WORDT
NIET UITGELEEND

Hylke Kleve
h.w.kleve@student.rug.nl

Using re-entrant mapping in neural network ensembles

Master's thesis

Submitted to the department of Computer Science
Rijksuniversiteit Groningen
Groningen, The Netherlands
April, 2004

Jan Jacobs
Supervisor at Océ-Technologies B.V.

Sebastian de Smet
Supervisor at Océ-Technologies B.V.

Jos Nijhuis
Supervisor at the university of Groningen

Rijksuniversiteit Groningen
Bibliotheek FWN
Nijenborgh 9
9747 AG Groningen

The writer was enabled by Océ-Technologies B.V. to perform research that partly forms the basis for this report. Océ-Technologies B.V. does not accept responsibility for the accuracy of the data, opinions and conclusions mentioned in this report, which are fully for the account of the writer.

WORDT
NIET UITGELEEND

Abstract

Combining multi-modal information can ease classification. Neurobiologist Gerald Edelman postulated a theory on how this kind of information is processed in the brain. He concluded that small local groups of neurons, each from within the context of their own domain, build up connections to each other when activated together. As a result, activation in one area results in activation in connected areas; this mechanism of association can be used for recognition.

This study investigates how this neurobiological model can be translated into a computational model and whether combining artificial neural networks this way will improve classification. Currently, neural network ensembles are often employed to combine networks. The ensemble techniques will play a key role in the implementation. An extension of these techniques makes it possible to benefit from unlabelled data.

The resulting system shows an improved classification accuracy, with reduced labelled data required for learning.

Contents

Rijksuniversiteit Groningen
Bibliotheek FWN
Nijenborgh 9
9747 AG Groningen

Chapter 1

Introduction

Océ-Technologies is a copying and printing specialist for the professional market. It has a full range of products that covers the entire spectrum of imaging needs, such as productive high-volume copying, high-performance printing and wide-format copying and printing. Océ-Technologies doesn't only focus on hardware, but also on software services and printing solutions.

Within the research group at Research & Development I participated in, currently the main focus is on making software more flexible and less labour-intensive. The usage of artificial neural networks fits this idea. By making use of neural networks, the explicit formulating of algorithms becomes obsolete. However, in order to address the performance demands, research is needed on improving the existing networks.

Recent studies [Swier, 2001] show that using a multiple classifier system can improve recognition rates in a document analysis setting. In a multiple classifier system, several experts predict on low dimensional information. The system as a whole has to make the same prediction as the experts. Due to statistical properties, the results of the experts can be combined to form a better prediction. An essential part of the multiple classifier system is therefore that which combines individual results. The way these classifiers are combined will be crucial for overall performance. These studies demonstrated a lack in the current combination schemes. This is the reason for further investigation in how classifiers can be combined.

1.1 Problem description

Humans have the ability of linking objects and concepts to each other, thereby performing categorisation; this is fundamental to human nature. How our brains

give rise to categorisation has been subject to many studies over time. The world, as we perceive it, consists of many features like size, shape, colour, and so on. Furthermore, we usually perceive things through a multitude of modalities. This provides us a way to check the consistency of what we perceive. More important, it gives us the ability to make better judgements about it.

The neurobiologist Gerald Edelman describes in his work [Edelman, 1993] a way of how this integration of information is done in the brain. He states that the brain can be divided into maps, where each map consists of local neuron groups. Each group responds to a certain feature. Functional segregated groups are interconnected by re-entrant links. These connections are abundant in number and perform a crucial role in recognition and learning.

This model provides us a different way to think about the integration of information in the brain. Furthermore, it shows how the brain integrates multiple modalities, similar to a multiple classifier system. We will investigate if this model can be applied to the development and integration of classifiers. After this, we will look at the benefits of this approach concerning the classification scores.

1.2 Research goals

The main objective in this thesis will be the exploration of a different method for combining artificial neural network classifiers. As re-entrant links are responsible for combining several functional related maps in the brain, this resembles the combining that takes place in a neural network ensemble. Ensemble systems have shown to be useful in a number of situations.

We will try to find a way to use Edelman's ideas on how the human brain works in a computer model. Furthermore, we will use it on a problem in document analysis to benchmark its performance. Hopefully, this will provide us a way to take advantage of the performance benefits as seen in a neural network ensemble, while attaining the software maintainability of a single classifier.

1.3 Overview

This thesis will discuss how the neurobiological theory on re-entrant mapping can be translated into a computational algorithm.
This document will be organised as follows:

- Chapter 2 provides an outline on biological neurons, artificially constructed neural networks, and the concerns with large input spaces in the section

on ensembles of classifiers.

- Chapter 3 introduces the concepts on information processing with re-entrant mapping; we will look at how brain parts connect for global perception.
- Chapter 4 surveys the current algorithms that exhibit key properties of re-entrant mapping.
- Chapter 5 provides some experiments using one of these algorithms, illustrating a possible application of re-entrant mapping with neural network ensembles.
- Chapter 6 ends this thesis with conclusions and further directions.

Chapter 2

Neural networks

The human brain is composed of a vast amount of neurons. On the order of 10^{11} neurons [Kandel et al., 2000] play a major role in learning. Neurons communicate with each other through synapses. Each neuron consists of a soma, also called a *body*, and a set of incoming and outgoing synapses (figure 2.1a). The incoming channels are called dendrites and the outgoing channels axons. Typically, a neuron receives information through its dendrites from approximately thousand synapses. In turn, the neuron's axon makes synaptic connections with thousand other neurons.

