, we did not focus on a potential specialization on gender here, partly because the MNIST data set does not provide information about the gender of the writers. Nevertheless, thanks to the simplicity of the perceptron neuron model, the synaptic weights after training offer an interpretation as the receptive field of the perceptron. Visualizing the resulting weights might give an indication as to which handwriting style the corresponding neuron was tuned to.

We trained a population of three perceptrons for each target digit on $lmax = 3000$ iterations through the training set each. Apart from the Tagging algorithm with three different decision thresholds, we evaluated classification performance on the test set for the ME method and an individual neuron. Both population approaches confirmed to be superior to the individual neuron across all individual digit detection problems (Figure 6.4). For some target digits, the ME model showed a tendency for lower errors, for other digits the Tagging algorithm did. Within the Tagging algorithm, the simulations with different decision thresholds were comparable with no clear overall preference. Differences between specialization strategies for different decision thresholds were reflected in the specialization measures. Decision thresholds of $d = 1$ required low response correlations within the target class, whereas positive correlations in the null class did not affect the population error and hence could be tolerated. For $d = 3$, this situation was reversed. Specialization on the null class with $d = 3$ was also accompanied by lower redundancy values than for the small decision thresholds. A possible explanation might be that for this specific problem the null class that contained all remaining digits was clearly more abundant and also more diverse than the target class and hence specialization therein allowed for less redundant responses.

As noted above, the synaptic weights after learning provide information about the preferred stimulus. Highly positive inputs very likely elicit a firing response when they project onto synapses with large positive weights. Similarly, largely negative inputs needs to be coupled with negative synaptic weights to activate the neuron. Hence, every synaptic weight directly represents the preferred gray-scale value in an image

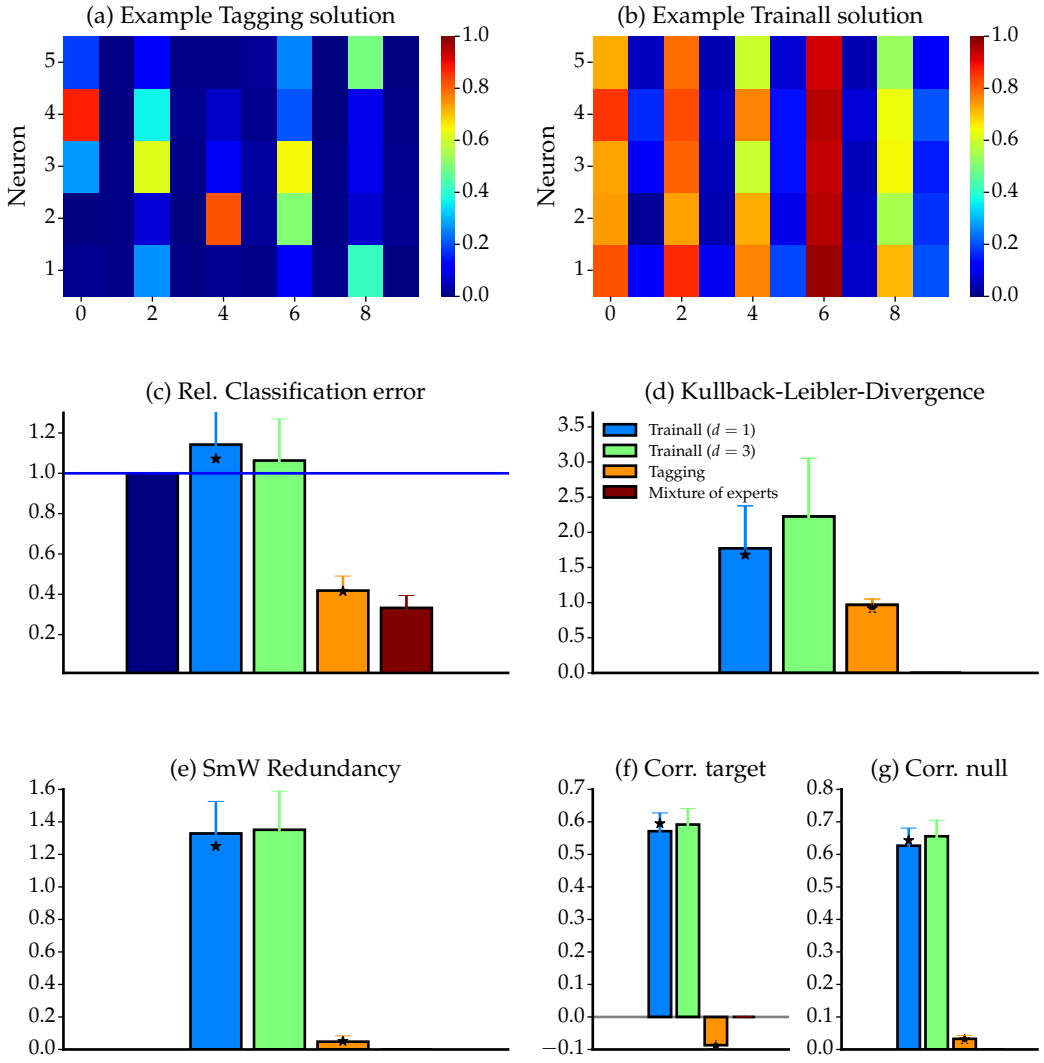

Figure 6.3.: HANDWRITTEN DIGIT CLASSIFICATION - ODD VERSUS EVEN. A single perceptron and five perceptrons within Tagging ($d = 1$), trainall and mixture of experts populations were trained to distinguish even (target) from odd (null) handwritten digits. (a)-(b) Representative firing matrices for Tagging and trainall ($d = 1$). Their specialization measure realizations are indicated by stars in the following panels. (c) Classification error relative to single neuron error. (d)-(g) Specialization measures, in analogy to Figure 5.1. Trainall populations showed very high correlations and redundancy. Values for the mixture of experts could not be obtained in a similar fashion due to the expert/gating structure that yielded extremely large expert weights and are hence omitted. All values were calculated on the test set.

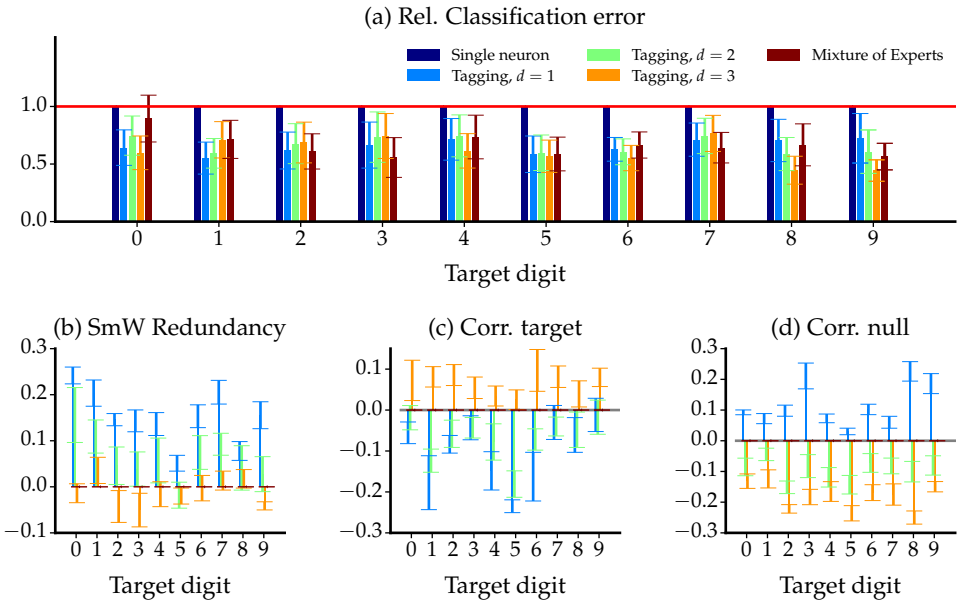

Figure 6.4.: HANDWRITTEN DIGIT RECOGNITION: INDIVIDUAL DIGITS. A single perceptron and three perceptrons within a Tagging population (with $d = 1, 2, 3$) and ME model were trained to distinguish a certain target digit from all remaining digits. (a) Classification error of the population variants relative to the single neuron classification for the different target populations. (b)-(d) Redundancy (measured on the individual-digit-substructure) and correlation within target and null class for all three Tagging populations. Low decision threshold focused on negatively correlated firing within the target class (blue bars), high decision threshold on negatively correlated firing within the null class (orange bars). The KL divergence could not be evaluated due to the lack of known discrete substructures within the target class. All values were calculated on the test set.

that evokes the largest field in the perceptron. For every target population, we illustrated the preferred stimuli of the three neurons of a representative Tagging population with $d = 1$ in Figure 6.5. Most of these fields looked relatively diffuse; however for example for the digits 'one', 'six' and 'seven', clear preferences for differently twisted versions of the corresponding handwritten digit could be observed. These findings emphasize the ability of the Tagging algorithm to uncover hidden structure in the input data also for populations of perceptrons.

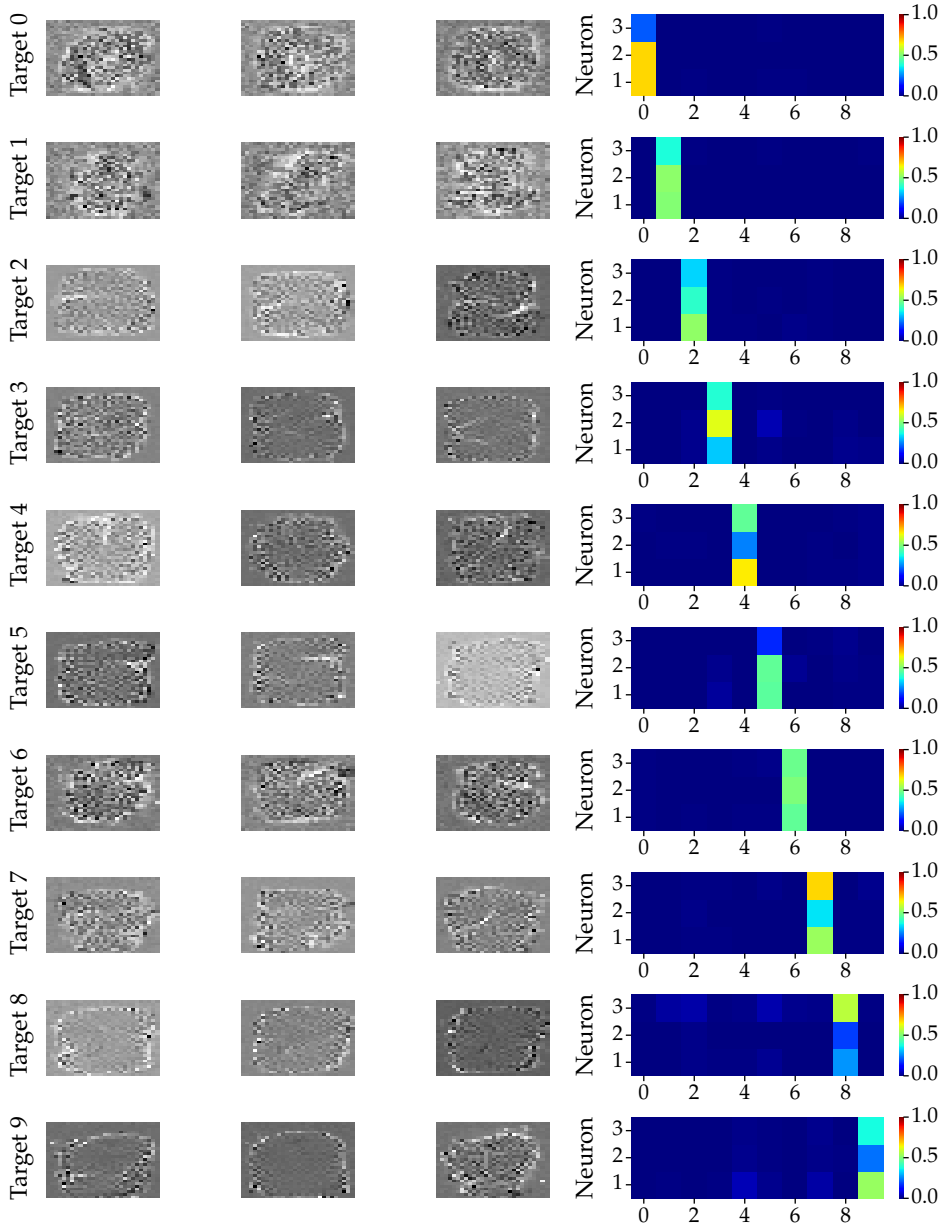

Figure 6.5.: HANDWRITTEN DIGIT RECOGNITION: RECEPTIVE FIELDS. Receptive fields of a representative Tagging population for one target population each (row) with decision threshold $d = 1$. All three neurons learned to fire for the specific target digit with at least 20% probability (firing matrices on the right panels). Here, sometimes clear specialization on different handwritings were observable via the receptive fields, as for example differently skewed 'one's, 'four's and 'seven's.

7

EXTENSION TO REINFORCEMENT NEURONS

In 2009, Urbanczik and Senn proposed a population learning mechanism for stochastic neurons that fulfills all the previously defined requirements on a biologically plausible population. The training algorithm scales the synaptic updates of every erroneously behaving neuron with a factor that depends on the number of correctly responding neurons in the population. The more neurons are correct, the weaker the learning gets. In contrast to trainall and Tagging, learning proceeds even though the population decision based on majority voting is already correct and thereby provides a kind of security margin to the population decision.
We have already evaluated the performance of this attenuated learning in populations of tempotrons in Chapter 4 and showed that the Tagging algorithm yields superior classification performance when the classification problem favored specialist solutions. However, the population margin in the attenuated learning procedure might be especially beneficial for neurons with stochastic outcomes, as for the escape noise neurons (Section 2.1.1) that this approach was constructed for. In this chapter, we implement the escape noise neuron together with its reinforcement learning updates and thereby address two major issues at the same time: (A) Use the attenuated learning population approach of Urbanczik and Senn (2009) in its original environment for fairer comparisons and (B) evaluate if the Tagging algorithm yields also reasonable results with stochastic neurons.

7.1 TAGGING WITH ESCAPE NOISE NEURONS

As also shown in the previous chapter, the Tagging algorithm is generally independent of the actual response generation and synaptic learning mechanism of the individual neurons in the population. The only demand it places on the neuron is that its binary response correlates with an internal continuous variable that can be used to select the subset of neurons that undergo learning. Here, we apply Tagging to a population of escape noise neurons that also rely on integrate-and-fire neurons, but differ from the tempotron model in two major points: First, neurons generate their spiking response stochastically, with a firing probability that is positively correlated with the size of the current membrane potential. Second, learning is induced on the basis of an eligibility trace, following the idea that in case of an erroneous decision the probability to elicit exactly the same output spike train should be reduced (Section 2.1.2).

In order to apply Tagging to a population of escape noise neurons, the first necessary step was to implement the neuron model. During the implementation process, however, we uncovered two major problems. First of all, reinforcement learning for this model turned out to be instable. For a substantial number of simulations, the synaptic weights of individual neurons in the population diverged without a limit.

Secondly, the escape noise neuron showed problems with unbalanced target and null classes which is a requirement for specialization. Whereas the second problem can be solved easily by including a normalized Tagging variant as we describe below, the first problem may have a more profound influence on the applicability of the neuron model itself. Hence, we will describe the weight divergence problem that strongly depends on the actual implementation of the firing process and the eligibility trace in greater detail in the last section of this chapter. Therein, we also provide a possible explanation of this divergence phenomenon

Tagging for Escape Noise Neurons

For the escape noise neuron, spiking does not occur deterministically as soon as the membrane potential crosses a firing threshold, but stochastically with a voltage-dependent firing intensity. Nevertheless, the maximum of this membrane potential trace still provides a reasonable correlate for the firing behavior. The higher it is, the more likely the neuron elicits as spike. Hence, we used the V_{max} as internal variable in analogy to the (local) tempotron Tagging algorithm. In case of a miss, if a neuron's maximum membrane potential exceeds a training threshold v_{train}, the neuron adjusts its weights on the basis of an eligibility trace.
With the originally used exponential firing intensity function $\phi(V(t)) = \kappa e^{\beta V(t)}$, the hypothetical firing threshold is $\vartheta = 0$ for infinitely large β. Hence, the previous choice of $v_{train} = 0.8$ for the starting thresholds is unreasonable, and the training thresholds were initialized with $v_{train} = -0.5$ instead. During learning, the training thresholds adjust according to the same rule with the same parameters as for the tempotron Tagging. For false positives, all neurons modify their weights as before.

When training a Tagging population with a small decision threshold to distinguish a certain number of poisson patterns as explored by Urbanczik and Senn (2009), we expect to obtain specialization within the target class. As discussed also in Section 4.3, for the individual neuron this relates to a subproblem where the probability to observe a target input is lower than the occurrence probability of null patterns. As already mentioned above, simulations with individual neurons, however, revealed that the escape noise neuron has problems solving classification tasks with unbalanced classes. For rarely occurring target patterns, the neuron failed to establish a consistent firing behavior in simulations even after long learning time (see Figure 7.1, blue curve in panel (a) and (b)). The failure to spike for rarely occurring patterns could be explained by very weak LTP steps compared to the LTD steps[1] (panel (c)). These weak LTP steps were probably due to very small synaptic weights (many depressing null patterns) and hence almost vanishing firing probabilities ϕ that determined the strength of an LTP step. Since the ability to fire for seldom patterns is crucial for specialization in the Tagging population, we slightly modified the learning algorithm to make up for this weakness. At each learning step, we normalize the size of the weight updates so that the L^2-norm of each weight update is fixed to a value $\tilde{\eta}$. This parameter $\tilde{\eta}$ directly takes over the role of the learning rate η of the original algorithm.

1 The strength of an LTD/LTP step is defined as the L^2 norm of the weight update, i.e. $\sqrt{\sum_{i=1}^{n} \Delta w_i^2}$.

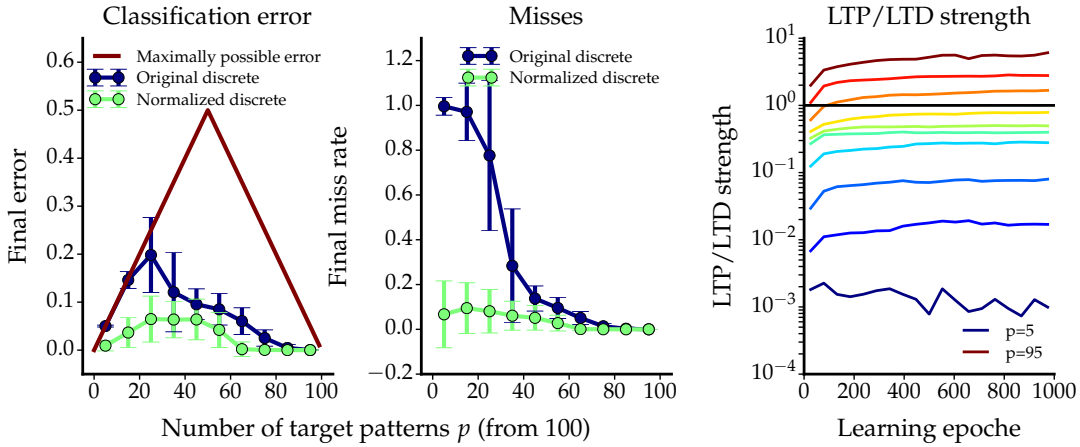

Figure 7.1.: SOLVING CLASSIFICATION PROBLEMS WITH PATTERN CLASS IMBALANCES. (a) Final classification error and (b) Miss rate dependent on number of target patterns (of $p = 100$) for the original and the normalized reinforcement learning. When trained with the original weight updates, the neurons remained silent when the target patterns occurred rarely. (c) Ratio of LTD/LTP strengths (L^2-norm) depended strongly on the number of target patterns. The fewer patterns the target class contains, the stronger the LTD step got compared to the LTP over learning time (color code denoting the number of target patterns out of 100 total patterns). A possible explanation is that for low target probabilities more LTD steps were performed and the weights got more negative so that the factor ϕ that determined the size of the LTP step was relatively small. For more patterns in the target than in the null class, the LTP predominated.

This modification enabled the neuron to establish firing for rare target patterns (green curve in Figure 7.1, panel (a) and (b)). Therefore, we incorporated this normalized training in the Tagging algorithm, even though it requires biologically implausible knowledge of the weight updates from other synapses.

Similar to the classification problem from Urbanczik and Senn (2009), we evaluated the Tagging and the attenuated learning approach on the task to discriminate between two sets of a fixed number of poisson patterns. Poisson patterns realized input spikes within a 500 ms time window with a mean firing rate of 6 Hz and were randomly split into target and null class. However, in contrast to the original simulations of Urbanczik and Senn (2009), the classes did not consist of only 15 patterns each (making 30 patterns in total), but 50 patterns each. The reason was that for the problem with 30 patterns, already a single neuron was able to discriminate both classes with an error rate smaller than 1% after sufficiently many learning cycles. The comparisons Urbanczik and Senn made to establish their attenuated learning procedure were based on the speed of different population algorithms. However, we do not claim any statement about the speed of Tagging and are more interested in classification problems that are too complicated to be solved by single neurons and hence require division of the task by different population members. Consequently, the target class as well as the null class contained 50 patterns each, providing a more complicated task with non-vansihing error for single neurons even after long training time. The individual neurons received inputs from roughly 40 of the 50 presynaptic spike trains by randomly drawing a connection from each input synapse with 80% probability, in

analogy to the studies by Urbanczik and Senn (2009).