Neurons communicate with each other through firing action potentials, by releasing neurotransmitters (chemical substances). The release of neurotransmitters is based on a spatio-temporal sum of the many incoming synapses. Some synapses excite neurons and cause them to generate signals called action potentials, large transient voltage changes that propagate down their axons. Other synapses are inhibitory and prevent the neuron from generating action potentials. The action potential propagates down the axon to the sites where the axon has made synapses with the dendrites of other nerve cells.

In the cerebral cortex, neurons form groups in columns. These columns appear to respond to certain features. Cortical connectivity between columns is known to be relatively sparse. Mostly, neurons tend to communicate with other neurons belonging to the same functional group. The inter-communication of functional groups will be discussed in the section about re-entrant mapping.

When we look at the human brain this way, we can view it as a densely interconnected, parallel structure that processes information. Extensive and elaborate neural circuits form, where neuronal signals pass from neuron to neuron.

Computational models have been inspired by the way these neural circuits

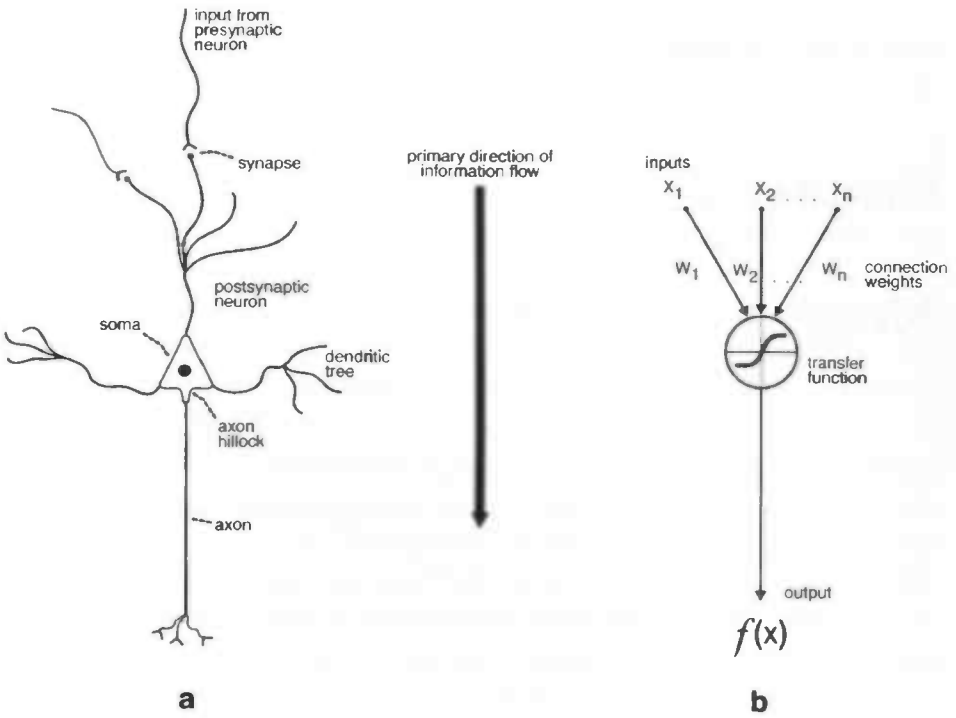

Figure 2.1: Modeling a neuron: (a) biological neuron, including neural processes (b) artificial neuron [Sajda, 2001].

work. These artificial neural networks exhibit the same robust learning capabilities. This chapter will provide an introduction to some of these models.

2.1 Artificial neurons

Discrete neuron models in computer science have been around since the last decades. When working together, these artificial neurons form a paralleldistributed processing system, similar to their biological counterpart. Such a system is called an artificial neural network (ANN). These systems are commonly used for pattern recognition problems and approximation problems.

The underlying mathematical models exhibit similar properties to adaptive biological learning. Artificial neurons are tied together with weights analogous to the synapses connecting biological neurons (see figure 2.1 for comparison). The weights determine the receptive field of a neuron. Learning typically involves adjusting this field. The connection weights are iteratively adjusted during the

training phase, minimising a predefined error function. The weights thus store the knowledge necessary to solve specific problems.

Learning can be based on function approximation, or training by example. Networks of the latter kind need to find the right output class for a sample. The networks of the former kind learn the function between input and output neurons. These networks can also be used as a classifier by learning the probability density of the output classes for the samples.

We will study the Kohonen self-organising map and Learning Vector Quantisation next. In our work, the computational model of Kohonen will take a prominent place because it shares some remarkable features with biology. Hereafter, the usage of a modular structure for large amounts of input information is discussed.

2.2 Kohonen self-organising map

The self-organising map [Kohonen, 1997] is a computational model that is popularised by Kohonen. It is considered to be biological plausible. A self-organising map (SOM) has a typical network structure of two layers of neurons (figure 2.2). The first layer is the input layer and the second is the self-organising layer.