We trained populations of $m = 3$ neurons to distinguish the 100 poisson patterns with the Tagging algorithm with decision thresholds $d = 1, 2, 3$ and normalized weight updates. Additionally, we set up a population with attenuated training once in its original variant and once with the normalized weight updates to ensure comparability. Learning rates for the normalized versions were varied as $\tilde{\eta} = [2, 10, 25, 50]$ and for the original weighted learning algorithm as $\eta = [500, 1000, 2500, 5000]$ (Urbanczik and Senn (2009) used $\eta = 2500$ for the attenuated learning). For the optimal learning rates of $\tilde{\eta} = 25$ (for global Tagging and local Tagging with $d = 1$) and $\tilde{\eta} = 10$ (for local Tagging with $d = 2, 3$), the Tagging populations all yielded lower classification errors than both variants of the attenuated learning with optimal learning rates $\eta = 2500$ (original learning) and $\tilde{\eta} = 10$ (normalized learning) (Figure 7.2). Additionally, specialization within the 100 patterns emerged for all decision thresholds. Whereas for $d = 1$ the neurons tended to specialize to fire for subgroups in the target class, for $d = 3$ specialization was observed in the null class. For this problem, specialization appeared to be more beneficial in the null class than in the target class, as training with $d = 1$ yielded significantly worse results than with the higher decision thresholds. This preference for higher thresholds might be related to the fact that the individual neurons with the original learning procedure had problems to detect rarely occurring target patterns.

Even though we only investigated a very selective classification problem for the neurons with stochastically fluctuating threshold, these results indicate that the Tagging algorithm is not only capable of inducing specialization within other neuronal models, but also that its performance is comparable - or even superior - to a recently published population approach.

7.2 INSTABILITY PROBLEMS

As already mentioned in the previous section, learning within the escape noise neuron showed to be instable. Unlike in the supervised tempotron learning, the reinforcement reward to a single neuron does not explicitly indicate the direction of the error: reducing the likelihood of generating an output spike train with one output spike could be achieved by stopping firing the next time, but also by generating more than one output spike. For correction of a null pattern, the second option would push the neuron further away from the correct response of zero spikes. This leads to an increase of synaptic weights in a situation where a decrease would be necessary to correct the response. However, theoretically, these weight changes into the wrong direction should be averaged out since the expected number of spikes and the realized number of spikes that together determine the direction of the weight change should agree on average. Specifically, Fremaux et al. (2010) pointed out that when the learning rule is used in an unsupervised manner without any reward feedback, the weight change should be zero on average due to this relation. However, their statement relies on the

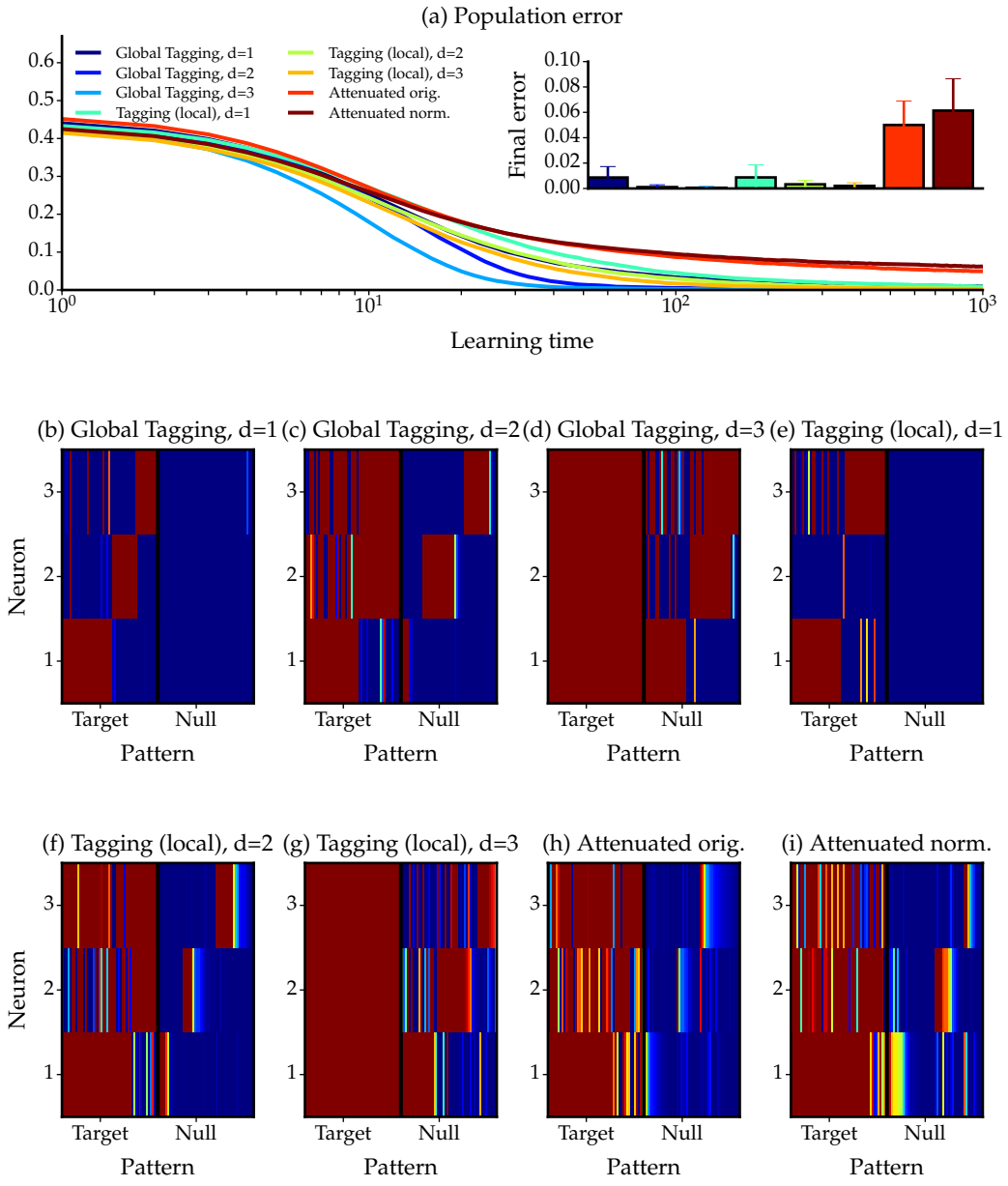

Figure 7.2.: TAGGING FOR ESCAPE NOISE NEURONS. (a) Population error for the Tagging algorithm with normalized weight updates for different decision thresholds and attenuated training with original and normalized weight updates, each for optimized learning rate. In each population, three neurons distinguished 50 target poisson patterns from 50 null poisson patterns. The final classification error in the inset panel is significantly lower for all Tagging populations than for the attenuated training. For the Tagging populations, the maximal decision threshold of $d = 3$ yielded the best results. (b) to (f) Exemplary chosen firing probability matrices for all different population algorithms. Depending on d, the neurons in the Tagging population either specialized to fire for specific patterns in the target or in the null class. For the attenuated training, the specialization was not as clear as for the Tagging, also for $d = 2$.

assumption of independent weight changes. In an extended simulation study of this equation we show that during the learning process, subsequent weight changes are not independent since the standard deviation of the possible weight changes depends on the current weight. As this violates the assumption underlying this equation, we cannot exclude a drift of the weights towards large values. We show that the degree to which we observe divergence in the weight traces depends specifically on how the spike generation process is implemented in the neuron model. The following analysis investigates the above statements in great detail.

7.2.1 *Implementations*

The escape noise neuron that builds the basis for the population approach by Urbanczik and Senn (2009) was originally described by Xie and Seung (2004) and Pfister et al. (2006). Based on a stochastically fluctuating threshold, the neuron generates spikes with a firing intensity function $\phi(u)$ and performs synaptic weight updates on the basis of an eligibility trace E (see also Section 2.1.1). Both, the spike generation process and the explicit representation of E can be implemented in different ways. In the following, we present two implementations that differ mainly in their way to approximate the continuous poissonian firing and to solve the low pass filter of the eligibility trace. We will show in the last paragraph of this section, that even though they approximate the same mathematical process, both implementations show significantly different behavior when we tried to reproduce the basic results from Urbanczik and Senn (2009).

Exact Implementation - Explicit Solution of the ODE

In reinforcement learning approaches, weight updates rely on an eligibility trace that represents a response memory trace of the individual neuron. Specifically, Urbanczik and Senn (2009) use a negative low-pass filtered version of the eligibility trace E to adjust the weights after individual erroneous behavior (derived in Section 2.1.1 by combining equations (2.6) and (2.7)). It is defined for every individual neuron by[2]:

$$\tau_M \frac{dE}{dt} = -E(t) + \underbrace{\beta PSP(t) \left(\sum_{s \in Y_t} \delta(t-s) - \phi(V(t)) \right)}_{\Psi(t)}$$

where $PSP(t)$ denotes the postsynaptic potential (for input synapse i), $V(t)$ the voltage and Y_t the output spike train up to time t. In general, for an ODE of the form

$$\frac{dx}{dt} + g(t)x = f(t)$$

2 For simplicity we drop the synapse index in the following calculations. However, E and PSP are assumed to depend on the individual input synapse.

an explicit solution is given by

$$I(t)x(t) = \int I(t)f(t)dt + C$$

for

$$I(t) = \exp(\int g(t)dt)$$

and a constant C satisfying the initial condition. In this scenario, $g(t) = 1/\tau_M$, $f(t) = \Psi(t)/\tau_M$ and hence $I(t) = \exp(t/\tau_M)$. Therefore, the exact solution of the ODE equals

$$E(T) = \exp(-T/\tau_M)\left(\int \exp(t/\tau_M)\Psi(t)/\tau_M dt + C\right).$$

With $C = 0$ we simplify this expression further to

$$\begin{aligned} E(T) &= \int \exp((t-T)/\tau_M)\frac{\Psi(t)}{\tau_M}dt \\ &= \int \exp((t-T)/\tau_M)\beta/\tau_M PSP(t)\left(\sum_{s\in Y_t}\delta(t-s) - \phi(V(t))\right)dt \\ &= \beta/\tau_M\left(\sum_{s\in Y_T}\exp((s-T)/\tau_M)PSP(s)\right. \\ &\quad \left. - \underbrace{\int_{-\infty}^{T}\exp((t-T)/\tau_M)PSP(t)\phi(V(t))}_{=A}\right). \end{aligned}$$

The integral A needs to be computed numerically. In our implementation, we approximate it by simply summing up the integrand for the discrete time steps.
Due to the exact solution of the differential equation, we will refer to this implementation as the exact variant in the following sections.

Original Implementation - Forward Euler Approximation

Urbanczik and Senn (2009) used a different approach to obtain the weight updates. They followed a discrete forward Euler approximation of the eligibility trace that led to the following equation:

$$\begin{aligned} E(t+\Delta t) &= E(t) + \frac{dE}{dt}\Delta t \\ &= E(t) - \frac{\Delta t}{\tau_M}E(t) + \frac{\Delta t}{\tau_M}\beta PSP(t)\left(\sum_{s\in Y_t}\delta(t-s) - \phi(V(t))\right) \\ &= \left(1 - \frac{\Delta t}{\tau_M}\right)E(t) + \frac{\Delta t}{\tau_M}\beta PSP(t)\left(\sum_{s\in Y_t}\delta(t-s) - \phi(V(t))\right) \end{aligned}$$

1) In case no output spike occurred in the time window $[t - \Delta t, t]$, the above expression can be simplified as

$$E(t + \Delta t) = \left(1 - \frac{\Delta t}{\tau_M}\right) E(t) - \frac{\Delta t}{\tau_M} \beta PSP(t) \phi(V(t)).$$

2) On the other hand, if a spike was observed, this expression becomes

$$E(t + \Delta t) = \left(1 - \frac{\Delta t}{\tau_M}\right) E(t) + \frac{\Delta t}{\tau_M} \beta PSP(t) \left(\delta(x) - \phi(V(t))\right)$$

where $x < \Delta t$. Finally, with the rectangle-approximation for the delta function we get

$$E(t + \Delta t) = \left(1 - \frac{\Delta t}{\tau_M}\right) E(t) + \frac{\Delta t}{\tau_M} \beta PSP(t) \left(\frac{1}{\Delta t} - \phi(V(t))\right) \tag{7.1}$$

$$= \left(1 - \frac{\Delta t}{\tau_M}\right) E(t) + \frac{1}{\tau_M} \beta PSP(t) - \underbrace{\frac{\Delta t}{\tau_M} \beta PSP(t) \phi(V(t))}_{=B}. \tag{7.2}$$

For infinitesimal small time steps Δt, the role of summand B becomes negligible. In the implementation of Urbanczik and Senn (2009), the authors regarded $\Delta t = 0.2$ ms as sufficiently small and hence discarded the subtraction of B for the time intervals in which an output spike was observed (personal communication). However, we will show in the following that smaller choices of Δt - as well as use of the exact solution - lead to instability problems.

7.2.2 *Reproducing the Results of Urbanczik and Senn (2009)*

To compare the exact and the original implementation on a representative classification problem, we reproduced the results of Figure 1 from Urbanczik and Senn (2009) with their proposed population learning variant. The investigated scenario treated $p = 30$ different poisson patterns from which exactly $p_{target} = 15$ were assigned to the target class. The authors trained a group of neurons to distinguish target from null patterns during a learning process where they presented only one randomly chosen pattern in each of $lmax = 2000$ learning iterations. The number of neurons within the population varied from $m = 1$ to $m = 33$. In contrast to the (previous and following) simulations in this thesis, the original paper used an accuracy measure defined as an exponentially decaying filter: $acc(l) = \gamma a(l) + (1 - \gamma) acc(l - 1)$ with $\gamma = 0.2/p$ and $a(l) = 1$ if the lth pattern was classified correctly, and $a(l) = -1$ otherwise. For comparison purposes, we also monitored this accuracy while reproducing the results with the same parameters ($\beta = 5$, $k = 0.01$, $\Delta t = 0.2$, $\eta = 2500$) for both implementations. Besides the proposed attenuated learning procedure with weighted learning rate dependent on the population performance, the authors evaluated population errors also for two simpler population approaches. In the global training, all neurons perform a weight update if the global reward is negative, regardless of individual behavior: $\Delta w_{j,i} = (R - 1) E_{j,i}(T)$ where $R = \pm 1$ defines the global reward with plus and minus

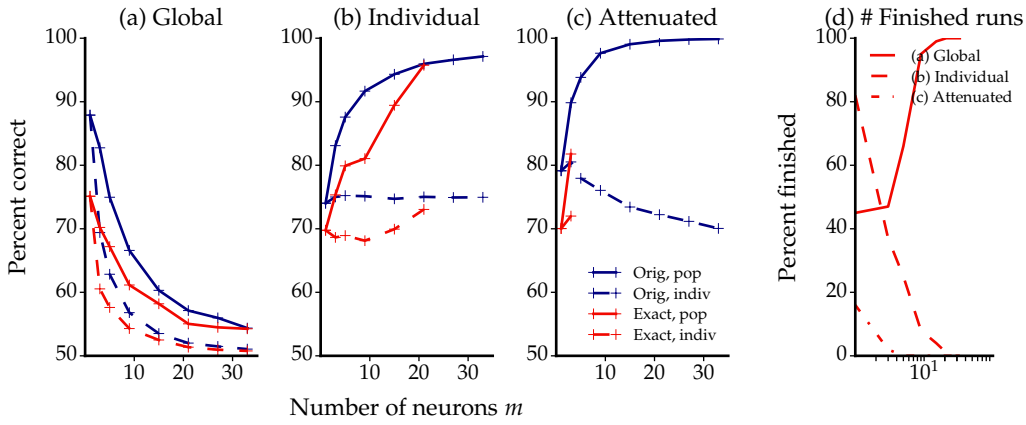

Figure 7.3.: REPRODUCTION OF FIGURE 1 FROM URBANCZIK AND SENN (2009). A population of varying number of neurons were trained with the three different population approaches from Urbanczik and Senn (2009) to distinguish 15 target patterns from 15 null patterns. (a)-(c) Percent correctly classified patterns after $lmax = 5000$ (global) and $lmax = 2000$ (individual and attenuated) presented patterns. Differently colored lines denote the two different implementations of the reinforcement learning of the escape noise neuron. Results from the discrete implementation are comparable with those depicted in the original Figure 1 of Urbanczik and Senn (2009). The exact implementation, however, showed severe problems of diverging synaptic weights, its number of non-diverging simulations depicted in panel (d).

sign for correct and incorrect population responses (based on the usual majority voting), respectively. In contrast, for individual training, only the individual performance determines the learning behavior, $\Delta w_{j,i} = (r_j - 1)E_{j,i}(T)$. We also compared these two simpler population approaches and reproduced the Figure 1 of Urbanczik and Senn (2009) with the same learning rates ($\eta = 625$ for individual learning, $\eta = 1250/m$ for global learning) in Figure 7.3. The values represent the moving average accuracies as described above after $l = 2000$ learning epochs for the individual and attenuated learning and after $l = 5000$ epochs for the global learning, consistent with Urbanczik and Senn (2009).

The accuracies of all 100 different starting seeds with our direct re-implementation of Urbanczik and Senn's neuronal learning were comparable to the original results stated in the paper (blue lines, panel (a) to (c)). However, for the exact implementations, we observed well-behaved learning only in few simulations, their number decreasing with the population size for the individual and attenuated learning (see Figure 7.3, panel (d)). For all other starting seeds, the synaptic efficacies diverged and exceeded values of 200 (at which point we terminated the learning loop), in comparison to moderate simulations with converged weights of size $w \approx 10$. The increase of well-behaved simulations in the global population learning setting is probably due to the decrease of the learning rate for larger populations and hence does not reflect the influence of the population learning itself.

Why does the learning performance depend so strongly on the implementation? The exact variant implements the same phenomenon with the same parameters as the binomial approximation. In fact, for infinitely small values of Δt, the original and the

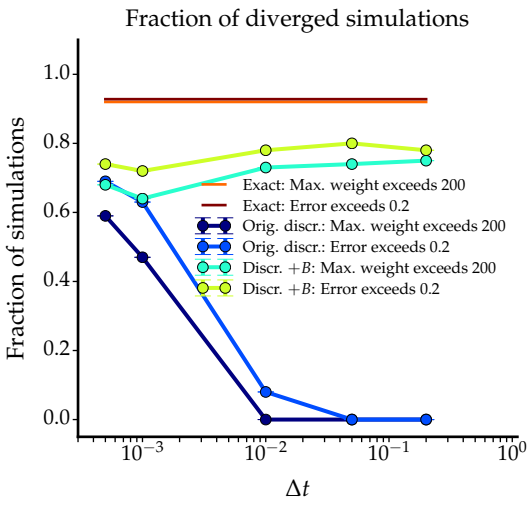

Figure 7.4.: INFLUENCE OF Δt ON LEARNING STABILITY. Fraction of simulations with at least one synaptic weight exceeding $w_{crit} = 200$ for different time steps Δt for the original discrete implementation from Urbanczik and Senn (2009) (dark blue curve) and the discrete implementation including term B from equation (7.2) (turquoise curve). With the exact implementation (for a fixed $\Delta t = 0.2$ for summation of the integral) more than 90% of the simulations diverged (orange benchmark line). For smaller time bins, also the divergence rate of the discrete implementation grows. Diverging weights were accompanied by larger classification errors (dark red, light blue and yellow curve, respectively), indicating bad learning behavior.

exact implementation should be identical.
To verify this identity, we repeated the simulations for a single neuron (for which already the exact variant diverged in more than 80% of the simulations for $\eta = 2500$) with the original discrete implementation for different values of Δt and monitored the number of well-behaving seeds.

Interestingly, with smaller time steps, the risk of weight divergence increased for the original implementation (see Figure 7.4, blue curves). When we additionally included the neglected term B of equation (7.2) to the eligibility trace during learning, we observed the high number of diverging weights already for larger values of Δt (green/yellow traces). What caused these instabilities? According to Urbanczik and Senn (2009) and Pfister et al. (2006), the eligibility trace follows a gradient descent approach and hence should lead to a minimization of the corresponding error function after a finite number of learning steps. We investigate the observed problem of diverging weights, that seems to be inherent in this kind of reinforcement learning, in a clearly simplified paradigm in the following section.

7.2.3 *(Un)biased Learning*

Fremaux et al. (2010) analyzed the learning rule that was used by Urbanczik and Senn (2009) in a more general framework. Here, the reinforcement learning was explicitly divided into an unsupervised learning part and an additional reward-based step which initiates weight updates only if necessary to increase the reward. The R-max rule that corresponds to the unsupervised part of the attenuated learning is defined as

$$UL_j(t) = (Y(t) - \phi(t)) \int_0^\infty \epsilon(s) X_j(t-s) ds$$

for synapse j, where ϕ denotes the firing intensity and $\epsilon(t)$ the PSP kernel function. The output spike train $Y(t)$ at time point t is defined here via the delta function as

$Y(t) = \sum_{s \in Y} \delta(t-s)$ with Y as the set of output spikes as before. Similarly, X_j denotes the input spike train to the *jth* synapse at time t (following the original notations from Fremaux et al. 2010). Combined with the reward-based updates, this rule is exactly the derivative of the log-likelihood function from Urbanczik and Senn (2009) with respect to the weights of synapse j (see equation (2.6)). So it would correspond to the direct (negative) weight update without the low-pass filter that is suggested by Urbanczik and Senn (2009).

For this learning rule, the authors argued that the weights on average should not show any drift if the updates are performed in an unsupervised manner. This statement directly follows from

$$\langle UL_j \rangle_{X,Y} = \left\langle \langle Y - \phi \rangle_{Y|X} \int_0^\infty \epsilon(s) X_j(t-s) ds \right\rangle_X = 0 \tag{7.3}$$

since ϕ is exactly the expectation value of Y and consequently their difference should vanish on average (angle brackets denote averages with respect to the subscripted variables). Hence, the weights should not change over repeated presentations of different arbitrary input patterns. During the actual reinforcement learning process, learning in the error correcting direction is achieved by performing update steps exclusively on negatively rewarded patterns (Fremaux et al. 2010).

In the following subsection, we explore the validity of the above equation for the different implementations of the learning rule in simplified paradigms to gain further insights into the problem of diverging weights obtained in Section 7.2.2.

(Un)biased Learning for Binomial Spike Generation

The previous argument of unbiased learning relies on the basic structure of the likelihood:

$$L \propto (Y(t) - \phi(t)) PSP(t)$$

To simulate development of the weights based on the unbiased learning rule and thereby explore equation (7.3) without any additional complex components, we simplified the above equation: We assumed that we had a fixed set of $\tilde{T}$ discrete time bins $\tilde{t}i$ and the PSPs were constant over all time bins ($PSP(t) = 1 \quad \forall t$). Furthermore, we omitted the reset after potential spike generation and considered only a single synapse. Hence, the learning consisted of

$$\Delta w = -\sum_{s \in Y} 1 + \sum_{\tilde{t}i} \phi(w + V_{rest}) = -\sum_{s \in Y} 1 + \tilde{T}\phi(w + V_{rest})$$

since also $V = V(t)$ does not depend on the time anymore but only on the reset potential and the current weight if the PSPs are constant.

To keep the scenario as close as possible to the simulations from Urbanczik and Senn (2009), each (constant) input stream consisted of $\tilde{T} = 2500$ time bins (correspond-

ing to $\Delta t = 0.2\,\text{ms}$) and the single synaptic weight was initiated with $w = 1.7$. With the original firing rate parameters $k = 0.01$ and $\beta = 5$, we mimicked an unsupervised learning with this simplified model on $p = 100000$ patterns. Weights were updated with a learning rate of $\eta = 0.002$. According to formula (7.3), the weights should follow a random walk without drift, i.e. the weight changes should be zero on average. We simulated learning for 1000 different random seeds. Due to the independent construction of the simulations, the statistic $z = \frac{<\Delta w>_{p_i}}{\sigma\sqrt{p}}$ (with weight changes averaged over all p patterns p_i, $i = 1, \ldots, p$) should be t-distributed. Hence, while individual weight traces could deviate strongly from the base line zero, under the null hypothesis of zero drift only α% of these z-standardized weight changes should either exceed the $(1 - \alpha/2)$% quantile or undershoot the $\alpha/2$% quantile. However, we observed a clearly skewed distribution, with 9.7 percent of the simulated z-realizations falling below the 2.5% or above the 97.5% quantile (second panel of Figure 7.5).

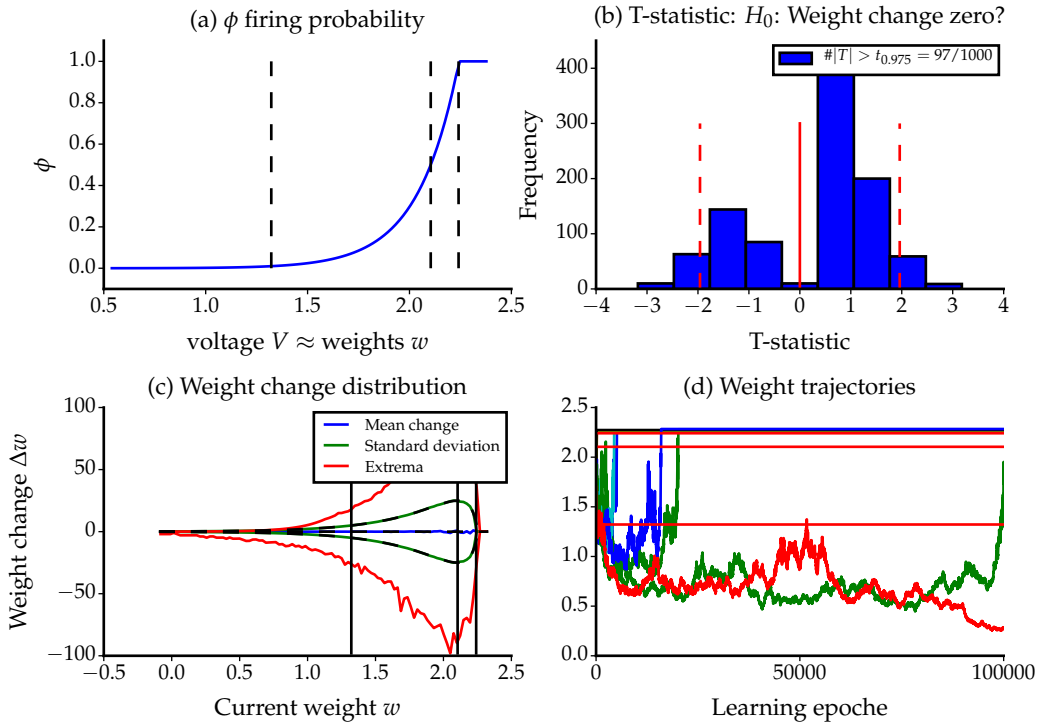

Figure 7.5.: UNSYMMETRICALLY BIASED WEIGHT CHANGES. Spikes were generated with firing probability ϕ as in panel (a) for 100000 constant inputs. Dotted lines represent the weights w_{01}, w_{50}, w_{99}, corresponding to 1%, 50% and 99% firing probability. (b) The probability density of z-statistics calculated for 1000 simulations did not follow a t-distribution as required by the null hypothesis of zero average weight change. (c) Empirical weight change distribution depending on the current weight w. The stronger w deviates from w_{50}, the smaller the standard deviation and hence the potential change gets. Dotted lines represent the theoretical standard deviation of the binomial distribution. Due to the firing probability of $\phi = 1$ for $w \geq 2.2243$, weight changes are zero beyond that value (panel (c)), yielding an absorbant state for the weight trajectories (panel (d)).

Why does this empirical distribution differ so strongly from the expected t-distribution? Does this finding contradict the formula (7.3) that states zero weight changes on

average? To understand this phenomenon, one has to take a closer look at the shape of the firing probability function $\phi(V) = k \cdot \exp(\beta V)$ (see panel (a), Figure 7.5). The larger the synaptic weight, the more closer the firing probability approaches one, hence the more deterministic gets the spiking response. For a weight exceeding $w = 2.2243$, the original ϕ function even crosses the 100% firing probability (in panel (a) truncated), totally eliminating any stochasticity for constant inputs. Whereas also very small synaptic weights approach almost deterministic non-spiking, only in intermediate weight ranges, where ϕ is about 0.5, the outcome of the spiking is hard to predict. Hence, considering a constant firing probability for all $\tilde{T} = 2500$ time bins, the number of output spikes is much more variable for intermediate than for extreme values of ϕ. Large variability in the output spike number is directly accompanied by larger possible deviations of the realized from the expected number of spikes. This in turn goes hand in hand with larger possible weight changes. Hence, the standard deviation of the distribution of possible weight changes for an intermediate w is much larger than for extreme values of w (panel (c), Figure 7.5, black lines in the figure indicate the weights $w1$, $w50$ and $w99$ that correspond to 1%, 50% and 99% of the firing probability, from left to right). The empirically measured standard deviation indeed agrees with the theoretical one for the binomial distribution: $\sqrt{\tilde{p}(1-\tilde{p})\tilde{T}}$, where $\tilde{p}$ equals the firing probability, in this case specifically the realization of ϕ for the current $w + V_{rest}$ (see dashed black line and note the concurring envelopes for $\tilde{p} = 1$). This systematically varying weight change distribution results in the observed weight evolution: If a large learning step towards one of the boarders is performed, the following weight changes will be based on a more tight weight change distribution. Consequently, the following step cannot compensate the previous one, leading to an attraction of the extreme weight values (panel (d), Figure 7.5). More severely, if a weight update yields $w > 2.2243$, the system has reached an absorbing state and does not allow for any further weight change. Together with the small weight changes for already small synaptic efficacies, these effects contributed to the biased two-peaked distribution of the t-statistic.

It is important to point out that not the extreme realizations of w itself cause the bias, but the learning. This can be seen when storing the trajectory of all realized weights w in a vector and perform weight changes based on these weights with new 'input patterns'=random numbers for the spike generation. In contrast to 'real' learning, in the following step we did not actually use the hypothetical new weights $w = w + \eta\Delta w$, but w from the previous simulation. Here, the weight changes for each pattern are independent and hence follow the equation (7.3). Consequently, the corresponding t-statistic to the test of $\Delta w = 0$ was t-distributed with about 5% of the values exceeding the corresponding quantiles as expected (panel (a) of Figure 7.6). This required independence of weight changes is not given for explicit learning. During explicit learning, a weight change at pattern p_i has an influence on the weight change distribution at pattern p_{i+1}. If the weight has been pushed towards a more extreme value due to the change at pattern p_i, the distribution for pattern p_{i+1} will be narrower, hence the previous weight change can not be compensated by Δw at pattern p_{i+1} (see also Figure 7.5

and 7.6). This means that the weight changes at each learning step depend on the previous learning steps since the standard deviation of the weight change distribution depends on the previous history.

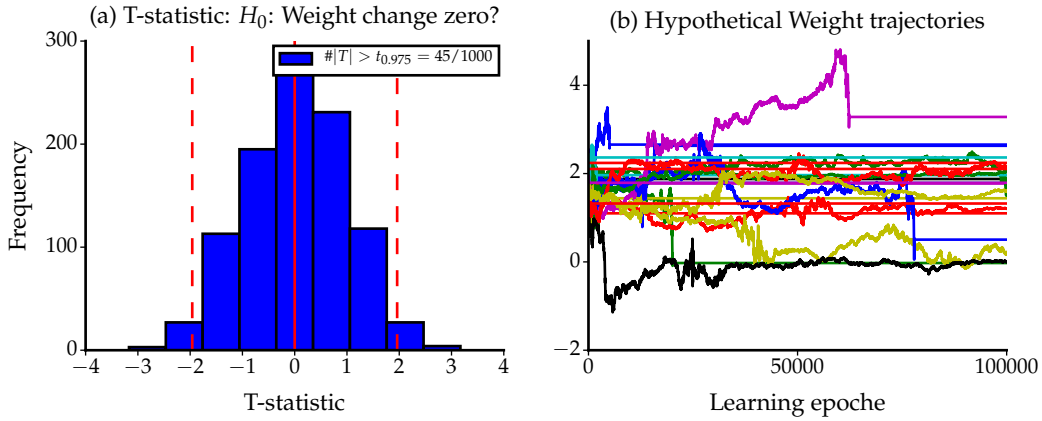

Figure 7.6.: INDEPENDENT UPDATES RESOLVE WEIGHT CHANGE BIAS. When the empirically measures weights from Figure 7.5 were combined with new constant inputs and new spike trials did not depend on the previous updated weights, the zero weight change hypothesis could not be rejected (panel (a)). (b) Hypothetical weight trajectories as cumulative sum of all independent weight updates do not show any deviations from random walks.

Time-varying PSPs

One has to keep in mind that the simulations presented here are simplified versions of the real problem. In real neuronal applications, the PSPs are not constant, but change over time. This is specifically important since also for very large weights the resulting sum of ϕ can be still quite far away from one (since in many time bins the PSP was negligibly small and hence the firing probability low). Consequently, there is much more freedom especially in the weight change distributions for larger weights. Hence, the vanishing tail for large weights as observed in the previous simulations with constant inputs (in panel (c) of Figure 7.5) widens for more realistic input patterns (see panel (a) Figure 7.7). For this simulation, we increased the learning rate to $\eta = 0.1$ to ensure weight changes of comparable sizes[3]. As a 'drawback' of using realistic PSPs, the upper bound of weights that we found in the simplified scenario that was due to deterministic firing is not observable for simulations with real patterns. Hence, we faced the risk of truly diverging weights.

(Un)biased Learning for Poisson Spike Generation

Urbanczik and Senn (2009) approximate the poisson firing process by discrete bernoulli variables in sufficiently small time bins. For low membrane potentials and hence small firing intensities, this procedure is justifiable. However, as shown above, due to the learning-induced correlations we often observe large values of w that make the approximation by bernoulli indefensible. This is especially the case for very large

3 The amplitude of the PSPs in the constant simulation was fixed to $PSP = 1$, for the time varying PSP we went back to the original parameters of Urbanczik and Senn (2009), yielding an amplitude of only 0.07.

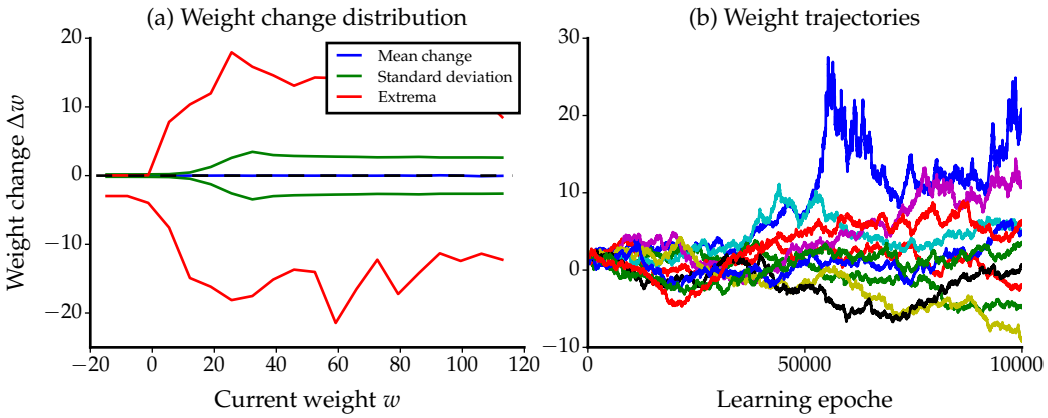

Figure 7.7.: REALISTIC INPUT REDUCED WEIGHT CHANGE BIAS. When generating spikes on more realistic input patterns, in temporal bins with low input kernel values the voltage could be small also for large weights. (a) Weight change distribution is wider due to more flexible spiking probabilities. (b) Consequently, weight trajectories do not show saturation for extreme weights.

weights, where we need to clip all firing probabilities above one in the binomial setting. Hence, we investigated how 'real' poisson firing with the same rates affect the weight change dynamics.

In real poisson spiking, the firing intensity function is not bounded, but grows exponentially. This is also reflected in the standard deviation of the weight changes that increases without a limit as $\sigma = \sqrt{\phi} = \sqrt{k \exp(\beta V_{rest})} \cdot \sqrt{\exp(\beta w)}$. These results could be confirmed also empirically in the simulations (see Figure 7.8, panel (c)). Similarly, the maximal positive weight change is described by ϕ itself, leading to exponentially increasing possible weight changes and hence instable simulation results. The results that we obtained in the corresponding explicit learning simulations (weight trajectories in panel (d)) look still relatively well behaved due to the relatively small learning rate of $\eta = 0.002$; for larger learning rates, weights diverged much faster.

As for the discrete implementations, the actual weight change distributions are different for time-varying PSPs. In this case, the learning process might even benefit from these potentially smaller firing intensities; however, it does not generally prevent weights from diverging, as also empirically obtained in the simulations of Section 7.2.2.

To sum up the results from this section, we need to interpret the equation (7.3) with care and soften the conclusions from Fremaux et al. (2010) about the unbiased learning. When weight updates are drawn independently, unbiasedness is guaranteed by equation (7.3). However, during the learning process, each current weight change has an effect on the following weight change. If the distribution with which updates are drawn varies for different realizations of w, the net weight change over several representations is biased. Even though this effect is relativized when we include time varying PSPs, for non-vanishing learning rates the standard deviation of the weight

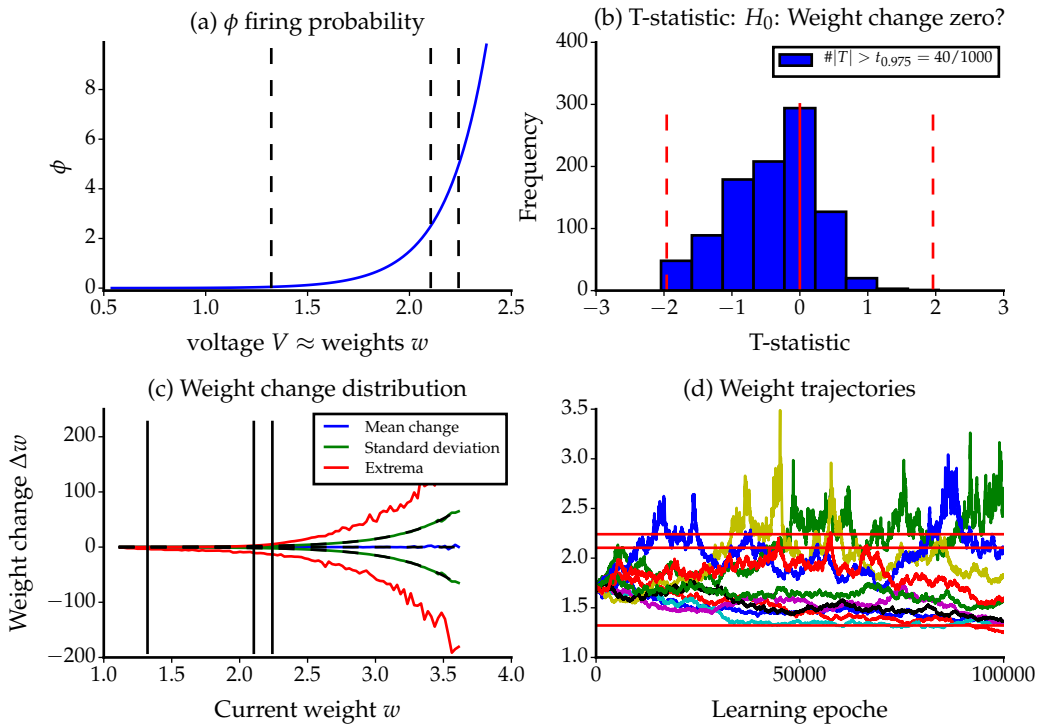

Figure 7.8.: REAL POISSON SPIKING FAVORS TRULY DIVERGING WEIGHTS. With real poisson spiking, exceedance of the firing intensity function of panel (a) is reflected in even more spikes. (b) T-statistic distribution is slightly shifted to the left. (c) Standard deviation of the empirical weight change distribution grows unboundedly for larger weights, which mirrors in sharper jumps of the weight trajectories in panel (d).

change distributions clearly depends on w, with quantitatively different behavior for different spike generation processes. These findings may explain the strongly diverging results obtained for the implementation based on poisson spike generation in Section 7.2.2. As these findings seem to represent properties of the learning rule inherent to the escape noise neuron, they may also influence other studies based on this type of learning to a certain degree.

8

SUMMARY AND DISCUSSION

8.1 SPECIALIZATION BY LOCAL WEAKLY SELECTIVE TRAINING

Sensory detection tasks can be highly complex, like identifying a predator in a noisy visual environment. One hypothesis how such complex problems are solved in the mammalian brain is that a large number of neurons distribute the task among each other and hence share the load. Distribution of the task can for example be achieved by individual neurons specializing to respond only to subsets of a behaviorally relevant target class. This idea is supported by the large number of selectively responding neurons in higher cortical areas (Quian Quiroga et al. 2005, Desimone et al. 1984, Tian et al. 2001, Kusmierek et al. 2012). How specialization automatically emerges within populations of neurons during a supervised classification task, however, is not obvious. Information accessible to individual neurons is limited and especially direct communication between neurons, that would help to actively divide the task, is biologically implausible.

In this thesis, we have introduced the Tagging algorithm, a population learning mechanism that does not require any inter-neuronal communication. The algorithm is based on a simple two-layer network architecture. Processing neurons that are arranged in a single layer receive common sensory input and project their binary response to an abstract readout neuron. The population decision is defined by the number of responding neurons that is compared to a decision criterion d. Whereas in case of a population false positive, each erroneously firing neuron adjusts its synaptic efficacies, we introduced a selective local rule for updates on population misses. Here, only neurons whose membrane potential exceeds a learning threshold are 'tagged' to correct their firing behavior. This biologically plausible, weakly selective learning procedure indeed showed to be sufficient to break the symmetry of the individual neurons and to induce specialization within a population of tempotron neurons.