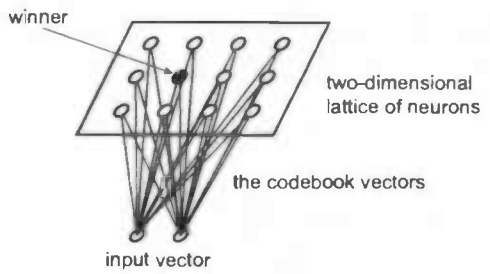

Figure 2.2: A self-organising map is fed by its input layer $\vec{x}$. The self-organising layer maps this information topologically. [Haykin, 1998]

The first input layer represents the stimulus of the network in the form of a feature vector $\vec{x}$.

$$\vec{x} = [x_1, x_2, \ldots, x_n]^T \tag{2.1}$$

The second layer acts upon this in a winner-takes-all fashion. This means that one neuron in this layer is declared 'the winner'. The neurons all receive a weighted input from the input layer. Each neuron in the input layer is connected

to each neuron of the self-organising layer through a weight. This means the neurons from the second layer have a weight 'vector' whose dimension equals that of the input vector. We can thus see this weight vector as another vector from the input space; we will call this a codebook vector.

$$m_j = [w_{j1}, w_{j2}, \ldots, w_{jn}]^T, \qquad j = 1, 2, \ldots \tag{2.2}$$

where w_{ji} is the weight between neuron j and the ith component of the input vector.

Another interesting property of the self-organising layer is its organisation as a map. The neurons are arranged in a lattice (as depicted in figure 2.2). Aside from a distance measure between different codebook vectors, also a measure of distance in this lattice is used. From now on, this will be referred to as the *map distance*. Usually, a two dimensional lattice is used with the euclidean distance function.

When the input vector $\vec{x}$ is presented, the neuron k, whose codebook vector, $\vec{m}_k$, resembles this vector most, will be declared the winner. Normally, the Euclidean distance is used as a measure of resemblance. In this case,

$$k = \arg \min_j \|\vec{x} - \vec{m}_j\| \tag{2.3}$$

Then, the winner adapts itself to the input by changing its codebook vector to resemble the input vector more. Furthermore, the winner will stimulate his neighbours, although to a lesser degree, to adapt their codebook vectors too.

$$\vec{m}_i(t+1) = \vec{m}_i(t) + h_{k,i}(t)[\vec{x}(t) - \vec{m}_i(t)] \tag{2.4}$$

The neighbourhood function $h_{k,i}$ determines to which degree the neurons in the neighbourhood are adjusted. This neighbourhood function depends on the map distance $d_{k,i}$ between neuron i and the winning neuron k. Usually a gaussian neighbourhood is used:

$$h_{k,i}(t) = \alpha(t) \exp\left(-\frac{d_{k,i}}{2\sigma^2(t)}\right) \tag{2.5}$$

During training, two functions in time determine the character of the neighbourhood function. The first is the learning rate $\alpha(t)$, the degree to which the winner and its neighbourhood adjust their weights. The second function controls the size of the neighbourhood $\sigma(t)$. As the training progresses, the size of this neighbourhood shrinks. This enables the specialisation of the neuronal receptive fields. The gradual decrease of the learning rate and neighbourhood ensures stability of the network. The training ends when the learning rate drops below a certain threshold.

2.2.1 Properties

The codebook vectors of the Kohonen self-organising map will converge to approximate the probability density function $f(x)$ of the input data. Training minimises the average expected squared distance between the input vector and the codebook vector of its corresponding winner. This is expressed in the following estimator:

$$E = \int \|\vec{x} - \vec{m}_{winner}\|^2 f(x)dx \tag{2.6}$$

A neurons receptive field includes all samples to which it will have the shortest distance, i.e. declare it the winner.

Because not only the winner but also it neighbourhood is updated during training, the neurons form a topological map. This means that two close stimuli will activate adjacent neurons. This topographic organisation is also found in many parts of the brain [Kandel et al., 2000] and is considered beneficial for the abstraction of information [Zrehen and Gaussier, 1994] as it groups neurons that are similar.

2.3 Learning vector quantisation

As stated before, the codebook vectors of the Kohonen self-organising map converge to approximate the probability distributions. This minimises the piecewise distance to each input vector (equation 2.6). Since no explicit class information is provided during the learning stage, boundaries are not optimally set. Figure 2.3 shows a Kohonen self-organising map, comprising of two neurons.

The crosses mark the positions of the codebook vectors; the circles show the corresponding data points. This shows how the codebook vectors are placed to minimise the distance to their surrounding data vectors. The colours of the circles indicate their class. As data samples activate the neuron with the nearest codebook vector, the line in the middle indicates the (linear) decision boundary of the receptive fields. Without the help of class information, the codebook vectors cannot be placed to separate the classes. Only with the use of class information, the boundaries of the codebook vectors can be optimised; this can only happen with some sort of supervision.

Kohonen developed a family of algorithms especially for the purpose of class separation [Kohonen, 1997]. These supervised algorithms are called Learning Vector Quantisation. Especially designed for decision-making tasks, these algorithms optimise border placement. We will encounter one of these LVQ algorithms, namely LVQ2.1.