By comparing the Tagging algorithm to a simpler population learning rule that trained all neurons that had been wrong on a corresponding population error trial ('trainall'), we showed that the obtained specialization within the network indeed resulted from the weakly selective training and not from population averaging alone. The classification error on binary problems where the target class incorporated a discrete or continuous substructure was significantly reduced with the Tagging algorithm compared to the naive trainall procedure. Also, a weighted training scheme, suggested by Urbanczik and Senn (2009) could not keep up with the high performance of Tagging for problems where the target class is diverse and hence high specialization is beneficial.

Similar Network Architectures in the Mammalian Cortex?

Can we find evidence in the brain that neurons are arranged in a similar manner to the Tagging population, with highly specialized neurons projecting to an accumulating readout neuron? A direct proof of the existence of such populations is challenging, since this would require experimental findings to provide two kinds of information. Firstly, how are all recorded neurons connected? And secondly, to which sensory features do all neurons respond to? Most recordings do either allow for measurement of responses of a large number of cells but do not provide information about the connectivity (for example multielectrode array recordings) or the other way around (for example electron microscopy) and only recently first attempts have been made to combine those techniques (Bock et al. 2011).
However, some experimental studies reveal neuronal properties that are in agreement with the proposed architecture. Especially neurons in the visual object recognition pathway show similarities to those within a Tagging population. The role of the processing neurons in a Tagging population can be compared to feature selective neurons in the inferior temporal cortex (ITC), while the readout neuron might be localized in the prefrontal cortex (PFC). The study of Sigala and Logothetis (2002) has shown that ITC neurons learn to respond more selectively to a subset of features that is relevant for the actual task the animal is performing than to task-irrelevant features. Additionally, Freedman et al. (2003) demonstrated that these ITC neurons show stronger individual stimulus-selective (representing subclasses determined by the corresponding features) than category-selective activity (representing the binary label). From the perspective of a Tagging architecture, these findings reflect the learned subclass-specific responses of processing neurons. On the other hand, category affiliation is encoded in the responses of PFC neurons throughout the whole trial. An overall later response onset of the PFC neurons indicates delayed information as provided by projections from the ITC. Only at long latencies, also ITC neuron responses contained information about the class affiliation. This suggests a top-down 'teacher' feedback from PFC neurons, consistent with the Tagging population structure where processing neurons receive a global feedback from the readout neuron.

Selective Training based on Self-Adjusting Learning Threshold

A complex classification problem could be easily divided across the population if communication between neurons would be unrestricted. However, neurons in the brain do not have access to internal states of other cells. In this thesis, we considered the even more conservative theory that each neuron only receives a global feedback about a possible population error and is unaware of all other individual processing neuron performances. Hence, specialization within the population needs to arise solely on the basis of locally available information.

The weakly selective Tagging algorithm resolved this issue by the definition of a neuron-specific training threshold v_{train}. For a population miss, every non-firing neuron whose membrane potential crosses the corresponding threshold undergoes an LTP step. Even though fix values of v_{train} are possible, the algorithm converges more robustly if training thresholds change during the learning process. This notion of

a sliding synaptic modification threshold is consistent with the Bienenstock-Cooper-Munro model, whose biological implementation has been experimentally validated throughout the last decades (Bear 1995, Stanton 1996, Cooper and Bear 2012, Hulme et al. 2013). We propose an adjustment procedure of the training threshold that does not require any knowledge of other neurons' internal states. Assuming that false positives and misses are balanced for an optimal classification, we adjust the training threshold in the direction of an equilibrium between LTD and LTP steps relative to the false positive to miss balance in the population. This equilibrium describes a stable state where learning does not show any drift. Biologically, adjustment of the plasticity induction threshold in a similar manner would require the individual neurons to keep track of the classification performance of the population in addition to its own learning history. One can speculate that this kind of information may be provided by nearby astrocytes. Whereas evidence for intracellular mechanisms to induce metaplasticity was provided already in 2001 (Abraham et al. 2001), a recent study of Hulme et al. (2013) also suggests the role of intercellular signaling pathways for example via astrocytes to modulate plasticity. Together, these mechanisms might account for the proposed adjustment approach using intra- and intercellularly available properties.

To obtain a smooth estimator of the deviation from the targeted equilibrium, one needs to evaluate the statistics over the history of individual learning properties and population errors. The time course over which the statistics are calculated is a crucial factor. For simplicity, we chose to adjust the training threshold in a sequential manner after every learning epoch of a fixed number of p presented input patterns (usually $p \approx 1000$). For a neuron in the brain, the choice of a fixed integration window seems unlikely. An established approach to implement averaging over a fixed time window in a biological plausible way is to use a low-pass filter in continuous time (Urbanczik and Senn 2009). This low-pass filtered eligibility trace mimics a decaying memory trace of a neuron. However, we tested another approximation that does not require any information storage within the neuron. In an online procedure, after each presented input pattern, we simply added or subtracted a small constant to v_{train}, dependent on which of the four components that determine the equilibrium were present at the current trial (see Section A.4 in the appendix). This method worked well as long as the individual training thresholds were initialized with nearby values, but broke down for highly dispersed thresholds. Further evaluations are needed to allow for a reliable use of this online adjustment rule.

8.2 PROPERTIES OF THE TAGGING ALGORITHM

Tagging Reduces Pairwise Correlations between Neurons

Can we make any predictions about specific neuronal properties that arise from the proposed weakly selective learning scheme? One main finding of this thesis is that specialization does not only emerge across discrete known substructures, but also within explicit structures by means of spreading decisions in time. This strategy reduced pairwise noise correlations between neurons within the population.

In particular, when we trained several neurons to distinguish a noisy single subclass from background noise, we observed that individual neurons focused on different temporal segments within their input spike patterns. The temporal window within which a neuron fired, became the narrower the more neurons shared the noisy pattern, keeping their pairwise noise correlations low. We uncovered that this low noise correlation was achieved at the expense of a higher individual neuron error. This finding can be explained as follows: For any target subclass represented by a spatiotemporal template spike pattern, we assume that there exists a short feature that optimally separates this template pattern from null patterns. A neuron that learns to respond solely in the presence of this feature yields a higher classification performance than other neurons whose synaptic weights are tuned to detect other suboptimal features. In a Tagging population, however, neurons specialize to respond to distinct features as a consequence of the weakly selective training. Hence, these neurons can not all choose the optimal feature but need to focus on suboptimal features, which in turn leads to higher individual errors than optimal.
On the other hand, when we trained the same set of neurons separately and combined their decision in a population after training, the neurons solved their individual problem with the minimally achievable individual error, but responded in a highly correlated manner. These higher correlations in turn led to a comparable population error for separate training as the Tagging algorithm provided. This 'high accuracy-low correlation' tradeoff had been reported and discussed for general regression ensembles based on (weighted) averaging (Krogh and Vedelsby 1995, Brown, Wyatt, and Tino 2005). Our findings suggest that also learning procedures based on the incorporated binary population decision face the same tradeoff.

The obtained spike count or noise correlation values for both, Tagging and separate training, were similar to those reported for neurons in different sensory brain areas (broad values between 0.01 and 0.26, see M. Cohen and Kohn 2011). Hence, the absolute correlation values did not directly rule out either one (albeit the individual training of course requires biologically implausible supervised explicit division of labor before training). However, M. Cohen and Kohn (2011) pointed out that spike count correlations are generally weaker between neurons with low firing rates, even though the mathematical definition of correlation is independent on the mean value. They argued that this relation can be ascribed to the fact that even if existent, correlated membrane fluctuations are masked if they do not cross the firing threshold. The same correlations in the membrane trace yield much higher spike correlations if the voltage crosses the firing threshold frequently than when it remains subthreshold most of the time. The neurons in the proposed population are binary and hence their maximal spiking response of one spike per $T = 500$ms would correspond to a firing rate of less than 2Hz. Following the above argument, is is reasonable to relate the obtained noise correlations of the tempotron populations to those values reported for rarely spiking neurons. Here, values between 0.01 and 0.11 for neurons with maximal 5Hz spiking rate agree with the average noise correlations of ≤ 0.05 for all populations that were trained with the weakly selective Tagging algorithm.

Further evidence, that learning in the brain could be a achieved by a similar mechanism as Tagging, was provided by an experimental study of Gu et al. (2011). Here, it was shown that training a group of macaque monkeys on a fine heading discrimination task reduced pairwise correlations in neurons in the dorsal medial superior temporal area. The neurons of the trained monkeys showed lower average noise correlations than those of naive monkeys. Correlation values were measured during passive fixation after training and hence the obtained difference could not be attributed solely to attention. The empirically observed decline of noise correlations as a result of learning are consistent with the dynamics in a Tagging population.

Ability of Tagging to Uncover Structure in more Realistic Data

Most simulations with the Tagging algorithm were performed on binary classification tasks where the complex target class was modeled as a composition of discrete poisson pattern input spike patterns. An etiological example was studied by Gifford et al. (2005). Here, neuronal responses discriminated two acoustically different communication calls that both indicated the presence of high-quality food from calls that signified low quality food. However, mimicking a similar task by discrete poisson patterns is of course a simplification. Additionally, some sensory structures are difficult to map to discrete subclasses. The viewing angle of a visual object for example can be interpreted as a continuously varying physical quantity. Modeling a corresponding classification problem would require an extension to continuously related input spike patterns.
To investigate if the results obtained for the discrete poisson pattern scenario could be translated to more realistic settings, we took two approaches: Firstly, we made the transition from discrete to continuously related input patterns to check whether clearly isolated features were required for specialization. Secondly, we evaluated the performance of Tagging on spike patterns generated from real acoustical speech signals to move away from artificially structured inputs.

For the continuously related spike patterns, the Tagging algorithm with a small decision threshold led to a specialization of the individual neurons within the continuous stimulus range. In contrast to the discrete patterns, each neuron selected a region of this parameter space, within which it generated spiking responses. Corresponding individual tuning curves showed bell-shaped characteristics with little overlap that spanned the whole stimulus range. If we interpreted the continuous scale for example as the view angle of a visual face, the readout neuron of such a Tagging population would respond to faces regardless of its view, in accordance with experimental findings of visual neurons in the IT cortex (Kourtzi and DiCarlo 2006, Quian Quiroga et al. 2005, Freiwald and Tsao 2010). Hence, a view-invariant readout neuron could be generated by accumulating responses from view-dependent processing neurons in the Tagging population. Indeed, a citation from Booth and Rolls (1998) already suggests a similar mechanism to generate view-invariance:

> 'Further, the anatomical location of the view-invariant cells reported in this paper suggest that these cells do not form a separate anatomical popula-

> tion of neurons in the IT, but rather intermingled with neurons that are view-dependent and require a certain feature or combination of features for activation. This supports the notion that these view-invariant responses are being formed by associating together the responses of view-dependent visual neurons.'

The Tagging algorithm also yielded promising results in a spoken digit detection task. Here, for most target digits, training of a Tagging population with two neurons reduced the classification error compared to the trainall algorithm. For at least some digits, this improvement was obtained by means of specialization on speaker-dependent properties such as the gender of the speaker. These results emphasize the usefulness of Tagging also for a long studied real world problem where the input feature space is diverse and not composed of clearly separable subclasses.

The Decision Criterion as Balance between AND and OR Operations

The Tagging algorithm assigns an input pattern to the target class if at least d neurons evoke a spiking response. This decision threshold d is the crucial parameter of the algorithm that constrains possible population solutions. The simulations in this thesis mostly relied on a small decision threshold d that yielded specialized processing neurons. However, some sensory tasks might be better represented by decision thresholds closer to the population size. In visual object recognition (DiCarlo et al. 2012), the hypothesis of recognition-by-components from Biederman (1987) postulates recognition of an object if a certain number of features is present in the visual image. Such feature-combining neurons have been reported by Freiwald et al. (2009) and Brincat and Connor (2004). A face selective neuron that responds only to an image of a (cartoon) face if at least hair and irises are present (Freiwald et al. 2009) could be modeled by a Tagging readout neuron with $d = 2$. Freiwald et al. (2009) report that several neurons in the middle face patch of neurons are influenced by one to four face parts. Some of these neurons responded highly nonlinearly to combinations of these features, as would be the case for AND classification with maximal d.

Neuronal specificity and invariance both increase along the ventral visual stream, with an abundance of highly specific and highly invariant neurons in different cortical areas (Booth and Rolls 1998, Kobatake and Tanaka 1994). However, within the same neuronal population in the ITC, neurons that are highly specific showed to be less tolerant and hence face a tradeoff (Zoccolan et al. 2007). This finding furthermore supports the idea that object specificity and invariance are generated by antagonistic AND-like and OR-like operations as implemented in state-of-the-art object recognition models (Serre et al. 2007, LeCun 2012). In the Tagging population, the decision threshold d balances between such AND and OR operations (for $d = m$ and $d = 1$ at both extremes, respectively). Hence, by changing this parameter, we could implement different readout neurons that move along the continuum of high specificity and high tolerance.

Choosing the decision threshold that allows for an optimal separation of target and null class on a given problem is challenging. Even though we deduced an explicit representation of the theoretical Tagging population error in this thesis, application of the formula to determine the optimal threshold for a given setting of m neurons and k subclasses is not practical. The formula requires knowledge about properties of individual neurons that cannot simply be measured in separate studies. As mentioned above, neurons in the Tagging population do not behave as individually trained neurons but operate in a suboptimal error regime to ensure low noise correlations in the population. Another option to avoid extensive searches for the optimal decision threshold in larger populations would be to use heuristics to adjust d during learning. Incorporating a learning mechanism of d into the current Tagging algorithm is an interesting future research direction that is beyond the scope of the present study.

8.3 GENERALIZABILITY TO OTHER NEURON MODELS

The idea of weakly selective training is not limited to a particular neuron model. The Tagging algorithm only requires a spike generating process that depends on an internal variable (like the maximum membrane potential). In addition to a binary integrate-and-fire neuron with the tempotron learning rule, we also tested the Tagging algorithm with perceptrons and neurons with stochastic output spike generation and reinforcement learning strategies. Here, the application to perceptrons allowed for a more theoretical investigation of the algorithm and a direct comparison with the basic mixture of experts model. On the classical XOR problem, we did not only proof the benefit of using a Tagging population with two neurons over a single perceptron, but also schematically exposed the qualitatively different solutions of mixture of experts and Tagging. Application of Tagging to the MNIST handwritten digit classification furthermore confirmed its specialization ability. Even though we did not follow a specific hypothesis about which structures in the handwritten data set could be utilized for specialization, the visual representation of the perceptron weights after learning hinted at potential handwriting styles as specialization cues for some digits. Whereas the mixture of experts model outperformed Tagging in the artificial task to discriminate odd from even digits, for the single digit classification both methods yielded comparable results.

Additionally, we chose a stochastically firing neuron model as basis for the Tagging ensemble. This model was chosen, firstly, to enable a fair comparison to the recently published attenuated learning population approach of Urbanczik and Senn (2009) and, secondly, to investigate the flexibility of the proposed approach to deal with stochastic firing behavior. We showed that the Tagging population approach yielded superior classification performance also for stochastic neurons and offered variable problem solving strategies by means of the additional decision threshold parameter d.

In the process of implementing the neuron model, we discovered a fundamental problem with the state-of-the-art implementation of the reinforcement learning proce-

dure. The reinforcement learning is based on the assumption that without any reward signal, synaptic changes according to the proposed learning rule should be unbiased since expected and realized spike numbers agree on average. However, this assumption is only valid for independent weight changes. In a careful analysis we showed that indeed weight change distributions, and hence also the realized updates, strongly depend on the current synaptic weights. This violation of the underlying assumption leads to instabilities during the learning process that may result in diverging synaptic weights. It remains to be seen how our findings on instable learning dynamics may influence previously published results based on the same neuron model.

8.4 OUTLOOK - TAGGING IN TIME?

Throughout the thesis we assumed that all features appear at a specific point in time and the population receives feedback about a potentially erroneous decision directly after that input pattern. However, more realistically, relevant features appear asynchronously in a continuous stream of spatiotemporal input activity and sometimes different events need to be accumulated before a clear detection of the sensory input is possible. One might think of specific phonemes in speech that only in combination signify a spoken word or images within a movie that in the right order define an action. Can Tagging also be extended to such temporally embedded features? View selective face detector neurons for example might be activated in a temporal sequence if we watch a person turning his head. We might also think of identifying an object that covers the visual field by a kind of temporal recognition-by-components: Directing our gaze first to a component that we recognize as an eye and shortly after to another component that corresponds to a nose will finally lead to the identification of a face. Individual neurons that each selectively respond to one of these components will be activated sequentially and provide a population prediction for a target as soon as d components are detected. For detection of such a temporally accumulated target, a long integration time of the readout neuron in the Tagging population gains even more importance than for the original Tagging. The readout neuron's memory needs to last at least the whole continuous stream in which behaviorally relevant features might appear that assemble the target object, whereas the processing neurons in the population operate on shorter time scales that correspond to the length of the individual features.

Assuming that the long integration time of the readout neuron could be biologically implemented, for instance by a more complex recurrent neuronal circuity (Seung et al. 2000), Tagging in time would be an interesting generalization of the proposed algorithm. Even more flexibility would be provided if we based the population on neurons that can learn to generate a precise number of output spikes. For this extended Tagging population, instead of the number of firing neurons, the total number of spikes summed across all neurons could be thresholded against a decision criterion. This population decision additionally enables an indirect weighting of the individual processing neurons - neurons that have a broader range of firing contribute more to the

population decision than those whose spike count varies only slightly. Selection of the neurons that undergo learning could be again based on an internal variable that correlates with how close the neuron was to fire an additional spike in case of a population miss.

Our first preliminary results with a Tagging population of multispike tempotrons (Gütig 2012) are promising. With features embedded in a continuous stream, we implemented OR and AND classification problems: The tasks was to identify a target pattern if either at least one behaviorally relevant feature occurred in the stream (according to an OR classification) or if all features were present (according to an AND classification). While the first classification problem would correspond to the above example of the turning head - as soon as one face detector neuron is active, the visual would be identified as a face -, the second AND task would relate to the temporal recognition-by-components paradigm. Application of the global Tagging rule and its biologically plausible local variant to a population of multispike tempotrons yielded promising results on these problems. Neurons within the population specialized to fire for different features and the total spike count was balanced across the population, indicating an equally shared load similar as obtained with the binary Tagging algorithm. These preliminary findings indicate that an extension of Tagging to the temporal domain is possible, opening up the attractive opportunity to implement more realistic sensory tasks in time and space.

A

APPENDIX

A.1 BACKGROUND: NERVE CELLS AND THEIR WAY OF COMMUNICATION

Nerve cells or *Neurons* are the processing units of our brain[1]. They are highly interconnected, with up to 100.000 (input- or *afferent*) connections in the human brain, and communicate with each other via directed electrical signals. The outer membrane of a neuronal cell body is negatively charged, with a *resting potential* between -40 mV and -90 mV at rest. When neurons receive exciting signals, their membrane potential increases. Small signals ebb away; however, if the excitement is strong enough to lift the electrical potential over a certain *firing threshold*, voltage dependent channels in the neuronal membrane open and allow for an influx of positive ions. As a positive feedback these further increase the voltage until an ion equilibrium is reached, yielding always the same amplitude and shape of the electrical impulse. This *action potential* or *spike* is transmitted to the 'output' terminal of the neuron (the *synaptic terminal* of the *axon*) where it again induces chemical cascades that release chemical molecules (*transmitter* molecules) outside the cell. Other neurons with 'input' endings (*dendrites*) close to the output ending of the releasing *presynaptic* cell can now uptake these molecules by specific receptors in their cell membrane. Depending on the type of the released molecule (that is inherent to the presynaptic neuron), chemical cascades in this *postynaptic* neuron either induce a decrease of the internal membrane potential (decreasing the probability to also elicit an action potential or *fire*) or an increase (increasing the probability to fire as it lifts the membrane potential closer to the firing threshold as described above). This whole communication process is called (chemical) *synaptic transmission* with the complex of pre- and postynaptic neuron endings summarized as *synapse*.

The amount by which the postsynaptic membrane potential is increased (or decreased) as a consequence of transmitter release varies between synapses. The amplitude of this (excitatory/inhibitory) *postsynaptic potential* (*(E/I)PSP*) defines the *synaptic strength* (also *synaptic weight/ synaptic efficacy*) of a synapse. Synaptic connections are not hard-wired but plastic. A common hypothesis is that learning new memories is achieved by strengthening synaptic connections between neurons that transmit the according information. Apart from short-time modifications that may last only a few milliseconds, different cellular and molecular mechanisms induce long-lasting synaptic changes, denoted by *long-term depression* (*LTD*) and *long-term potentiation* (*LTP*) for decrease and increase of synaptic strength, respectively.

1 All following information in this section is a condensed summary of chapters 1,2 and 5 from Purves et al. 2008.

A.2 SUPPLEMENTARY METHODS

The main methods have been already introduced in Section 2. However, a detailed description of the simulated experiments is necessary if someone intends to reproduce the results. In the following we summarize the parameters used for the individual simulations and the initialization of the tempotrons. Additionally, we describe the explicit calculation of the errors and other statistical measures as well as the complex generation of the single neuron error curves.

A.2.1 *Simulation details*

Initialization of the Tempotrons

To initialize the individual tempotrons, synaptic weights are chosen from a Gaussian distribution with zero mean and standard deviation of $\sigma_w = 0.01$, independently for each neuron. In order to start with a reasonable firing rate and hence speed up the learning, we pretrain all neurons on a set of patterns that have the same statistics as the null class pattern with random labels until the neurons fired for at least 50% of the patterns (monitored over the last 1000 patterns). Specifically, for the discrete and the continuous embedded poisson pattern scenarios, the patterns in the pretraining were identical to randomly labeled patterns from the infinite null class. Due to the finite data set in the digit recognition problem, we decided to not present the actual patterns in the pretraining process, but generated similar patterns. For this, we averaged the spike rates of each input channel over all digits and presented constructed poisson patterns with this empirically obtained rate during the pretraining process. Again, individual pretraining stopped as soon as the neurons fired for more than 50% of the patterns.

For the poisson pattern problem, the time constants for the individual integrate-and-fire neurons are set to $\tau_m = 10ms$ and $\tau_s = 2.5ms$. In the digit recognition task, we use $\tau_m = 50ms$ and $\tau_s = 12.5ms$ due to longer input features. The starting membrane potential was fixed to $V_0 = 0$ and the firing threshold to $\vartheta = 1$. The learning rate of $\eta = 0.01$ was kept constant throughout all simulations.

Parameters for Learning Threshold Adjustment

As discussed in Section 4.1, learning works best if the learning threshold in the local Tagging variant is adjusted according to the relation of population error components to individual learning statistics. We used the sequential threshold adaptation throughout the thesis for all local Tagging simulations based on tempotrons and escape noise neurons. Hereby, except for Section 4.1, where we evaluated different described parameters, we we used the following constellations: All tempotron neurons started with an initial learning threshold of $v_{train} = 0.8$ (at a firing threshold of $\vartheta = 1$) and all escape noise neurons with $v_{train} = -0.5$ (at a hypothetical firing threshold of $\vartheta = 0$ in the limit of large β). Regardless of the neuron type, the firing threshold was up-

dated after each learning epoch, under the constraint that all components needed for the adjustment appeared at least $c_{min} = 5$ times after the previous update. In this case, thresholds were changed according to the update rule in equation (3.1) with $a_\chi = 0.0001$. Silent neurons were defined as those that have not fired a single spike within the last $silent_{lim} = 1000$ input patterns. All of these neurons reduced their learning threshold by $a_{dec} = 1e - 6$.

Simulation Parameters

For the simulations based on tempotron learning, the noise parameters and running times that we used to generate the individual figures are summarized in Table A.1.

Figure number	p_{del}	σ	p	$lmax$
4.1	0.2	0.07s	1000	10000
4.2	0.2	0.07s	1000	30000
A.1	0.2	0.07s	1000	30000
4.4	0.5/0.2[a]	0.07s	1000	20000
4.5	0.8	0 s	1000	20000
4.6	0.8	0 s	1000	50000
4.7	0.8	0 s	1000	50000
4.8,4.9	0.4	0.1s	1000	10000/20000/10000[b]
4.10, 4.11	0.2	0.07 s	1000	10000
5.1	0.3	0.01 s	2061/2075[c]	50000
5.2,5.3	0.3	0.01 s	4136	20000
6.1,6.2	0	0	$2 \cdot m$	10000
6.3	0	0	36017	10000
6.4,6.5	0	0	36017	3000

Table A.1.: SIMULATION PARAMETERS - TAGGING WITH TEMPOTRONS AND PERCEPTRONS. Spikes were shifted with Gaussian noise of standard deviation σ and deleted with probability $p_{del} \cdot 100\%$. Learning was applied for $lmax$ learning epochs on p presented patterns each.

a 0.5 for the more difficult subclass, 0.2 for other subclasses and null patterns
b individual neurons/tagging/separate training
c Male/Female data set

For the application of Tagging to reinforcement learning in Chapter 7, we oriented ourselves on the simulations from Urbanczik and Senn (2009). Here, the authors performed binary classification on a set of fixed p patterns with $p_{target} = p/2$ belonging to the target class and $p_{null} = p/2$ belonging to the null class. Since they did not add any noise sources, the following table A.2 summarizes only the number of patterns p, the number of target patterns p_{target} and the learning times.

A.2.2 *Error and Firing Matrix Calculations*

Error Rates

For all simulations in the main Chapter 4, the classification error is defined as the number of misclassified patterns divided by the number of actually presented patterns. We

Figure number	p	p_{target}	Δt	$lmax$
7.3	30	15	0.2 ms	2000/5000[a]
7.4	30	15	0.0005 - 0.2	20
7.1	100	5-95	0.2	1000
7.2	100	50	0.2	1000

Table A.2.: SIMULATION PARAMETERS - TAGGING WITH ESCAPE NOISE NEURONS. Number of total patterns p, target patterns p_{target} and time grid Δt for the reinforcement learning studies. Learning was applied for $lmax$ learning epochs on the p randomly shuffled patterns each.

a With only one pattern per learning epoch, see Section 7.2.2

evaluated the misclassification error during learning in each learning iteration separately. Specifically, the classification error at iteration l is defined as

$$e(l) = \frac{\#\{x_j : \ell(x_j) \neq \hat{\ell}(x_j), j = 1, \ldots, p\}}{p}$$

The weights of the individual neurons very likely change during this calculation due to the corresponding learning rule. If no fixed data set is available (as for example the case for the spoken digit detection), a usual learning cycle contains $p = 1000$ patterns unless stated otherwise.

Whenever learning curves are visualized, we used a exponentially smoothed version of this time-varying error to minimize the effect of stochastic noise fluctuations. Here, the presented data points are calculated as

$$MA(e(l)) = \alpha MA(e(l-1)) + (1-\alpha) \cdot e(l)$$

for $l > 1$ and $MA(e(1)) = e(1)$ as initial value. The smoothing factor α is chosen as $\alpha = 0.8$ for all simulations.

In general, all depicted error curves are averaged over 100 different starting seeds for the initial weights and the spike pattern generation process.

Final errors usually denote an average of the last $l_{mean} = 1000$ learning iterations of $e(l)$. Only if a fixed pattern batch exists, final errors are defined as the error rate, evaluated once on this specific data set after $lmax$ learning iterations. For the simulations in Section 4.3, population errors for the local Tagging and the separate training were obtained after complete learning simultaneously with the noise correlation by presentation of $p_{test} = 5000$ patterns per target subclass/null class each.

In all figures that include error bars, the length of the bar corresponds to the standard deviation across all simulations. We decided to depict the standard deviations instead of the standard error of the mean to visualize the variability within the data. Whenever information about statistically significant error differences are provided in the text, significance was evaluated on a two-sample t-test for independent samples on a 99% confidence level.

Firing Matrices

A similar evaluation process as for the empirical errors applies to the represented firing probabilities. We monitored firing for every neuron and every subclass (for discrete classes) during individual learning epochs and normalized them by the number of actually occurring patterns from the subclass in the corresponding learning epoch. Figures of the firing matrices denote the firing probabilities within the last learning epoch. For the continuous patterns in Section 4.4, actual firing rates were explored after learning by presenting patterns with α-values on a 0.01-spaced grid, each 100-times with the corresponding spike time jitter values and spike deletion probabilities. For the spoken digit and the MNIST task with fixed training and test set, firing matrices are evaluated on the test set after learning. Here, presented each test pattern once, for both data sets without noise. For representation purposes in case of the discrete poisson pattern classification (Figure 4.1 and Figure 4.4), we sorted firing matrices template-wise, starting with the templates represented by the highest individual firing rate to the lowest individual firing rate. The neurons are sorted accordingly on a a second level. For all classification problems with 'meaningful' categories (as digits in the odd vs even classification), no sorting was applied and neurons were represented in the order of their random labeling.

Output Spike Time Calculations

In order to calculate the output spike times for Figures 4.5 and 4.6 we presented 1000 random patterns from the single target subclass and evaluated the response of the individual neurons. For Figures 5.2 and 5.3, all available patterns from the corresponding target digit were shown once. If the neuron fired a spike, the temporal difference between output spike and onset of the relevant stimulus was listed and added to the statistics. For the embedded poisson patterns, each target template was presented after exactly 500ms so that the onset time of the relevant stimulus was fixed to that time point. For the digits, onset of the relevant stimulus agreed with onset of the spike pattern ($0\,ms$). For the list of collected output spike times, histograms with equally spaced bins were calculated and plotted for each neuron individually (speech) or on the basis of common bins for all neurons (poisson patterns).

Specialization Measures

The specialization measures described in Section 2.4 are used for estimating the degree of specialization within a population in different ways. We evaluated these measures based on the final synaptic weights on all p patterns after the learning process for the spoken digit detection as well as the MNIST data set.

Representative Examples

For all figures that contained a visual representation of a firing matrix, we selected the simulation seed on the basis of the corresponding measure of interest. Unless otherwise stated, we chose representative examples as simulation seeds that yielded a

final classification error closest to the median final error among all 100 seeds (Figures 4.6, 4.10, 6.5 and 7.2). For some paradigms, different performance measures are used; for Figure 4.1, typical examples in panel (a) and (b) are based on the median of the off-diagonal sum. Also the learning thresholds for the automatic learning threshold update evaluations in Figure 4.2 were chosen from simulations with median error. For Figure 4.4, we chose a simulation example where the mean latencies of the first and second specialist neurons were closest to the total average of the first and second specialist firing times among all simulations. Similarly, Figure 4.5 shows the output latencies of the simulation where the sum of individual mean latency differences to the overall sorted mean latencies was minimal.

For the individual digit recognition in Figure 5.2 and 5.3, we chose examples that were closest to the corresponding specialization measure. The odd against even studies for the spoken as well as the handwritten digit recognition (Figure 5.1 and 6.3) used the combination of all measures to select the representative example:

$$S_i = \arg\min \left(\sum_{sm \in SM} \frac{(sm - s\bar{m})^2}{\mathrm{s\bar{m}}^2} \right)$$

where sm denotes a specialization measure out of the set SM containing the relative classification error, the KL divergence, the SmW redundancy as well as the average pairwise correlations within target and null class. The median of each measure is denoted by $s\bar{m}$.

Error rates for converged simulations on the XOR problem were always zero and could hence not be used for selecting a representative example. For the illustration of the perceptron (expert) weights in Figures 6.1 and 6.2 we hence chose the simulation that found a solution in the least number of learning epochs for simplicity.

A.2.3 *Details on the Tradeoff Experiments in Section 4.3*

Individual Error Curves

To compare the theoretical population error with the actually observed Tagging error in specific classification problems, knowledge of the individual error values f and $ms = 1 - h$ is necessary. For Section 4.3 we hence trained neurons with the same neuron-specific parameters as in the population individually on target detection tasks with targets composed of one to five subclasses. Since for rarely occurring targets, the neuronal classifier focuses more on minimizing its false positive rate and vice versa, we varied the target probability to mimic such situations. Specifically, for each neuron and target pattern scenario we ran different simulations with target probabilities ranging from 2% to 98% in 2% steps (yielding 49 different target probabilities). The diverse individual miss and false positive pairs that resulted from these target probability variations are plotted in the first panel of Figure 4.8 for five exemplary seeds.

To fit an average curve, for every number of subclasses and every seed we calculated miss values on a linearly-spaced grid $\hat{f}$ (100000 grid points) as linearly interpolated values between any two adjacent measured false positive/miss pairs. Specifically, the interpolated miss values were computed as $\hat{ms}(\hat{f}_i) = ms_l + a \cdot (\hat{f}_i - f_l)$ for $i = 1 \ldots 100000$, where f_l denote the largest obtained false positive value smaller than $\hat{f}_i$, ms_l the miss value obtained for that f_l and $a = (ms_r - ms_l)/(f_r - f_l)$ with f_r the smallest measured false positive rate larger than $\hat{f}_i$ and ms_r the corresponding miss value. The interpolated values were averaged over all seeds, yielding the plotted curves in panel (a) of Figure 4.8 for every number of subclasses.

It is important to note that averaging over a large number of patterns is necessary to obtain clean results for the empirical f/ms pairings: The standard error of the estimator for the false positive rate for example is $\sigma(\hat{f}) = \frac{1}{\sqrt{p}}\sqrt{\hat{f}(1-\hat{f})}$ and can be calculated as follows: The number of false positives F in p null patterns is binomial distributed with $B(p, f)$. We simply estimate F by counting the number of false positive patterns in a single learning epoch. The standard deviation of $\hat{F}$ is given by the standard deviation of the binomial distribution $\sigma(\hat{F}) = \sqrt{pf(1-f)}$ Estimating f as $\hat{f} = \hat{F}/p$ yields $\sigma(\hat{f}) = \frac{1}{\sqrt{p}}\sqrt{\hat{f}(1-\hat{f})}$. For Figure 4.8 each cross represents a (f,ms) pair of a simulation averaged over $2 \cdot 10^6$ patterns.

Separate Training

For the separate training, we used the average linearly interpolated individual neuron curves to obtain the optimal strategy $\tilde{S}$ and the optimal false positive rate/miss pairing for any decision threshold d via the formula (4.2). Any of the $m = 20$ neurons was trained on $\tilde{S}$ of the subclasses, whereas the assigned templates only shifted by one for each neuron. Specifically, neuron j was trained to distinguish targets composed of templates $j, \ldots, j + \tilde{S}$ (with circular boundary conditions) from noise. Since false positives and misses that yield the optimal population error are usually not balanced, we needed to adjust the target probability to obtain the optimal miss/false positive pairing in the separate training simulation. For the theoretical optimal values we chose the target probability to match that one used for generating the closest obtained false positive rate in the single neuron simulations. So for a given optimal f^* and the measured set of tuples $(p_{target,k}, f_k, ms_k)_k$ for every individual neuron simulation $k = 1, \ldots, 49 \cdot 100$, we chose p_{target} to be the target probability $p_{target,l}$ that minimized $(f_l - f^*)^2$. After training individual neurons based on these preselected templates with this close to optimal target probability, we combined the final neurons in a population and evaluated the Tagging error based on $p_{test} = 5000$ patterns for each subclass and null class.

Measuring Neuron-Specific Error Rates from the Tagging Population

Whereas for individual neurons calculating the miss rate is straight forward, in a specialized population we face the problem that not every neuron is obliged to detect a specific pattern if other neurons are specialized for it. Hence, we based the miss rate

of a neuron in a local Tagging population on the number of target patterns that did not elicit a spiking response but generated a voltage trace that exceeded the training threshold v_{train}. These patterns correspond to the ones where a learning step would occur during training. To normalize this measure, we divided it by the number of all patterns where the corresponding voltage trace exceeded v_{train}, hence considering only patterns where the neuron would either fire or learn in the Tagging algorithm. We evaluated this miss rate together with the standard false positive rate (here simply number of null patterns that elicited a spike relative to the number of presented null patterns) and the population error on $p_{test} = 5000$ patterns for each target subclass and null class after the simulation.

Based on the fraction of patterns that evoked v_{train} exceeding responses, we furthermore defined the number of subclasses the neuron is tuned to. If at least 2500 of the 5000 patterns of a specific subclass lifted the membrane trace above v_{train}, this subclass was added to the group of specialized subclasses of a neuron. The number of subclasses that the majority in the Tagging population was specialized to according to the previous definition is color coded in panel (c) of Figure 4.9.

A.3 FROM MIXTURE OF EXPERTS TO GLOBAL TAGGING

The global Tagging algorithm that we introduced in Section 3.1 relies on a weakly selective training method that aims at enhancing initial preferences of neurons and thereby induces specialization within a population. The idea of enhancement of initial preferences also motivates the state-of-the-art mixture of experts model that we reviewd in Section 2.2.2. Here, we describe the relation between both models by transforming the mixture of experts model stepwise to the global Tagging algorithm.

The mixture of experts model relies on two sets of neurons. Expert neurons specialize on subregions of the input space and provide a solution to the classification problem for patterns in this subregion (modeling conditional probabilities of the mathematical theorem of total probability). Gating neurons, on the other hand, gate current input patterns to the corresponding experts (modeling the probability of the pattern to origin from a specific subregion). This strategy effectively *divides* the problem into smaller subproblems (gating neurons) that are *conquered* by the corresponding expert neurons. Formally, the population readout of this divide-and-conquer approach mirrors a weighted average:

$$\hat{\ell} = \Theta\left(\sum_{j}^{M} g_j y_j - 0.5\right)$$

with g_j denoting the outputs of the gating network that sum up to one and hence represent relative responsibilities, and y_j the continuous outputs of the corresponding expert neuron. In the following, we describe the steps that lead to a transformation of the mixture of experts idea to the simplified global Tagging learning.

(1) One idea to avoid this multiplicative readout is to replace the continuous output of the expert neurons by a binary one. In fact, we even define the expert neurons' output independent on the input pattern. Specifically, we select a number of 'target' experts $m < M$ and set $y_j = 1$ iff $j \leq m$ and zero otherwise. Like this, the whole classification is performed by the gating neurons - if they gate the input pattern to the target detectors, the population output is one, and zero otherwise. Apart from a modified response function

$$\hat{\ell} = \Theta\left(\sum_j^m g_j - 0.5\right),$$

also the error function and hence the weight updates change for constant expert outputs. Updates for the synaptic weights of the target gating neurons that are coupled to the target expert neurons (and with inverse sign for the null gating neurons) are given by:

$$\Delta \boldsymbol{v_j} = \begin{cases} g_j \left(\frac{1}{\sum_{i=1}^m g_i} - 1\right) \boldsymbol{x} & \ell = 1 \\ -g_j \boldsymbol{x} & \ell = 0 \end{cases}$$

(2) The factor $(\frac{1}{\sum_{i=1}^m g_i} - 1)$ has a positive sign for $\sum_{i=1}^m g_i < 1$ and approaches zero if the probability to gate the input to target experts reaches one. Hence, this update scales down the learning, the more correct the population decision is. However, this also implies that learning is induced even though the population response is already correct. Instead of this attenuated learning, we decided to tolerate incorrect individual responses and apply learning only on population errors with full learning rate. Specifically, we update $\Delta \boldsymbol{v_j} = (2 \cdot \ell - 1) g_j \boldsymbol{x}$ whenever $\hat{\ell} \neq \ell$.

(3) Originally, the definition of g_j as

$$g_j = \frac{\exp(z_j)}{\sum_i \exp(z_i)}$$

describes a soft-max function on the fields of the corresponding gating neurons. For a clearer specialization of the (target/null) neurons on the (target/null) patterns and a simpler biological realization, we consider the hardmax function to be more appropriate. Using $g_j = 1$ if $j = \arg\max_i z_i$ and zero otherwise yields learning for only that neuron that has the maximal response. With this definition, the readout of the whole population is $\hat{\ell} = 1$ if the gating neuron with the maximal internal state belongs to the target detector subpopulation, and zero otherwise.

(4) One problem regarding the biological plausibility that has not been tackled so far is that for the current readout the population still requires knowledge about the inner states of the gating neurons. For a realistic output, we hence need to consider discrete binary individual spiking outcomes. Instead of testing if one gating neuron from the target detector population yields the maximal internal state, we test if one

gating neuron from that population yields an internal state above the firing threshold. This corresponds to a simplified readout of

$$\Theta\left(\left(\sum_{j=1}^{k}\Theta(z_j)\right)-1\right),$$

where the null neurons do not play a role in the learning or classification process anymore. However, whereas before the target neurons were required to be less active than any of the null neurons, with this new readout, the target neurons need to be silent for null inputs which is a stricter constraint. The learning rule for the target neurons remains as

$$\Delta \boldsymbol{v_j} = \ell \boldsymbol{x} \quad \text{for} \quad j = \arg\max_i z_i.$$

(5) With $\Delta v_j = \ell \boldsymbol{x}$ being the original synaptic update rule for an individual neuron in the population, this learning procedure agrees with the Tagging algorithm for $d = 1$. However, we allow more flexibility in the Tagging algorithm. For a large number of neurons in the population, silencing all neurons in response to a null pattern would require an infinitely small false positive rate of the individual neurons. The inclusion of a decision threshold d to account for more flexible population solutions reveals the global Tagging algorithm.

A.4 ONLINE ADJUSTMENT OF THE TRAINING THRESHOLD

Throughout the thesis, we use a sequential update of the learning threshold to reach a desired equilibrium of population error components and individual learning rates according to equation (3.1). Here, we update the training threshold of each neuron after each learning epoch of p patterns if all of the four required components appeared at least c_{min} times within the previous learning epoch. The minimally required number of components herein ensures a sufficiently large statistic of the fraction χ.

However, another more approximate way to stabilize the equilibrium would be to 'online' update the learning threshold after each input pattern presentation. For this, we note that $\chi = \log(FP) - \log(MS) + \log(LTP_j) - \log(LTD_j)$ and $v_{learn,j} = v_{learn,j} + a_\chi \left(\log(FP) - \log(MS) + \log(LTP_j) - \log(LTD_j)\right)$. A very naive approximation is $v_{learn,j} = v_{learn,j} + \tilde{a}_\chi FP - \tilde{a}_\chi MS + \tilde{a}_\chi LTP_j - \tilde{a}_\chi LTD_j$ with $\tilde{a}_\chi \approx 1/1000 a_\chi$. Like this, we can directly update the learning threshold whenever one of the four components is active. Specifically, we increase $v_{learn,j}$ by $\tilde{a}_\chi$ if the population decision caused a false positive and the corresponding neuron j did not undergo an LTD step. With the proposed population learning this is equivalent to the situation that the neuron correctly remained silent and hence stabilized its current behavior by increasing the threshold. On the other hand, we implement a decrease of the learning threshold by $\tilde{a}_\chi$ if the population missed a target pattern and the corresponding neuron did not learn at this pattern. Dependent on d that means that it has been either correctly firing or its maximum membrane potential stayed below the current learning threshold.