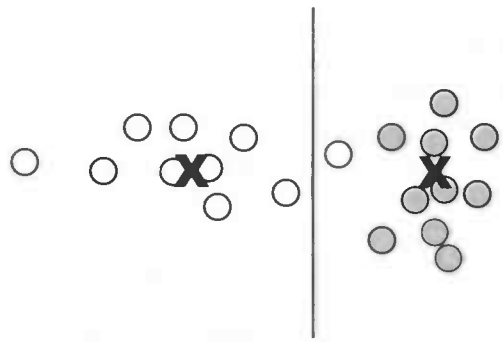

Figure 2.3: A projection of the position of codebook vectors, samples and decision boundary for two neurons. The codebook vectors are placed according to the density approximating convergence criteria of Kohonen's SOM. For the purpose of class separation the codebook vectors are placed sub optimal. [de Sa, 1994]

In this algorithm the winner and its runner-up, $\vec{m}_i$ and $\vec{m}_j$, will change their weight vector. Note that these neurons are both associated with a category, an output class. When one of these categories (not both) is the true category of the sample, the associated codebook vectors are changed:

$$\vec{m}_i(t+1) = \vec{m}_i(t) - \alpha(t)[\vec{x}(t) - \vec{m}_i(t)] \tag{2.7}$$

$$\vec{m}_j(t+1) = \vec{m}_j(t) + \alpha(t)[\vec{x}(t) - \vec{m}_j(t)] \tag{2.8}$$

Here, $\vec{m}_i$ is the codebook vector associated with the right output class. Adjustments are only made when the sample falls within a *window* from the decision border between $\vec{m}_i$ and $\vec{m}_j$. This is for certain ratios of d_i and d_j, respectively the euclidean distance between the sample $\vec{x}$ and codebook vectors $\vec{m}_i$, $\vec{m}_j$.

$$min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s, \qquad \text{where } s = \frac{1-w}{1+w} \tag{2.9}$$

These algorithms abandon the concept of a topological map by not updating the neighbourhood; only the winner and the runner-up are updated.

2.4 Neural networks ensembles

When the classification scores of a neural network need to improve, classifiers must either deal more efficient with their input or use more information to make

discrimination borders more explicit. Usually, the performance of a classifier is bound by its algorithm. This is why we cannot expect dramatic classification improvements here. However, an increase of the input information seems reasonable, as the more relevant patterns one obtains the better decisions one can make. In practice, this is partially true. Studies have shown that finding clusters in higher dimensionality becomes increasingly difficult [Jain et al., 1999]. A possible explanation follows.

As the weights in a classifier are tuned through successive input of samples, each weight can only reach an (sub) optimal value after sufficient number of training samples. Each extra weight, for example necessary when the input dimensionality is raised, obliges the training set for a proportional expansion. In order to set the weights optimal, a specific ratio between weights and training samples is needed. A bad weights-to-samples ratio usually results in poor classification scores. This is why there is an optimum to the number of training samples for a specific neural network; this is called the *peaking phenomenon*. Consequently, it may not always be useful to add the number of features.

The peaking phenomenon can be prevented by using multiple classifiers. Input information then has to be split into lower order feature vectors. Each classifier can then be trained with an optimal weights-to-samples ratio. The classification results of this ensemble of classifiers can then be combined using a combination scheme. This set up has shown to improve the overall classification score in numerous experiments. Figure 2.4 illustrates a model of a neural network ensemble.

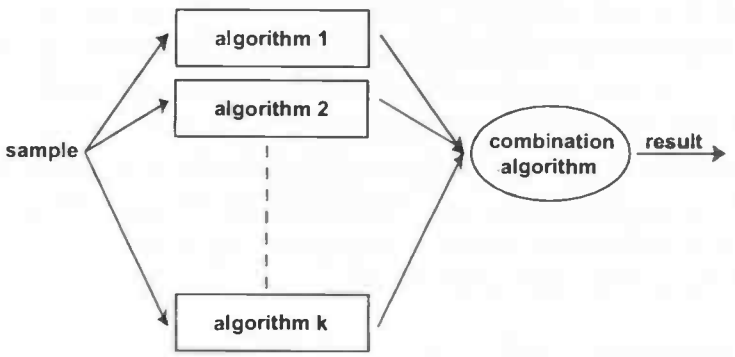

Figure 2.4: The structure of a neural network ensemble.

As each classifier categorises a sample into categories, it encounters overlap in categories. Because the classifier parameterises the input data, the data

gradually changes from one category into another (figure 2.3). If other classifiers perform a different parameterisation or use different data, this might lead to other decision boundaries in the neighbourhood of the input sample. By combining the different results of the classifiers, the variance of the decision boundaries will be reduced. Furthermore, the local experts will have learnt their decision boundaries more accurately because of an improved weights-to-samples ratio.

Currently, there are many ways for combining the algorithms. Popular choices are voting rules, belief functions, statistical techniques, Dempster-Shafer evidence theory, and other integration schemes [Jain et al., 1999]. They all have in common that they combine classifier results with a minimum number degrees of freedom, sustaining the improved weights-to-samples ratio.

Of course, the recognition rates will only improve when the classifiers make different errors. The combination schemes should aim at exploiting local expertise of classifiers on certain input domains. However, current combination schemes are not powerful enough for getting maximal recognition results according to [Swier, 2001].

We will consider two important combination schemes, as they will be used further on.

2.4.1 Majority voting

Here, all the classifiers provide their best estimation of the output class. The combination algorithm then uses the most frequent output class amongst the classifiers as a final prediction. Some examples will illustrate situations that benefit from this combination algorithm and those that don't. In these examples, a one denotes the correct prediction and a zero the incorrect one. An example that illustrates this combination scheme is shown in figure 2.5.

In some situations the majority-voting algorithm fails to use the expertise in the different classifiers (figure 2.6). This occurs even though both examples show that there is at least one expert at a time that can predict the right category. None of the current combination schemes is able to solve such problems.