We also investigated the performance of this online adjustment compared to the sequential updates for the same classification problem as in Section 4.1.2 where the number of discrete target subclasses and neurons in the population agreed. When the neurons started with common learning thresholds, indeed also the online method with $\tilde{a}_\chi = 5 \cdot 10^{-6}$ arrived at specialized solutions after a similar number of learning iterations. The corresponding error rates were comparable between the sequential and online variant, basically regardless of the initial values (with a tendency to larger fluctuations for the online method for larger starting values, see Figure A.1, blue line in panel (a)). However, when neurons started learning from dispersed thresholds, the mean and the standard deviation of the classification error increased dramatically (panel (a), outermost blue bar). In this situation, both, the best and the worst simulation, showed at least partly unreasonable behavior of the learning threshold trajectory. The maximally reasonable value of $\vartheta = 1$ was exceeded by both versions and several neurons for relatively long time spans. Even though the final distribution of learning thresholds looks a bit smoother (inset in panel (b) of Figure A.1), still a large fraction of values terminate beyond $v_{train} = 1$. We do not exclude that this problem can be circumvented by choosing the update steps more carefully. However, we decided to not address this issue and utilize the better behaving sequential learning threshold adjustment instead for all simulations with the (local) Tagging algorithm in this thesis.

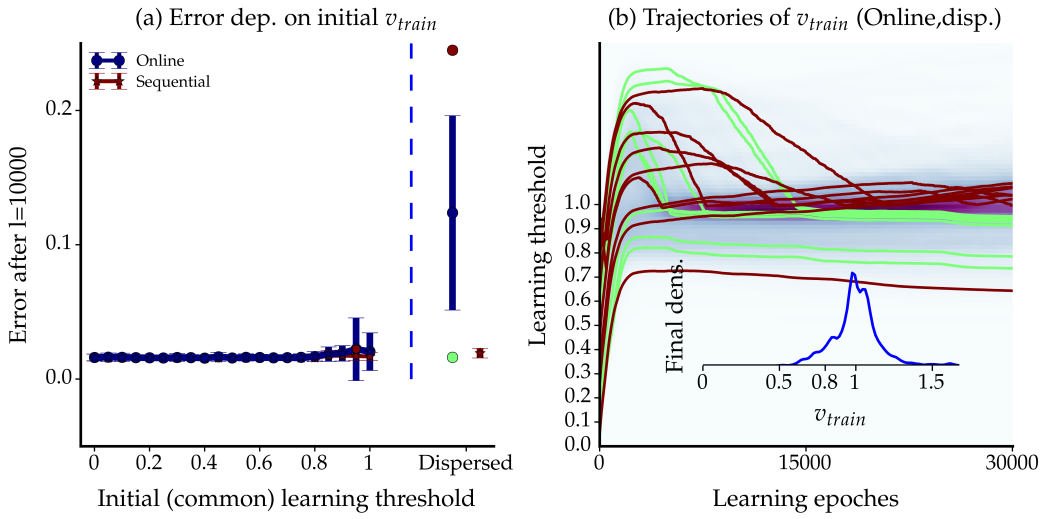

Figure A.1.: ONLINE VS. SEQUENTIAL LEARNING THRESHOLD ADJUSTMENT. (a) Population error dependent on the initial common training thresholds in local Tagging populations with sequential (red curve, reproduction of Figure 4.2, panel (a)) and online threshold adjustment (blue). Whereas the online updates behave reasonably well for low and intermediate starting values, large values show large fluctuations. With dispersed initial values, the online procedure fails to find good solutions in almost all simulations. (b) Evolution of learning thresholds for the online adjustment rule. Red and green trajectories illustrate learning threshold evolution for all neurons in the simulation yielding highest and lowest error, respectively. The underlying density plot is much more diffuse than for the sequential training, with almost 50% of the final threshold exceeding the firing threshold $\vartheta = 1$.

A.5 NOTES ON PERCEPTRON LEARNING

In Chapter 6, we solve the common XOR problem by applying the Tagging algorithm to a population of perceptrons. Here we show formally that the Tagging algorithm finds a solution to the problem within a finite number of steps under certain assumptions. Furthermore, we discuss some problems arising from possibly violation of those due to weight norm peculiarities in the following.

A.5.1 *XOR Proof*

Notations

Let $\boldsymbol{I_1} = (z, 1, -1)$ and $\boldsymbol{I_3} = (z, -1, 1)$ denote the target patterns, the zeroth coordinate $z = \sqrt{2}$ chosen such that they are orthogonal to each other. Similarly, let $\boldsymbol{I_2} = (z, 1, 1)$ and $\boldsymbol{I_4} = (z, -1, -1)$ denote the null patterns, again orthogonal to each other. Whenever a neuron performs a learning step on pattern $\boldsymbol{I_k}$, the weights are updated according to $\Delta \boldsymbol{w} = \eta l_k \boldsymbol{I_k}$. Hence, the fields of the neuron with a pattern I_j changes after learning with pattern $\boldsymbol{I_k}$ to $h_{j,new} = (\boldsymbol{w_{old}} + \Delta \boldsymbol{w}) \cdot \boldsymbol{I_j} = h_{j,old} + \eta l_k \boldsymbol{I_k} \cdot \boldsymbol{I_j} = h_{j,old} + \Delta h_{j,k}$. The deviation from the previous field $\Delta h_{j,k}$ after learning of pattern $\boldsymbol{I_k}$ is denoted the learning effect of $\boldsymbol{I_k}$ on $\boldsymbol{I_j}$. In vector notation, the learning effects are $\Delta \boldsymbol{h_{:,1}} = (4, 2, 0, 2)$ for $\boldsymbol{I_1}$, $\Delta \boldsymbol{h_{:,2}} = (-2, -4, -2, 0)$ for $\boldsymbol{I_2}$, $\Delta \boldsymbol{h_{:,3}} = (0, 2, 4, 2)$ for $\boldsymbol{I_3}$ and $\Delta \boldsymbol{h_{:,4}} = (-2, 0, -2, -4)$ for $\boldsymbol{I_4}$. Let P_1 denote the first perceptron, P_2 the second perceptron. The fields of perceptron i are denoted with $h_{ij} = \boldsymbol{w_i} \cdot \boldsymbol{I_j}$, $j = 1, 2, 3, 4$.

Assumptions

The learning rate λ is assumed to be sufficiently small such that no jumps are allowed in the sense even with a change from $h_{11} > 0$ to $h_{11} < 0$ the relation $h_{11} > h_{21}$ holds when $h_{11} > h_{21}$ before the learning iteration (which also implies $h_{21} < 0$). More precisely, let η be smaller than the minimal difference $|h_{11} - h_{21}|$ and $|h_{13} - h_{23}|$ (for the initial fields) such that the responsibilities for a target pattern can not change within one cyclic representation.

Furthermore, it is assumed that the input patterns occur in cyclic order which guaranties that a change from $h_{11} > 0$ to $h_{11} < 0$ induced by a null pattern will be followed by a possible correction of the step by pattern $\boldsymbol{I_1}$, such that the 'net drift' of $\Delta h_{11} = -2 + 4 = 2$ can be considered. Unless otherwise stated, the net drift will denote the total change within one learning cycle.

The proof makes use of the individual perceptrons' convergence theorem. This theorem can be applied, if the individual labels of the single perceptrons form a linearly separable space and stay constant over the learning epochs.

Structure

The structure of the proof will be as follows:

(A) Any situation where both perceptrons fire for the same target pattern, is instable and irreversible, leading to a situation where at most one perceptron fires for each target pattern.

(B) If the individual labels of the two perceptron form two independent linearly separable problems, including that each perceptron does not fire for the target pattern of the other perceptron, the system converges. Formally, this means: $h_{11} > h_{21}, h_{23} > h_{13}, h_{21} < 0, h_{13} < 0$ (w.l.o.g).

(C) If one perceptron initially dominates the target patterns, i.e. $h_{11} > h_{21}, h_{13} > h_{23}$, and at most one perceptron fires for each target pattern, the system converges.

A

First of all we consider the case where two perceptrons fire for the same target pattern, without loss of generality say $\boldsymbol{I_1}$. Here, either none of the perceptrons fires also for $\boldsymbol{I_3}$ which results in an error in $\boldsymbol{I_3}$, or at least one of the perceptrons fires for $\boldsymbol{I_3}$ and hence for one of the null patterns as well, producing an error in the null pattern. We investigate the situation according to the existence of an error in one of the null patterns:

1. No error in null pattern: With none of the perceptrons firing for a null pattern, the inequalities $h_{13} < 0$ and $h_{23} < 0$ hold as well. The perceptron with the larger field for $\boldsymbol{I_3}$, say P_1 is attracted with a net drift $\Delta h_1 = (0, 2, 4, 2)$ (no responsible null pattern error). This error does not have a direct effect on h_{11} until the perceptron fires for one of the null patterns (it has to cross the null pattern first before reaching $\boldsymbol{I_3}$) and hence leads to the situation

2. Error in null pattern: The perceptron P_1 that is responsible for an error in the null pattern (i.e. that has the largest field $\arg\max_i(h_{i,2})$), say $\boldsymbol{I_2}$, has a net drift of at least $\Delta h_{11} = (-2)$ in the first target pattern (a positive effect on $\boldsymbol{I_1}$ is only possible by an error in the target pattern itself due to the orthogonality of $\boldsymbol{I_1}$ and $\boldsymbol{I_3}$). With an additional net drift of $\Delta h_{12} = (-4)$ (and $\Delta h_{14} = (-2)$ which is not of interest here) the following situations may occur:

   (a) P_1 stops firing for $\boldsymbol{I_1}$: Since P_2 still fires for $\boldsymbol{I_1}$, this does not cause an error and no attraction of $\boldsymbol{I_1}$ is possible. Hence in this case we leave the instable state (**Goal**).

   (b) P_1 stops firing for I_2 (but still fires for $\boldsymbol{I_1}$): This is only possible if $h_{13} < 0$. In this case, we will either have an error in $\boldsymbol{I_3}$ (in case P_2 does not fire for $\boldsymbol{I_3}$ either) that needs to be corrected by the perceptron whose field is larger for $\boldsymbol{I_3}$, starting again at `No error in null pattern` (1.), but with overall smaller fields for $\boldsymbol{I_1}$. Or the other perceptron fires for $\boldsymbol{I_3}$ and hence produces an error in the null pattern which needs to be corrected, starting again at `Error in null pattern` (2.) for the other perceptron P_2.

   (c) P_1 looses the responsibility for I_2, when $h_{1,2} < h_{2,2}$. In this case, the learning continues with P_2 instead of P_1 before either P_2 ends up in situation (a) or (b) or the responsibility changes again. Since the net drift on the current

responsible neuron $\Delta h_{i1} = (-2)$ is still negative, one of the situations (a) and (b) need to be present after a finite number of steps.

An infinite iteration of the different situation in the state is not possible, since the net effect on h_{11} and h_{21} is negative. This will sooner or later lead to the realization of situation (a): One of the perceptrons stops firing for $\boldsymbol{I_1}$. Hence, the situation that both perceptrons fire for the same target pattern is not stable and the step from moving away from this state is irreversible. Hence, for the following proof we only have to consider states where at most one of the perceptrons fires for the target patterns in the following.

B

Hypothesis: If the individual labels of the two perceptron form two independent linearly separable problems, including that each perceptron does not fire for the target pattern of the other perceptron, the system converges. Formally, this means: $h_{11} > h_{21}, h_{23} > h_{13}, h_{21} < 0, h_{1,3} < 0$ (w.l.o.g).

Proof: In this initial situation, both perceptrons face two linearly solvable problems with local labels $(1,0,0,0)$ and $(0,0,1,0)$ for P_1 and P_2 respectively. If we show that these labels do not change during learning, the system converges according to the individual perceptron convergence theorem. Specifically, we have to show that the inequalities $h_{23} > h_{13}, h_{11} > h_{21}$ hold. First, consider the inequality $h_{11} > h_{21}$. here, we have the following possibilities:

(a) $h_{11} > 0$: The first perceptron fires for target pattern $\boldsymbol{I_1}$, errors can only occur by the remaining patterns $\boldsymbol{I_2}, \boldsymbol{I_3}, \boldsymbol{I_4}$. With $h_{23} > h_{13}$, the first perceptron is only responsible to correct for errors of maximal the two null patterns. The mean effect in one learning iteration on the field h_{11} is hence one of the three possible values $\Delta h_{11} \in \{0, -2, -2 \cdot 2\}$ (for no error, one and two errors being corrected by P_1 , respectively). As long as $h_{11} > 0$, the inequality holds due to the requirement $h_{21} < 0$. If the perceptron stops firing, we arrive at the following situation

(b) $h_{11} < 0$: If P_1 does not fire for $\boldsymbol{I_1}$, an error occurs in the target pattern since also $h_{21} < h_{11} < 0$[2]. With additional potential errors of the null patterns $\boldsymbol{I_2}$ and $\boldsymbol{I_4}$, the perceptron P_1 changes its field h_{11} with a mean drift of $\Delta h_{11} \in \{4, 2, 0\}$ (corresponding to no (responsible) error in the null patterns, one (responsible) error and two (responsible) errors).

For both possible situations assuming a cyclic presentations of the input patterns and a sufficiently small learning rate, the net learning effect on h_{11} is larger or equal to zero and hence the inequality $h_{11} > h_{21}$ can only be violated by an increase of h_{21}. This is not possible since an increase of h_{21} can only be induced by an update by the target pattern $\boldsymbol{I_1}$ itself. Since the inequality holds, only the perceptron P_1 is trained with this pattern, in summary guaranteeing $h_{11} > h_{21}$.

2 A jump from situation (a) directly beyond h_{21} is only possible with a small probability if the learning rate is chosen sufficiently small

On the other hand, training on the null patterns only decreases the fields of the target patterns. Hence, learning for P_1 will on average result in a decrease of h_{13}, strengthening the second inequality.

Since the problem is totally symmetrical for $h_{23} > h_{13}$, using the same arguments also the second inequality holds. Hence, the local labels of the two perceptrons do not change during learning, assuring the convergence of each individual perceptron and hence the convergence of the tagging algorithm.

C

Hypothesis: If one perceptron initially dominates the target patterns, i.e. $h_{11} > h_{21}, h_{13} > h_{23}$ and at most one perceptron fires for each target pattern, the system converges.

Proof: Without loss of generality assume $h_{11} > h_{21}$ and $h_{13} > h_{23}$, i.e the perceptron P_1 dominates the system and hence (at least in the case that $h_{11} < 0$ and $h_{13} < 0$) tries to solve the problem with local labels $(1, 0, 1, 0)$ which is as XOR problem not linearly separable. We hence have to proof that during learning the inequalities change and we arrive at the previous situation (B).

Having this initial condition, the fields h_{21} and h_{23} can not increase since potential errors in $\boldsymbol{I_1}$ and $\boldsymbol{I_3}$ have a learning effect on P_1. In this situation, the second perceptron is automatically dropped out of the learning after a short number of iterations because it can fire for at most one of the null patterns:

(1) No firing: If P_2 does not fire for any of the pattern, it does not come into play naturally, since the field of P_1 is larger for both target patterns.

(2) Firing for one null pattern: If P_2 fires for one null pattern, say $\boldsymbol{I_2}$, it is either responsible for the induced error or not.

   (a) P_2 is responsible: If $h_{22} > 0 > h_{12}$ or $h_{22} > h_{12} > 0$, we observe a negative drift of $\Delta h_2 = (-2, -4, -2, 0)$ which distracts the second perceptron from $\boldsymbol{I_2}$. The fields for the targets are reduced as well, strengthening the initial inequalities. P_2 is distracted from the pattern $\boldsymbol{I_2}$ until it stops firing for $\boldsymbol{I_2}$ or it looses the responsibility and we hence end up in (b).

   (b) P_1 is responsible: If $h_{12} > h_{22} > 0$, we have to investigate the drift depending on the firing state for the other patterns. (i) We observe a negative drift in h_{12} as long as at least one of the target patterns is classified correctly, reducing the field of P_1 in $\boldsymbol{I_2}$. This will either lead to P_1 loosing its responsibility for $\boldsymbol{I_2}$ and hence leading back to (a) (but also here infinite looping is not possible since the net drift on $\boldsymbol{I_2}$ is negative and at some point one neuron will stop firing for $\boldsymbol{I_2}$) or to P_1 stopping to fire for a target pattern: (ii) In case, the perceptron does not fire for any of the target patterns, the net drift will be $\Delta h_1 = (-2, -4, -2, 0) + (4, 2, 0, 2) + (0, 2, 4, 2) = (2, 0, 2, 4)$. The field for $\boldsymbol{I_2}$ is neutral, but the perceptron is attracted by the target patterns, crossing one of them in a finite number of learning iterations. In this case, we will land again in (i). Iteration between state (i) and (ii) is possible,

but the net effect on h_{12} will be negative, such that at some point P_2 will be responsible for the error in $\boldsymbol{I_2}$ (arriving at (a)).

Hence, even after finite iterations between the above states, the second perceptron will stop firing for any of the pattern eventually.

Since the net drift for h_{11} and h_{13} is neutral, the second perceptron will not come into play again in a natural way if we assume cyclic presentation of the input patterns and a sufficiently small learning rate. In this situation where the first perceptron rotates while trying to solve the inseparable XOR problem and the second perceptron does not fire anymore, the algorithmic trick with the minimal firing rate applies. After the minimal firing threshold is crossed, each field update induced by an error in one of the target patterns is applied to the second silent perceptron, yielding a positive drift ($\Delta \boldsymbol{h_2} = (4,2,0,2)$ or $\Delta \boldsymbol{h_2} = (0,2,4,2)$ for the pattern $\boldsymbol{I_1}$ and $\boldsymbol{I_3}$ respectively) in h_{21} and h_{23} (and also in the null patterns). This positive drift applies until

(a) the second perceptron's field h_{21} or h_{23} is larger than h_{11} or h_{13} respectively. In this case, the second perceptron is responsible for the corresponding target pattern, say $\boldsymbol{I_1}$, and the local labels have changed to $(0,0,1,0)$ for the first and to $(1,0,0,0)$ for the second perceptron. Hence, we arrived at the situation B that has been shown to converge.

(b) the second perceptron starts firing for a null pattern, say $\boldsymbol{I_2}$, before it is close enough to a target pattern. A distraction is caused by the null pattern $(-2,-4,-2,0)$ which leads again to the start of the situation: $h_{21} < 0 \; AND \; h_{23} < 0$. Nevertheless, the net drift in h_{21} and h_{23} is positive such that the chance to arrive at one of the target patterns before firing for a null pattern increases each time the algorithmic reactivation is applied. Hence, after a finite number of iterations, we arrive at the converging Situation B.

Therefore, with the help of the reactivation of silent neurons for the minimal firing rate we ensure a change of the local labels to the balanced situation B, so convergence is guaranteed.

A.5.2 *Decreasing Weight Norm and Implications for XOR Learning*

Weight Norm Decrease

One important issue regarding the previous proof for the XOR problem that we will address in this paragraph is that, in fact, the amount by which the current weight plane of a perceptron changes its angle and its offset is determined not by the absolute learning rate but by the relative rate related to the norm of the weight vector.

For large values of $\boldsymbol{w}$, changing the weight vector by a fix small learning rate η will only slightly correct the the scalar product with the trained pattern $(\boldsymbol{w} + \Delta \boldsymbol{w}) \cdot \boldsymbol{I} = \boldsymbol{w} \cdot \boldsymbol{I} + \eta \ell ||\boldsymbol{I}||^2$.[3] However, for a large learning rate compared to the

3 Here, $||\boldsymbol{x}|| = \sqrt{\sum_{i=1}^{n} x_i^2}$ denotes the L^2 norm of a vector $\boldsymbol{x} = (x_1, \ldots, x_n)$

size of $\boldsymbol{w}$ this new scalar product can already change its sign. Hence, we will take a closer look at the norm of the weights.

(1) Changes of the (squared) weight norm:

$$\begin{aligned} ||\boldsymbol{w} + \Delta\boldsymbol{w}||^2 &= \boldsymbol{w}\cdot\boldsymbol{w} + 2\eta\ell\boldsymbol{w}\cdot\boldsymbol{I} + \eta^2\boldsymbol{I}\cdot\boldsymbol{I} \\ &= ||\boldsymbol{w}||^2 + 2\eta\ell h + \eta^2||\boldsymbol{I}||^2 \end{aligned}$$

Since the pattern $\boldsymbol{I}$ was an erroneous pattern, the summand $2\eta\ell h$ is always negative. The norm of the weights will increase if and only if $\eta^2||\boldsymbol{I}||^2$ dominates:

$$||\boldsymbol{w} + \Delta\boldsymbol{w}||^2 \geq ||\boldsymbol{w}||^2 \tag{A.1}$$

$$\iff \eta||\boldsymbol{I}||^2 \geq 2|h| \tag{A.2}$$

(2) New classification of $\boldsymbol{I}$ with new field h^*:

$$\ell h^* = \ell(\boldsymbol{w} + \Delta\boldsymbol{w})\cdot\boldsymbol{I} = \ell h + \eta||\boldsymbol{I}||^2$$

With this, the new weights correctly classify $\boldsymbol{I}$ if and only if

$$\eta||\boldsymbol{I}||^2 \geq |h|, \tag{A.3}$$

where we again make use of the fact that $\ell h < 0$ due to a previous misclassification of $\boldsymbol{I}$.

The inequalities A.2 and A.3 together provide an interesting conclusion

$$\underbrace{\underbrace{\eta||\boldsymbol{I}||^2 < |h|}_{\text{No pattern correction}} < 2|h|}_{\text{Weight norm decreases}}$$

Consequently, whenever a learning step is performed that does not directly correct the misclassified pattern, the norm of the perceptron weights sinks. The norm can only grow after a learning step that qualitatively changes the response to the trained input pattern.

This also includes that when the learning rate is chosen sufficiently small as to change the hyperplanes only slightly, the weight norm sinks as long as an input pattern is being approached. The more the norm decreases, the smaller gets the absolute value of the scalar product $|\boldsymbol{w}\cdot\boldsymbol{I}|$. Hence the decrease in the norm weakens until an equilibrium is achieved at $|\boldsymbol{w}\cdot\boldsymbol{I}| = \eta/2||\boldsymbol{I}||^2$ on average (see also Figure A.2, panel (b)).

Implications for the XOR Learning

In the XOR convergence proof in A.5.1 a crucial assumption was that whenever an update step is performed on $\boldsymbol{I_k}$, the field relation between the two neurons does not change qualitatively. This assumption was made for small η. However, as we have seen, the learning rate has to be put into relation to the current field and hence the weight norm of the individual neurons. When a single neuron performs many update steps on a problem that it can not solve, its norm decreases until the equilibrium is achieved. In this case, the effective learning rate is very high, violating the assumptions of the proof. Indeed, this situation can occur also in the Tagging learning algorithm if one of the neurons arrives at a situation where it stops firing and being responsible for all patterns (see Figure A.2, panel (a) for a perceptron hyperplane trajectory of the first 100 learning epochs). As explained above, direct reintegration can be achieved by preferred training mechanisms. However, we implemented this in a way that we wait for a certain number of silent patterns before this mechanism comes into play. During that time, the other neuron tries to solve the inseparable XOR problem alone, decreasing its weight norm until equilibrium (Figure A.2, panel (c) and (d)). Every weight update will lead to an immediate correction of the previous error and generation of a new one. The situation becomes absorbent if the silent neuron is reactivated to fire for a null pattern with very small field (its norm is still high such that the same learning rate has only a minor update effect on that neuron). In that case the gradient like learning will always choose the neuron with the smaller norm to permute between all patterns, leading to an effective weight change of $\Delta \boldsymbol{w} = 0$ (Figure A.2, panel (b) for later iterations - the four presented hyperplanes for the blue neuron will repeat through all remaining iterations).

To circumvent such a situation where the basic assumptions of the XOR proof are violated, we introduced also a normalized Tagging variant in Section 6.1. Here, after each update step, the weights of both neurons are normalized to the average norm of both neurons. Like this, both neurons are again equally affected by the weight updates and the silent neuron would enter the classification as an equivalent learner.

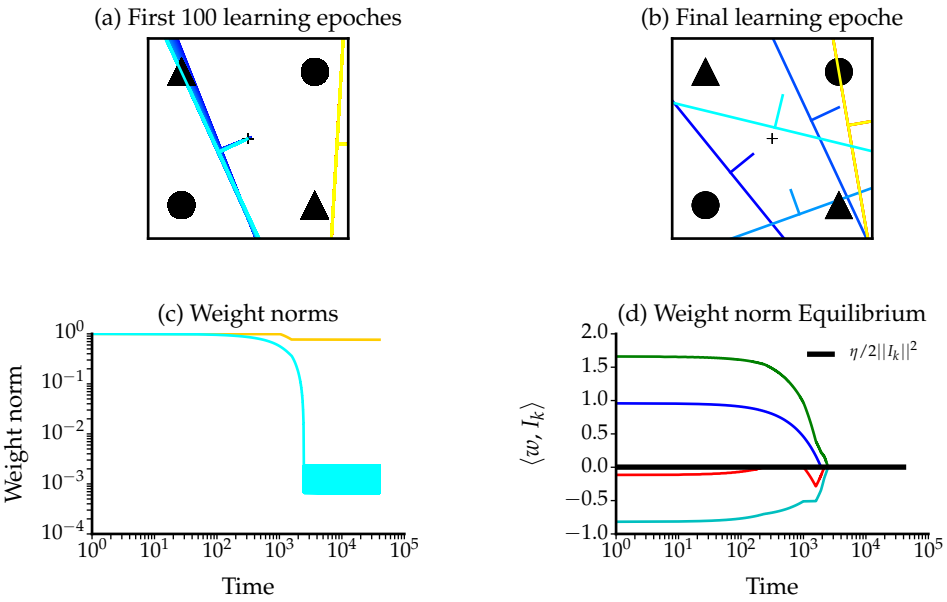

Figure A.2.: NON-CONVERGENT XOR SITUATION. Two perceptrons in a Tagging population fail to solve the XOR problem. (a) In the first 100 learning epochs, the blue perceptron is pulled smoothly towards the target triangle in the upper left corner and away from the null circle (time history from dark blue to turquoise). The yellow perceptron is silent and does not change. (b) After many learning epochs, the yellow perceptron learned to fire (erroneously) for a null pattern via direct reactivation. The blue perceptron permutes through the pattern set to vainly solve the XOR problem. Due to the low weight norm, every learning step induces a qualitative change in the hyperplane (single learning steps in the order from dark blue to turquoise). (c) Evolution of the weight norm of the yellow and blue perceptron over time. Due to solely training the blue perceptron, its weight norm decreases until it reaches the equilibria of $\boldsymbol{w} \cdot \boldsymbol{I_k} = \eta/2||\boldsymbol{I_k}||^2$ for every pattern $\boldsymbol{I_k}$,colored traces in panel (d), and alternates between them.

BIBLIOGRAPHY

Abraham, W. C., Mason-Parker, S. E., Bear, M. F., Webb, S., and Tate, W. P. (2001). "Heterosynaptic plasticity in the hippocampus in vivo: a BCM-like modifiable threshold for LTP". *Proc. Natl. Acad. Sci. USA* 98, pp. 10924–10929 (cit. on pp. 41, 107).

Anselmi, F., Leibo, J. Z., Rosasco, L., Mutch, J., Tacchetti, A., and Poggio, T. (2015). "Unsupervised learning of invariant representations". *Theoretical Computer Science* http://dx.dio.org/10.1016/j.tcs.2015.06.048 (cit. on p. 6).

Bear, M. F. (1995). "Mechanisms for a sliding synaptic modification threshold". *Neuron* 15 (1), pp. 1–4 (cit. on p. 107).

Biederman, I. (1987). "Recognition-by-Components: A Theory of Human Image Understanding". *Psychological Review* 94 (2), pp. 115–147 (cit. on pp. 36, 110).

Bienenstock, E. L., Cooper, L. N., and Munro, P. W. (1982). "Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex". *Journal of Neuroscience* 2, pp. 32–48 (cit. on p. 41).

Bizley, J. and Cohen, Y. (2013). "The what, where and how of auditory-object perception". *Nature Reviews Neuroscience* 14 (10), pp. 693–707 (cit. on p. 11).

Bock, D., Lee, W.-C., Kerlin, A., Andermann, A., Hood, G., Wetzel, A., Yurgenson, S., Soucy, E., Kim, H., and Reid, C. (2011). "Network anatomy and in vivo physiology of visual cortical neurons". *Nature* 471, pp. 177–182 (cit. on p. 106).

Booth, M. and Rolls, E. (1998). "View-invariant Representations of Familiar Objects by Neurons in the Inferior Temporal Visual Cortex". *Cerebral Cortex* 8, pp. 510–523 (cit. on pp. 109, 110).

Bowers, J. S. (2009). "On the biological plausibility of grandmother cells: Implications for neural network theories in Psychology and Neuroscience". *Psychological Review* 116 (1), pp. 220–251 (cit. on p. 10).

Breiman, L. (1996). "Bagging predictors". *Machine Learning* 24, pp. 123–140 (cit. on p. 4).

Brincat, S. L. and Connor, C. E. (2004). "Underlying principles of visual shape selectivity in posterior inferotemporal cortex". *Nature Neuroscience* 7(8), pp. 880–886 (cit. on p. 110).

Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). "Diversity Creation Methods: A Survey and Categorization". *Journal of Information Fusion* 6 (1), pp. 5–20 (cit. on p. 4).

Brown, G., Wyatt, J., and Tino, P. (2005). "Managing diversity in regression ensembles". *Journal of Machine Learning Research* 6, pp. 1621–1650 (cit. on p. 108).

Burrone, J., O'Byrne, M., and Murthy, V. N. (2002). "Multiple forms of synaptic plasticity triggered by selective suppression of activity in individual neurons". *Nature* 420, pp. 414–418 (cit. on p. 38).

Chandra, A., Chen, H., and Yao, X. (2006). "Trade-Off Between Diversity and Accuracy in Ensemble Generation". In: *Studies in Computational Intelligence (SCI)*. Vol. 16, pp. 429–464 (cit. on p. 63).

Chechik, G., Globerson, A., Tishby, N., Anderson, M. J., Young, E. D., and Nelken, I. (2001). "Group Redundancy Measures Reveal Redundancy Reduction in the Auditory Pathway". *Advances in Neural Information Processing Systems* 14, pp. 173–180 (cit. on p. 31).

Cichy, R., Pantazis, D., and Oliva, A. (2014). "Resolving human object recognition in space and time". *Nature Neuroscience* 17 (3), pp. 455–462 (cit. on p. 8).

Cohen, M. and Kohn, A. (2011). "Measuring and interpreting neuronal correlations". *Nature Neuroscience* 14 (7), pp. 811–819 (cit. on p. 108).

Cooper, L. N. and Bear, M. F. (2012). "The BCM theory of synapse modification at 30: interaction of theory with experiment". *Nature Reviews Neuroscience* 13, pp. 798–810 (cit. on pp. 40, 41, 107).

Dayan, P. and Abbott, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press (cit. on pp. 16, 19).

Desimone, R., Albright, T. D., Gross, C. G., and Bruce, C. (1984). "Stimulus-selective properties of inferior temporal neurons in the Macaque". *The Journal of Neuroscience* 4(8), pp. 2051–2062 (cit. on pp. 2, 9, 105).

DiCarlo, J., Zoccolan, D., and Rust, N. (2012). "How does the brain solve visual object recognition?" *Neuron* 73, pp. 415–434 (cit. on pp. 8, 110).

Freedman, D. J., Riesenhuber, M., Poggio, T., and Miller, E. K. (2003). "A Comparison of Primate Prefrontal and Inferior Temporal Cortices during Visual Categorization". *The Journal of Neuroscience* 23(12), pp. 5235–5246 (cit. on pp. 10, 11, 106).

Freedman, D. J., Riesenhuber, M., Poggio, T., and Miller, E. K. (2006). "Experience-Dependent Sharpening of visual shape selectivity in inferior temporal cortex". *Cerebral Cortex* 16, pp. 1631–1644 (cit. on p. 10).

Freiwald, W. and Tsao, D. (2010). "Functional compartmentalization and viewpoint generalization within the macaque face-processing system". *Science* 330 (6005), pp. 845–851 (cit. on pp. 3, 9, 109).

Freiwald, W., Tsao, D., and Livingstone, M. (2009). "A face feature space in the macaque temporal lobe". *Nature Neuroscience* 12 (9), pp. 1187–1196 (cit. on pp. 9, 110).

Fremaux, N., Sprekeler, H., and Gerstner, W. (2010). "Functional Requirements for Reward-Modulated Spike-Timing-Dependent Plasticity". *The Journal of Neuroscience* 30(40), pp. 13326–13337 (cit. on pp. 92, 98, 99, 103).

Freund, Y. and Shapire, R. (1997). "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of computer and system sciences* 55, pp. 119–139 (cit. on p. 4).

Freund, Y. and Shapire, R. (1999). "A short introduction to Boosting". *Journal of Japanese Society for Artificial Intelligence* 14 (5), pp. 771–780 (cit. on p. 4).

Gales, M. and Young, S. (2008). "The Application of Hidden Markov Models in Speech Recognition". *Foundations and Trends in Signal Processing* 1(3), pp. 195–304 (cit. on p. 70).

Gerstner, W. and Kistler, W. M. (2002). *Spiking neuron models*. Cambridge University Press (cit. on p. 18).

Gharavian, D. and Ahadi, S. (2007). "Statistical evaluation of the effect of gender on prosodic parameters and their influence on gender-dependent speech recognition". In: *Information, Communications and Signal Processing, 2007 6th International Conference on* (cit. on p. 69).

Gifford, G. W. I., MacLean, K. A., Hauser, M. D., and Cohen, Y. E. (2005). "The neurophysiology of functionally meaningful categories: Macaque ventrolateral prefrontal cortex plays a critical role in spontaneous categorization of species-specific vocalizations". *Journal of Cognitive Neuroscience* 17 (9), pp. 1471–1482 (cit. on pp. 2, 3, 12, 13, 27, 43, 63, 109).

Gollisch, T. and Meister, M. (2008). "Rapid neural coding in the retina with relative spike latencies". *Science* 319(5866), pp. 1108–1111 (cit. on p. 20).

Griffith, V. and Koch, C. (2014). "Quantifying Synergistic Mutual Information". In: *Guided Self-Organization: Inception*. Vol. Part 2, 9. Emergence, Complexity and Computation. Springer Berlin Heidelberg. Chap. 6, pp. 159–190 (cit. on p. 31).

Gross, C., Rocha-Miranda, C., and Bender, D. (1972). "Visual Properties of Neurons in Inferotemporal Cortex of the Macaque". *Journal of Neurophysiology* 35(1), pp. 96–111 (cit. on p. 9).

Gross, C. G., Bender, D. B., and Mishkin, M. (1977). "Contributions of the Corpus Callosum and the anterior commissure to visual activation of inferior temporal neurons". *Brain Research* 131, pp. 227–239 (cit. on p. 9).

Gu, Y., Liu, S., Fetsch, C. R., Yang, Y., Fok, S., Sunkara, A., DeAngelis, G. C., and Angelaki, D. E. (2011). "Perceptual learning reduces interneuronal correlations in Macaque Visual cortex". *Neuron* 71, pp. 750–761 (cit. on p. 109).

Gütig, R., Gollisch, T., Sompolinsky, H., and Meister, M. (2013). "Computing Complex Visual Features with Retinal Spike Times". *PLoS ONE* 8(1), e53063 (cit. on p. 21).

Gütig, R. (2012). "The multi-class tempotron: a neuron model for processing of sensory streams". In: *Cosyne Abstracts 2012, Salt Lake City USA* (cit. on p. 113).

Gütig, R. (2013). "Self-supervised neuronal processing of continuous sensory streams". In: *Cosyne Abstracts 2013, Salt Lake City USA* (cit. on p. 28).

Gütig, R. (2014). "Unsupervised emergence of continuous tuning curves and sensory maps in spiking neural networks". In: *Cosyne Abstracts 2014, Salt Lake City USA* (cit. on p. 28).

Gütig, R. and Sompolinsky, H. (2006). "The tempotron: a neuron that learns spike timing-based decisions". *Nature Neuroscience* 9, pp. 420–428 (cit. on pp. 12, 16, 17, 20, 21).

Gütig, R. and Sompolinsky, H. (2009). "Time-Warp-Invariant Neuronal Processing". *PLoS Biology* 7 (7), e1000141 (cit. on pp. 21, 29, 69).

Henry, G., Dreher, B., and Bishop, P. (1974). "Orientation specificity of cells in cat striate cortex". *Journal of Neurophysiology* 37(6), pp. 1394–1409 (cit. on p. 65).

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Westview Press (Perseus Books) (cit. on pp. 16, 19, 36, 80).

Hillenbrand, J., Getty, L. A., Clark, M. J., and Wheeler, K. (1994). "Acoustic characteristics of American English Vowels". *The Journal of the Acoustical Society of America* 97 (5-1), pp. 3099–30111 (cit. on p. 69).

Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). "Improving neural networks by preventing co-adaptation of feature detectors". *CoRR* 1207.0580 (cit. on p. 6).

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". *Signal Processing Magazine, IEEE* 29 (6), pp. 82–97 (cit. on p. 70).

Hubel, D. H. and Wiesel, T. N. (1962). "Receptive fields, binocular interaction and functional architecture in cat's visual cortex". *Journal of Physiology* 160, pp. 106–154 (cit. on p. 6).

Hulme, S. R., Jones, O. D., Raymond, C. R., Sah, P., and Abraham, W. C. (2013). "Mechanisms of heterosynaptic metaplasticity". *Philosophical Transactions of the Royal Society B* 369, p. 20130148 (cit. on pp. 41, 107).

Hung, C., Kreiman, G., Poggio, T., and DiCarlo, J. (2005). "Fast readout of object identity from Macaque inferior temporal cortex". *Science* 310, pp. 863–866 (cit. on pp. 1, 6).

Ison, M. J., Mormann, F., Cerf, M., Koch, C., Fried, I., and Quian Quiroga, R. (2011). "Selectivity of pyramidal cells and interneurons in the human medial temporal lobe". *Journal of Neurophysioloy* 106, pp. 1713–1721 (cit. on p. 10).

Jacobs, R. A. (1999). "Computational studies of the development of functionally specialized neural modules". *Trends in Cognitive Sciences* 3 (1), pp. 31–38 (cit. on p. 35).

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). "Adaptive Mixture of Local Experts". *Neural Computation* 3, pp. 79–87 (cit. on pp. 5, 23, 35).

Johansson, R. and Birznieks, I. (2004). "First spikes in ensembles of human tactile afferents code complex spatial fingertip events". *Nature Neuroscience* 7, pp. 170–177 (cit. on p. 20).

Jordan, M. (1994). "Hierarchical Mixture of Experts and the EM algorithm". *Neural Computation* 6 (2), pp. 181–214 (cit. on p. 6).

Kheradpisheh, S. R., Sharifizadeh, F., Nowzari-Dalini, A., Ganjtabesh, M., and Ebrahimpour, R. (2014). "Mixture of feature specified experts". 20, pp. 242–251 (cit. on p. 6).

Kietzmann, T., Swisher, J., König, P., and Tong, F. (2012). "Prevalence of selectivity for mirror-symmetric views of faces in the ventral and dorsal visual pathways". *Journal of Neuroscience* 32, pp. 11763–11772 (cit. on p. 9).

King, A. J. and Nelken, I. (2009). "Unraveling the principles of auditory cortical processing: can we learn from the visual system?" *Nature Neuroscience* 12 (6), pp. 698–701 (cit. on p. 11).

Kobatake, E. and Tanaka, K. (1994). "Neuronal Selectivities to complex object features in the ventral visual pathway of the Macaque cerebral cortex". *Journal of Neurophysiology* 71(3), pp. 856–867 (cit. on pp. 9, 110).

Kourtzi, Z. and DiCarlo, J. (2006). "Learning and neural plasticity in visual object recognition". *Current Opinion in Neurobiology* 16, pp. 1–7 (cit. on pp. 8, 109).

Kreiman, G., Koch, C., and Fried, I. (2000). "Category-specific visual responses of single neurons in the human medial temporal lobe". *Nature Neuroscience* 3 (9), pp. 946–953 (cit. on p. 9).

Kreiman, G. (2013). "Computational Models of Visual Object Recognition". In: *Principles of Neural Coding*. CRC Press (cit. on pp. 1, 6).

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., pp. 1097–1105 (cit. on p. 7).

Krogh, A. and Vedelsby, J. (1995). "Neural Network Ensembles, Cross Validation, and Active Learning". In: *Advances in Neural Information Processing Systems*. MIT Press, pp. 231–238 (cit. on pp. 2, 4, 63, 108).

Kumar, P., Jakhanwal, N., Bhowmnick, A., and Chandra, M. (2011). "Gender classification using pitch and formants". In: *Conference: Proceedings of the 2011 International Conference on Communication, Computing and Security, ICCS* (cit. on p. 69).

Kuncheva, L. (2004). *Combining Pattern Classifiers - Methods and Algorithms*. First Edition. Wiley-Interscience (cit. on pp. 4, 5).

Kusmierek, P., Ortiz, M., and Rauschecker, J. P. (2012). "Sound-identity processing in early areas of the auditory ventral stream in the macaque". *Journal of Neurophysiology* 107, pp. 1123–1141 (cit. on pp. 11, 105).

LeCun, Y. (2012). "Learning invariant feature hierarchies". In: *Proc. 12th European Conf. Comput. Vision*, pp. 496–505 (cit. on pp. 6, 110).

LeCun, Y., Bengio, Y., and Hinton, G. (2015). "Deep learning". *Nature* 521, pp. 436–444 (cit. on p. 7).

Ley, A., Vroomen, J., and Formisano, E. (2014). "How learning to abstract shapes neural sound representations". *Frontiers in Neuroscience* 8 (132), pp. 1–11 (cit. on p. 12).