2.4.2 Weighted voting

The weighted voting algorithm computes a linear weighted combination of the base classifiers. As each classifier computes a prediction of the sample's position in the output class space O, they are functions $f_j : \mathbb{R}^n \rightarrow O$ (f_j is a base classifier). Frequently, this is replaced by a series of confidence values for each

sample	algorithm 1	algorithm 2	algorithm 3	majority-voting
1	0	1	1	1
2	0	1	1	1
3	0	1	1	1
4	1	0	1	1
5	1	0	1	1
6	1	0	1	1
7	1	1	0	1
8	1	1	0	1
9	1	1	0	1

Figure 2.5: Performance of majority voting is better than the performance of each of the algorithms. [Swier, 2001]

sample	algorithm 1	algorithm 2	algorithm 3	majority-voting
1	0	0	1	0
2	1	0	0	0
3	0	1	0	0
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	1	1	1

Figure 2.6: Performance of majority voting is worse than the performance of each of the algorithms. [Swier, 2001]

class; this is essentially the same. The combination algorithm calculates the weighted sum of these base predictions:

$$F(x) = \sum_{j \in J} w_j \cdot f_j(x) \tag{2.10}$$

The weights w_j ($j \in J$) need to be tuned in order to optimise the overall classification accuracy. Paradigms like linear programming [Swier, 2001], evolutionary programming [Dolfing, 2004], etcetera can be used for this purpose. When equal weights are used, this is called an averaging ensemble.

The output class with the largest combined class probability will then be chosen.

2.5 Conclusion

Neurons are the basic information processing elements in the brain. Used in large quantities, they form information-processing networks. These neural networks are capable of learning. Computational models have been inspired by these neurobiological circuits. This has led to the development of information theoretic artificial neural networks. Two network models were discussed, the self-organising map and Learning Vector Quantisation. The former is able to develop a topological map from samples. The latter enables optimisations of class borders.

To ease discriminating categories, we desire to obtain as many relevant input dimensions as possible. This is a straightforward way to improve classification scores. However, due to the dense synaptic structure of neural network models, extra input features not always yield better performance. The ensemble techniques provide a better combining strategy, using multiple classifiers on split input information and recombining the results.

Chapter 3

Re-entrant maps

The concept of re-entrant maps originates from Edelman [Edelman, 1993]. It is the most important tenet of his *Theory of Neuronal Group Selection* (TNGS). The first part of this theory describes the carving up of network structure. The different brain areas give rise to a multitude of maps, each possessing a particular function. Local groups of neurons then take on the function of responding to a certain feature (as introduced in chapter 2). As these maps develop, their neuron groups will develop in order to respond more selectively to certain stimuli.

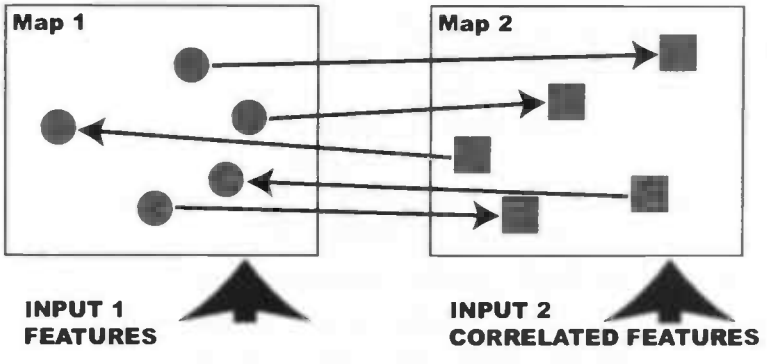

Figure 3.1: When two maps receive correlated features, re-entrant links emerge to connect the correlated neuron groups.

An important characteristic of the information of these maps is temporal synchronicity. Each map processes a part of the information of a scene. Maps receive their input independently from each other, coming either from other maps or from the world. Because of the parallel nature of a neural network, features of the same scene are processed and recognised at the same time. Be-

cause of the overlap and equivalence in functionality, activity in neuron groups from different maps will tend to co-occur. Repeated co-occurring activation will cause a re-entrant link to grow from one group to another (figure 3.1). These links will strengthen the more their activation patterns correspond.

The re-entrant links influence the activity pattern of the connected area. Activation of a neuron group will either excite or inhibit the connected areas. Not all of the reciprocal and cyclic connected neurons have to be activated in order to activate the whole chain. This means that an area can be stimulated, without a stimulus from its original input domain. The same kind of behaviour can be observed in the Pavlov effect. With the concept of re-entrant links, we can thus relate physiology to psychology. The linking of segregated related neuron groups is currently seen as the fundamental mechanism underlying recognition in the brain.

This chapter will be organised as follows. The upcoming section "Re-entrant connections" will address the issue how re-entrant connections are established. How these links can make robust recognition possible is explained in the section "Domain integration". Finally, the section "Self-supervision" describes how the individual maps benefit from robust recognition.

3.1 Re-entrant connections

The group-to-group communication of neuron groups can take on several forms. We can differentiate between different types of communication; figure 3.2 shows the diverse nature of connections. Groups can communicate on a one-on-one, or on a many-to-one basis. This means that a link originates from multiple groups and ends in another. Furthermore, they can operate with or without links stemming from other groups or outside signals.

The theory of re-entrant mapping postulates that all kinds of variants of network structure emerge during the development phase, resulting in a structural diversity. Then, through experience, unused links and neurons are removed. How the connectivity of re-entrant structures emerges from the dependence of activity is of crucial importance. Exactly how this comes about is currently not understood.