Liu, Y. and Yao, X. (1999). "Ensemble learning via negative correlation". *Neural Networks* 12, pp. 1399–1404 (cit. on p. 5).

Maadeed, S. A. and Hassaine, A. (2014). "Automatic prediction of age, gender , and nationality in offline handwriting". *ERASIP Journal on Image and Video Processing* 10, pp. 1–10 (cit. on p. 85).

Marre, O., Botella-Soler, V., Simmons, K., Mora, T., Tkacik, G., and Berry, M. I. (2015). "High Accuracy Decoding of Dynamical Motion from a Large Retinal Population". *PLoS Comput Biol* 11(7), e1004304 (cit. on p. 31).

Masoudnia, S., Ebrahimpour, R., and Arani, S. A. A. A. (2012). "Incorporating of a regularization term to control negative correlation in mixture of experts". *Neural Processing Letters* 36, pp. 31–47 (cit. on p. 6).

Mayr, A., Binder, H., Gefeller, O., and Schmid, M. (2014). "The evolution of Boosting Algorithms". *Methods of Information in Medicine* 53 (6), pp. 419–427 (cit. on p. 4).

McCulloch, W. S. and Pitts, W. H. (1943). "A logical calculus of the ideas immanent in nervous activity". *Bulletin of Mathematical Biophysics* 5, pp. 115–133 (cit. on pp. 15, 16).

McKee, J. L., Riesenhuber, M., Miller, E. K., and Freedman, D. J. (2014). "Task dependence of visual and category representations in prefrontal and inferior temporal cortices". *The Journal of Neuroscience* 34 (48), pp. 16065–16075 (cit. on p. 11).

Milgram, J., Cheriet, M., and Sabourin, R. (2006). "One Against One or One Against All: Which One is Better for Handwriting Recognition with SVMs". In: *Proceedings of 10th International Workshop on Frontiers in Handwriting Recognition* (cit. on p. 84).

Miller, B. T. and D'Esposito, M. (2005). "Searching for "the Top" in Top-Down control". *Neuron* 48, pp. 535–538 (cit. on p. 10).

Mirghafori, N., Morgan, N., and Bourland, H. (1994). "Parallel Training of MLP probability Estimators for Speech Recognition: A Gender-Based approach". In: *Conference: Neural Networks for Signal Processing. Proceedings of the 1994 IEEE Workshop* (cit. on p. 69).

Moerland, P. (1997). *Some Methods for Training Mixtures of Experts* (cit. on p. 24).

Mohamed, A.-r., Dahl, G., and Hinton, G. (2012). "Acoustic Modeling using Deep Belief Networks". *IEEE Transactions on Audio, Speech and Language processing* 20 (1), pp. 13–22 (cit. on p. 70).

Montijn, J. S., Vinck, M., and Pennartz, C. M. A. (2014). "Population coding in mouse visual cortex: response reliability and dissociability of stimulus tuning and noise correlation". *Frontiers in Computational Neuroscience* 8 (58), pp. 1–15 (cit. on pp. 30, 62).

Mormann, F., Kornblith, S., Quian Quiroga, R., Kraskov, A., Cerf, M., Fried, I., and Koch, C. (2008). "Latency and Selectivity of Single Neurons indicate Hierarchical processing in the Human Medial Temporal Lobe". *Journal of Neuroscience* 28 (36), pp. 8865–8872 (cit. on p. 10).

Myers, E. B., Blumstein, S. E., Walsh, E., and Eliassen, J. (2009). "Inferior frontal regions underlie the perception of phonetic category invariance". *Psychological Science* 20 (7), pp. 895–903 (cit. on p. 11).

Ou, G. and Murphey, Y. L. (2007). "Multi-class pattern classification using neural networks". *Pattern recognition* 40, pp. 4–18 (cit. on p. 84).

Pepiot, E. (2013). *Voice, speech and gender: male-female acoustic differences and cross-language variation in English and French speakers* (cit. on pp. 69, 72).

Petkov, C. I., Kayser, C., Steudel, T., Whittingstall, K., Augath, M., and Logothetis, N. K. (2008). "A voice region in the monkey brain". *Nature Neuroscience* 11 (3), pp. 367–374 (cit. on p. 11).

Pfister, J.-P., Toyoizumi, T., Barber, D., and Gerstner, W. (2006). "Optimal Spike-Timing-Dependent Plasticity for Precise Action Potential Firing in Supervised Learning". *Neural Computation* 18, pp. 1318–1348 (cit. on pp. 21, 25, 94, 98).

Pinto, N., Cox, D., and DiCarlo, J. (2008). "Why is Real-World Visual Object Recognition Hard?" *PLoS Computational Biology* 4 (1), e27 (cit. on p. 1).

Purves, D., Augustine, G. J., Fitzpatrick, D., Hall, W. C., LaMantia, A.-S., McNamara, J. O., and White, L. E. (2008). *Neuroscience*. Fourth Edition. Sinauer Associates, Inc. (cit. on pp. 8, 12, 115).

Quian Quiroga, R. and Kreiman, G. (2010). "Measuring sparseness in the brain: comment on Bowers (2009)". *Psychological Review* 117 (1), pp. 291–299 (cit. on p. 10).

Quian Quiroga, R., Reddy, L., Koch, C., and Fried, I. (2007). "Decoding Visual Inputs from multiple neurons in the Human Temporal Lobe". *Journal of Neurophysiology* 98, pp. 1997–2007 (cit. on p. 10).

Quian Quiroga, R., Reddy, L., Kreiman, G., Koch, C., and Fried, I. (2005). "Invariant visual representation by single neurons in the human brain". *Nature* 435, pp. 1102–1107. (Cit. on pp. 10, 105, 109).

Rieke, F., Warland, D., Ruyter van Steveninck, R. de, and Bialek, W. (1999). *Spikes - Exploring the neural code*. Computational Neurosciences series. MIT Press (cit. on pp. 27, 31).

Riesenhuber, M. and Poggio, T. (2000). "Models of object recognition". *Nature Neuroscience* 3, pp. 1190–1204 (cit. on p. 6).

Riesenhuber, M. and Poggio, T. (2002). "Neural mechanisms of object recognition". *Current opinion in neurobiology* 12, pp. 162–168 (cit. on p. 6).

Rosenblatt, F. (1958). "The Perceptron: A probabilistic model for information storage and organization in the brain". *Psychological Review* 65 (6), pp. 386–408 (cit. on pp. 15, 79).

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Available at http://babel.hathitrust.org. Washington: Spartan Books (cit. on pp. 15, 19).

Roy, A. (2012). "A theory of the brain: localist representation is used widely in the brain". *Frontiers in Psychology* 3 (551), pp. 1–4 (cit. on p. 10).

Roy, A. (2013). "An extension of the localist representation theory: grandmother cells are also widely used in the brain". *Frontiers in Psychology* 4 (300), pp. 1–3 (cit. on p. 10).

Schafer, P. B. and Jin, D. Z. (2014). "Noise-Robust Speech Recognition Through Auditory Feature Detection and Spike Sequence Decoding". *Neural Computation* 26, pp. 523–556 (cit. on p. 70).

Schneidman, E., Bialek, W., and Berry Il, M. J. (2003). "Synergy, Redundancy, and Independence in Population Codes". *The Journal of Neuroscience* 23(37), pp. 11539–11553 (cit. on p. 31).

Sedaaghi, M. (2009). "A comparative study of Gender and Age classification in Speech signals". *Iranian Journal of Electrical and Electronic Engineering* 5 (1), pp. 1–12 (cit. on p. 70).

Serre, T., Oliva, A., and Poggio, T. (2007). "A feedforward architecture accounts for rapid categorization". *PNAS* 104 (15), pp. 6424–6429 (cit. on pp. 6, 7, 110).

Sesa-Nogueras, E., Faundez-Zanuy, M., and Roure-Alcobe, J. (2015). "Gender Classification by Means of Online Uppercase Handwriting: A Text-Dependent Allographic Approach". *Cognitive Computation* 4 (cit. on p. 85).

Seung, S., Lee, D., Reis, B., and Tank, D. (2000). "Stability of the Memory of Eye Position in a Recurrent Network of Conductance-Based Model Neurons". *Neuron* 26, pp. 259–271 (cit. on p. 112).

Seyfarth, R., Cheney, D., and Marler, P. (1980). "Monkey responses to three different alarm calls: Evidence of Predator classification and semantic communication". *Science* 210, pp. 801–803 (cit. on p. 1).

Shapire, R. (1990). "The Strength of Weak Learnability". *Machine Learning* 5.2, pp. 197–227 (cit. on p. 4).

Sharkey, A. (1996). "On Combining Artificial Neural Nets". *Connection Science* 8, pp. 299–313 (cit. on pp. 4, 5).

Shipp, C. A. and Kuncheva, L. (2002). "Relationships between combination methods and measures of diversity in combining classifiers". *Information Fusion* 3, pp. 135–148 (cit. on p. 30).

Sigala, N. and Logothetis, N. (2002). "Visual categorization shapes feature selectivity in the primate temporal cortex". *Nature* 415 (6869), pp. 318–320 (cit. on pp. 2, 10, 106).

Singer, W. (1999). "Time as Coding Space?" *Current Opinion in Neurobiology* 9, pp. 189–194 (cit. on p. 20).

Stanton, P. K. (1996). "LTD, LTP, and the sliding threshold for long-term synaptic plasticity". *Hippocampus* 6, pp. 34–42 (cit. on p. 107).

Tang, B., Heywood, M., and Shepherd, M. (2002). "Input partitioning to Mixture of Experts". In: *Proceedings of the 2002 International Joint Conference on Neural Networks* (cit. on p. 25).

Thorpe, S., Delorme, A., and Van Rullen, R. (2001). "Spike-based strategies for rapid processing". *Neural Networks* 14, pp. 715–725 (cit. on p. 20).

Thorpe, S., Fize, D., and Marlot, C. (1996). "Speed of processing in the human visual system". *Nature* 381, pp. 520–522 (cit. on p. 1).

Tian, B., Reser, D., Durham, A., Kustov, A., and Rauschecker, J. P. (2001). "Functional specialization in Rhesus monkey Auditory Cortex". *Science* 292, pp. 290–293 (cit. on pp. 11, 105).

Tomai, C. I., Kshirsagar, D. M., and Srihari, S. N. (2004). "Group Discriminatory Power of Handwritten Characters". In: *Proceedings of SPIE-IS&T Electronic Imaging*. Vol. 5681, pp. 504–515 (cit. on p. 85).

Tsao, D., Freiwald, W., Knutsen, T., Mandewille, J., and Tootell, R. (2003). "Faces and objects in macaque cerebral cortex". *Nature Neuroscience* 6(9), pp. 989–995 (cit. on p. 9).

Tsao, D. and Livingstone, M. (2008). "Neural mechanisms for face perception". *Annual Review of Neuroscience* 31, pp. 411–438 (cit. on p. 9).

Tsunada, J., Hoon Lee, J., and Cohen, Y. E. (2011). "Representation of speech categories in the primate auditory cortex". *Journal of Neurophysiology* 105, pp. 2634–2646 (cit. on p. 11).

Turrigiano, G. G. (2008). "The Self-Tuning Neuron: Synaptic Scaling of Excitatory Synapses". *Cell* 135, pp. 422–435 (cit. on pp. 38, 40, 41).

Urbanczik, R. and Senn, W. (2009). "Reinforcement learning in populations of spiking neurons". *Nature Neuroscience* 12, pp. 250–252 (cit. on pp. 7, 18, 21, 22, 25, 35, 45, 47, 89–92, 94–99, 102, 105, 107, 111, 117).

Valdez, A., Papesh, M., Treiman, D., Smith, K., Goldinger, S., and Steinmetz, P. (2015). "Distributed representation of visual objects by single neurons in the human brain". *Journal of Neuroscience* 35 (13), pp. 5180–5186 (cit. on p. 10).

Vergin, R., Farhat, A., and O'Shaughnessy, D. (1996). "Robust Gender-Dependent Acoustic-Phonetic Modeling In Continuous Speech Recognition Based On A New Automatic Male/Female Classification". In: *In Fourth International Conference on Spoken Language Processing*, pp. 1081–1084 (cit. on p. 70).

Xie, X. and Seung, S. (2004). "Learning in neural networks by reinforcement of irregular spiking". *Physical Review E* 69 (041909), pp. 1–10 (cit. on pp. 21, 94).

Yang, J., Zeng, X., Zhong, S., and Wu, s. (2013). "Effective neural network ensemble approach for improving generalization performance". *IEEE Transactions on neural networks and learning systems* 24 (6), pp. 878–887 (cit. on p. 6).

Yang, S. and Browne, A. (2004). "Neural network ensembles: combining multiple models for enhanced performance using a multistage approach". *Expert Systems* 21 (5), pp. 279–288 (cit. on p. 6).

Yuksel, S. E., Wilson, J. N., and Gader, P. D. (2012). "Twenty years of mixture of experts". *IEEE Transactions on neural networks and learning systems* 23.8, pp. 1177–1193 (cit. on pp. 6, 25).

Zoccolan, D., Kouh, M., Poggio, T., and DiCarlo, J. (2007). "Trade-Off between object selectivity and tolerance in Monkey Inferotemporal Cortex". *The Journal of Neuroscience* 27 (45), pp. 12292–12307 (cit. on p. 110).

ACRONYMS

BCM	Bienenstock-Cooper-Munro
(E/I)PSP	(Excitatory/Inhibitory) postynaptic potenial
KL	Kullback-Leibler
LTD	Long-term depression
LTP	Long-term potentiation
ME	Mixture of experts (model)
SMW	SumMinusWhole

DANKSAGUNG

Auf meinem bisherigen Weg durch die Promotion haben mich so viele Menschen begleitet, ohne die ich diese Arbeit niemals hätte fertigstellen können.

Mein erster Dank gilt natürlich meinem Betreuer Robert Gütig. Ich kenne keinen anderen Menschen, der mit so viel Begeisterung und Akribie für ein wissenschaftliches Thema brennen kann. Vielen Dank, dass du mich mit dieser Begeisterung auch etwas anstecken konntest und mir zu jeder Zeit mit gutem Rat und immensem Ideenreichtum zur Seite standest.

Großer Dank gilt ebenfalls den Mitgliedern meines Betreuungskomitees, Tim Gollisch und Fred Wolf. Die konstruktive Kritik in den regelmäßigen Treffen hat mir immer wieder geholfen, den Blick auf das Wesentliche nicht zu verlieren. Ebenfalls danken möchte ich an dieser Stelle Tobias Moser, Alexander Gail und Thomas Kneib für den sehr unkomplizierten Kontakt und die Bereitschaft, in meiner Prüfungskommission mitzuwirken.

Mehr als drei Jahre täglich am Computer zu verbringen ist nur in einer angenehmen Arbeitsatmosphäre aushaltbar. Dass dies der Fall war, habe ich insbesondere meinem Bürokollegen Rafael zu verdanken. Eiskaffee- und Schokopausen haben sich hier als Wundermittel zum kurzzeitigen Frustabbau herausgestellt. Tausend dank gilt Rafael aber auch wegen seiner unerschütterlichen Geduld, mir zum x-ten Mal Funktionsweisen von Linux zu erklären oder - im Fachjargon gesprochen - 'meinen Computer zu reparieren'.
Thanks a lot also to Qiang for helping me with my attempts to convert my programs to C - but not for gaining way too much expertise in the Doppelkopf games that did not leave us much of a chance to win ;)

Auch wenn es fachlich kaum Überschneidungen gibt, möchte ich hier ebenfalls der AG Statistik/Ökonometrie danken. In keiner anderen Arbeitsgruppe hätte man sich so zugehörig fühlen koennen, obwohl man schon lange nicht mehr dazu gehörte. Danke, dass ich trotz meiner Abtrünnigkeit noch an diversen Kanu-, Wander- und Fahrrausflügen, Weihnachtsfeiern und Kohlfahrten teilnehmen durfte. Obwohl mich natürlich wurmt, dass ich nicht einmal den begehrten Titel des Büro des Monats für meine 'Aussenstelle' ergattern konnte.
Den größten Anteil an diesem Zugehörigkeitsgefühl hat sicherlich mein ehemaliger Betreuer Thomas Kneib. Tausend Dank, dass du mir auch nach meinem Wechsel noch das Gefühl gegeben hast, dass ich jederzeit mit deiner Unterstützung rechnen konnte, und 'nur das Beste' von dir nicht nur nach einer Floskel klingt.
Klarerweise möchte ich natürlich auch Elisabeth Waldmann nicht unerwähnt lassen. Danke, dass du mir jederzeit -die ersten Jahre persönlich im Büro oder in der WG,

später dann hauptsächlich per Facebook- für ausführliches Woiseln zur Verfügung standest. Die Wunderwirkung einer ausgiebigen Mittwochsdisko zu Baba Yetu wissen wirklich nur die Besten zu schätzen.
Ihr, Lukas, Rafael, Fernando und Thomas gilt zusätzlich ein ganz besonderer Dank für grandioses Korrekturlesen.

Da ein Doktorand von Wissenschaft alleine nicht leben kann, gilt ein großer Dank auch Stefan Treue und dem NEUROSENSES Promotionsprogramm sowie der Max-Planck-Gesellschaft für die finanzielle Unterstützung während meiner Promotion. Ebenfalls danken möchte ich der GGNB und speziell dem Promotionsprogramm SMN für die großartige Organisation und die inspirierenden Retreats.

Wissenschaftliches Arbeiten bedeutet nicht nur Forschen im Elfenbeinturm, sondern auch ein reger Austausch auf Fachtagungen. Vielen Dank an all die tollen Menschen, die ich während meiner Promotion auf interessanten Konferenzen kennen lernen durfte - stellvertretend dafür nenne ich einmal den Mayr Andy und die Ferienwohnungs-Crew. Die Klassenfahrten mit euch waren nicht nur wissenschaftlich ein Highlight.

Singen befreit die Seele und kann über jegliche Frustration hinweg helfen. Ich danke meinem Chor Unicante für eine Vielzahl solch befreiender Momente sowie für die wunderbaren Menschen, die ich dabei kennen gelernt habe. Durch ihn und sie habe ich mich in Göttingen immer mehr zuhause gefühlt.

Vielen Dank natürlich auch an alle anderen Menschen, die mich in den letzten Jahren begleitet und aufgemuntert haben: Entspannte Joggingrunden mit Volkert, Patricia, Katha und Alicia haben die Zeit ebenso angenehm gestaltet wie Doko-Donnerstage mit Simone, Benji und Alex und lange Sommernächte auf dem Willi mit meinen Chormädels und -jungs - und wer immer noch so bereit war, Zeit zu verschwenden.

Selbstverständlich wäre ich nicht an diesem Punkt ohne die großartige Unterstützung meiner Familie und insbesondere meiner Eltern. Danke, dass ihr immer an mich geglaubt habt und in schwierigen Situationen immer wusstet, die Laune mit Vitaminpaketen und Zimtsternketten aufzubessern. Eine bessere Familie kann man gar nicht haben. Neben den besten Eltern habe ich zusätzlich das Glück, die besten Schwiegereltern erwischt zu haben. Vielen lieben Dank auch an Marina und Uli, dass ihr mich so gut aufgenommen und in dieser Zeit unterstützt habt.

Mein größter Dank gilt natürlich meinem Mann. Tausend Dank dir, Hilli, dass du trotz der räumlichen Entfernung immer für mich da warst und ich bei dir immer zur Ruhe kommen konnte. Die Aussicht, nach Abschluss dieser Promotion wieder bei dir sein zu können, war die grösste Motivation, die ich haben konnte.

CURRICULUM VITAE

JULIA HILLMANN

18.03.1986 Born in Karlsruhe, Germany

EDUCATION:

2005–2008 Bachelor's degree in Mathematics (Focus on Biomathematics),
at the Carl-von-Ossietzky University Oldenburg, Germany

08/2007-01/2008 ERASMUS Exchange Semester at Linköpings universitet (LiU), Sweden

2008–2010 Master's degree in Mathematics (Focus on Applied Mathematics),
with Prof. Dr. Thomas Kneib (advisor)
at the Carl-von-Ossietzky University Oldenburg, Germany
Thesis title: "Categorical Regression Models for the Analysis of Neural Data"

12/2010-04/2012 Research Fellow,
with Prof. Dr. Thomas Kneib (advisor)
at the Carl-von-Ossietzky University Oldenburg, Germany
and the Georg-August-Universität Göttingen, Germany
Project title: "Statistical Learning Procedures for the Classification of Spike Trains"
Fellowship from NEUROSENSES PhD training in integrative neurosensory sciences

SINCE 05/2012 PhD student,
with Dr. Robert Gütig (advisor)
in the program of Sensory and Motor Neuroscience (GGNB)
at the Georg-August-Universität Göttingen, Germany
Fellowship from NEUROSENSES PhD training in integrative neurosensory sciences

PUBLICATIONS:

2014 "A bivariate cumulative probit model for the analysis of nerve cell responses",
J. Hillmann, T. Kneib, L. Koepcke, L.M. Juarez Paz and J. Kretzberg.
Biometrical Journal, 56(1), 23-43.

COLOPHON

This document was typeset in LaTeX using the typographical look-and-feel `classicthesis`. The bibliography is typeset using `biblatex`.