It is important to understand the relationship between the Hebb's postulate [Hebb, 1949] and re-entrant mapping.
Hebb's postulate of learning is:

> *When an axon of cell A is near enough to excite a cell B and repeatedly and consistently takes part in firing it, some growth process*

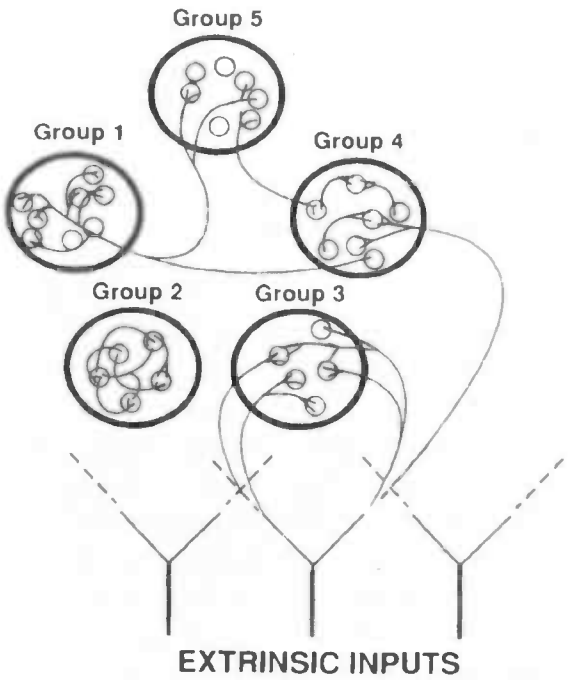

Figure 3.2: The re-entrant links, the channels that interconnect groups, can exhibit different kinds of connectivity. They can link neuron groups on a one-on-one (i.e. groups 4 and 5) or many-to-one (i.e. groups 1,4, and 5) basis. [Edelman, 1993]

> *or metabolic change takes place in one or both cells such that A's efficacy, as one of the cells firing B, is increased.*

According to Edelman, this is exactly the same way (whole) neuron groups connect. This not only implies a larger grain size of the basic connecting units, but also involves enlargement of their scale. Re-entrant links typically connect groups at large distances. An example of their existence is the *corpus collosum*, the main fibre bundle that connects both brain parts with each other. This is a thick axon bundle that consists of many fibres; current estimates indicate approximately two hundred million fibres.

As the different processing areas in the human brain recognise mutually relevant information for each other, they have subjects to recognition in common. This is the reason why typically a bi-directional structure (figure 3.1) will evolve between two maps, connecting both areas.

As the human brain contains far more than two maps, also many-to-one links interconnect the maps. It shows that all areas of the brain are interconnected, with links going back and forth from different maps. This can now be seen as a direct consequence of Hebb's postulate.

3.2 Domain integration

Sometimes, the interpretation of gathered information can be misleading. The recognised features can be contradictory with each other. When we observe that the recognised features are contradictory or confirming, this must also inform us about how the activity of the corresponding processing areas is correlated usually, and therefore how these areas are linked. The re-entrant links provide a *global mapping*. By translating activity in one map to that in another, it provides a means of relating the different information to each other. Edelman has shown that the links literally synchronise observations (the activity in neuron groups). This raises the idea that we can improve performance when we make use of these relations. Sometimes, results of individual maps are insufficient for making the right decision but combined with the interrelationships they are. When re-entrant links alter the perception in a modality, this is called *perceptual change*. This realises robust behaviour in humans. Underlying the mechanism of robustness means that our brains exploit the redundancy of the presented information. Redundant information can then be seen as a form of confirmation or contradiction for a concept.

An example of how contradicting information influences perception is seen in the McGurk effect [McGurk and Mac Donald, 1976]. Here, the experimental subject is shown a video. This video contains the visual footage of a person pronouncing /ga-ga/, but with the audio dubbed to that of the sound /ba-ba/. When this video is shown, the person seems to be saying /da-da/.

This shows how different modalities are integrated to provide robust perception. When the combining of perceptions would be done at a higher level of consciousness, the showing of the video would result in confusion, as both /ba-ba/ and /da-da/ are perceived. This effect, however, implicates that the information processing units of each modality influence each other in their perception. Therefore, each information-processing group searches for a perception that fits other neural groups most. This means that global competition arises through these interconnections. The contradiction of input information is not noticed by the higher order consciousness; it is solved in the lower consciousness. The McGurk effect implies that the combining of information is obligatory.

3.3 Self-supervision

Another function of the re-entrant links is self-supervision. Information is processed in local groups, where each group processes information from its own domain. When a local group wants to respond to a certain feature, it needs information from a global point of view. This information is necessary to improve its classification capability. As depicted in section 2.3, a target signal is needed for topological maps to specialise.

As labelled data usually is scarce, our brains are mostly taught in an unsupervised way. This does not mean no internal targets can be devised. Often the context provides a lot of information. The context should be in accordance with what is perceived. The re-entrant fibres are supposed to play a major part in this. As the re-entrant links make a global mapping, this translates information in one processing area to another. Furthermore, to make sure that our perception is uniform, this information is integrated. However, not only the perception benefits from this robustness, the integrated information also provides a teaching signal. As this signal is less likely to be influenced by local disturbances, this improves the robustness of learning.

The developmental role of the global mapping is called *adaptive categorisation* by Edelman. Adaptive categorisation is a way of tuning neurons to be more selective in their response without previous built-in knowledge of the data. Self-generated supervision should make it possible to learn less distinguishable signals from better distinguishable signals. This enables bootstrapped learning, where it is not necessary to provide (initial) labels.

3.4 Conclusion

Re-entrant mapping discusses how local processing areas connect to influence each other in perception and learning. The topological maps processing areas form, map their input domain without optimising the ability to discriminate amongst certain events. Due to consequent co-occurring activity, neuron groups interconnect. These re-entrant links map information of one domain to another. By doing this, information can be checked on consistency and increases the robustness of perception. The neuron groups synchronise for a unified and consistent perception. Furthermore, discriminative development of neurons is stimulated and categorical boundaries form. This so-called self-supervision is possible under the assumption that the different modalities should experience the same cross-modal object.

Chapter 4

Related work

A lot of research relates to this field somehow. The most resembling projects have been selected for a comparative study. The work done by de Sa [de Sa, 1994] and Yamauchi et al [Yamauchi et al., 1998] shows some innovative ideas on integrating information from modular networks. The Darwin project [Krichmar et al., 2000] takes a more neurobiological and behavioural approach with the construction of an autonomous robot. While, in contrast, Reynaud [Reynaud, 2001] uses a psychological approach to develop a multi-modal system.

Next, we will discuss these four algorithms that all exhibit important properties of re-entrant mapping. Some of these algorithms show an improvement in learning by using multiple modalities for semi-supervised learning. Others, focus more on the improvement in recognition. We will survey these works with respect to how the functionality discussed earlier (chapter 3) is implemented.

4.1 The Darwin project

A series of brain-based devices have been developed to test the theory that supports re-entrant mapping in practise. Cooperation between Krichmar, Snook, Sporns and Edelman [Krichmar et al., 2000] resulted in an autonomous robot partly based on the theory of Edelman. The Darwin robotic system (figure 4.1) has a simulated nervous system. Its neuroanatomy includes an auditory system, a visual system, a taste system, sets of motor neurons capable of triggering behaviour, a visual tracking system, and a value system. These parts of its neuroanatomy process local information; they are allocated in advance. The value system ascribes penalties or rewards, based on basic rules for pleasure or pain.

During the conditioning experiments the local processing parts of the sys-

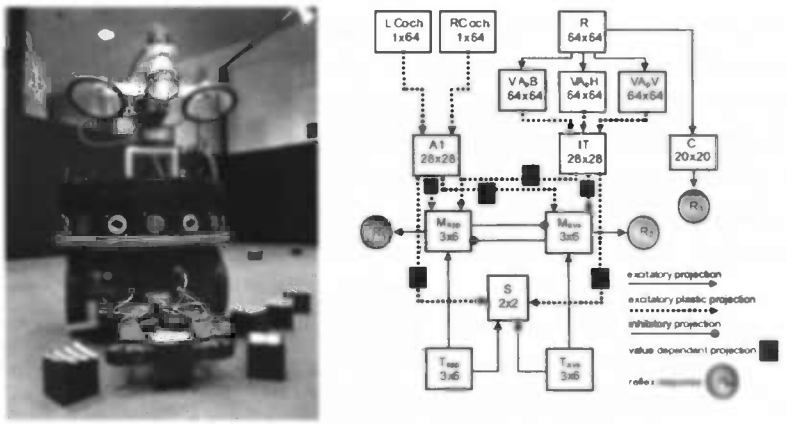

Figure 4.1: Darwin IV: (a) the Darwin autonomous robot in action (b) the different processing areas in its neuroanatomy connect during conditioning experiments.

tems are integrated by synaptic connections (figure 4.1b); consequently the robot's behaviour is conditioned. The robot uses these experience-based associations to avoid negative associated actions, and elaborates on its positive experiences. This means learning the distinction between certain events (i.e. pain-based and pleasure-based events) and responding appropriately to them. Figure 4.1a shows the Darwin robot, and figure 4.1b its neuroanatomy.

This research tries to understand the relationship between brain structure, function, and behaviour. Their conditioning experiments learn the robot to behave itself in a simplified world. This tests the hypothesis whether the TNGS theory (including re-entrant mapping) suffices to create autonomous behaviour. Their research thus focuses more on addressing the foundations of perceptual categorisation and conditioned behaviour, thereby pursuing more neurobiological instead of computational principles.

4.2 Minimising Disagreement

Virginia de Sa shows in her work [de Sa, 1994] that it is possible to use the cross-modal links for learning. Her Minimising Disagreement (MD) network (figure 4.2) enables the development of maps by using class information from other modalities. Two maps (possible topological) are used. Each map initially consists of a representative start population of neurons. During the learning phase, samples are fed pair-wise to the networks by presenting input patterns $X_1(n)$

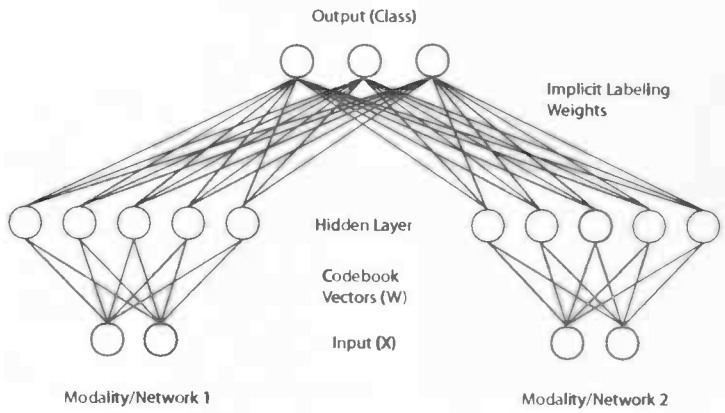

Figure 4.2: The network structure of the Minimising Disagreement algorithm [de Sa, 1994].

and $X_2(n)$ to their respective modalities. Then, the two nearest codebook vectors in modality 1 – w_{1,i_1^*}, w_{1,i_2^*}, and modality 2 – w_{2,k_1^*}, w_{2,k_2^*} to the respective input patterns have to be found. Each network then has a hypothesised output class as given by the label of the closest codebook vector.

After the networks independently classify the input pattern as one of the output classes, the weights of the neurons (the winner and runner-up) from both maps can be adjusted to optimise their decision boundary and minimise their difference in classification. To do this, a 'supervised' target class for each network is set. This is the hypothesised output class of the other network (which received a co-occurring signal).

Each modality updates the weights according to the following rules (Only the rules for modality 1 are given):

If neither or both w_{1,i_1^*} and w_{1,i_2^*} have the same label as w_{2,k_1^*} or $X_1(n)$ does not lie within the window no updates are done, otherwise

$$w_{1,i^*}(n) = w_{1,i^*}(n-1) + \alpha(n) * \frac{(X_1(n) - w_{1,i^*}(n-1))}{\|X_1(n) - w_{1,i^*}(n-1)\|} \quad (4.1)$$

$$w_{1,j^*}(n) = w_{1,j^*}(n-1) - \alpha(n) * \frac{(X_1(n) - w_{1,j^*}(n-1))}{\|X_1(n) - w_{1,j^*}(n-1)\|} \quad (4.2)$$

where w_{1,i^*} is the codebook vector with the same label, and w_{1,j^*} is the codebook vector with another label. The adjustment of weights is comparable with Kohonen's LVQ2.1 algorithm (equation 2.7 and 2.8) as shown in [de Sa and Ballard, 1993].

The effective use of self-supervision means training does not require samples to be labelled, as labels are internally devised. When the label is known, however, the teaching signal generated by self-supervision can be replaced with that of the real label. Learning then corresponds to normal supervised LVQ. The minimising of disagreement is actually gradient descent learning on the mean outcome; this effectively uses the outcome of an averaging voting scheme as a common target signal. Both networks benefit from this; it improves their robustness.

Self-supervision can only work properly when neurons are connected to their true output class. Their connection with the output neurons have to be established in initial labelling stage. A labelling algorithm can pursue two strategies. In case partially labelled data is available, this can be used to label the neurons. If, however, there is an indication of the number of classes that exist, the co-occurrence of activity (induced by unlabelled data) has to be used to connect the neurons to the output neurons.

The Minimising Disagreement algorithm optimises the categorical response of the neurons; when training is finished they will optimally respond to *one* output class. When the two networks differently classify a sample, there is no trivial deduction of the real output class possible. This means that weak classifiers are unsuitable for this algorithm. The self-supervised multi-layer perceptron was designed to solve this issue (see section 4.3).

4.3 Self-supervised multi-layer perceptron

Yamauchi, Oota and Ishii have proposed a way for perceptual grouping with multi-layer perceptrons [Yamauchi et al., 1998]. Figure 4.3 shows the structure of their system. The self-supervised system consists of an ensemble of classifiers, numbered 1 to n. Each classifier $j \in \{1, 2, \ldots, n\}$ calculates the output vector $f[\theta_j^f, x_j(t)]$ using input pattern $x_j(t)$ and the forward structure weights θ_j^f. This output vector, the so-called forward prediction, approximates class probability scores.
The integrating unit I uses a weighted voting scheme on these results:

$$I(t) = \sum_{j=1}^{n} \xi_j(t) f[\theta_j^f, x_j(t)] \tag{4.3}$$

Here $\xi_j(t)$ are the weights used for voting. A control strategy will be used for their values, varying from averaging values to priorities based on backward prediction.

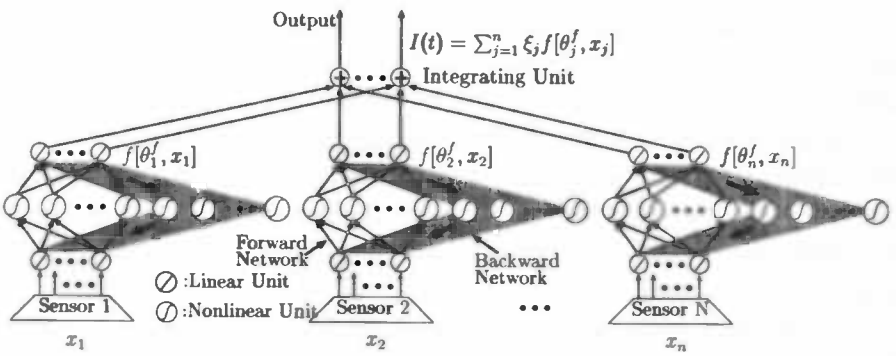

Figure 4.3: The self-supervised multi-layer perceptron uses weighted voting on the forward predictions of the base classifiers. These data compression based classifiers compress samples to their class density probability vectors, forward predictions f. Their accuracy and convergence in learning is improved by reusing averaged class density probability vectors as teaching signal.

Each classifier performs a forward and a backward prediction. Together, the forward and backward structures form a data compression network. In the middle of this network, there is an information bottleneck layer. The size of this neuron layer corresponds to the number of categories. The data compression network targets at reconstructing the sample pattern. Due to data reduction properties, the neurons in the bottleneck will converge to the class probability distribution.

Based on the discussed functionality, we can now compose a number of error functions. The integration unit should provide a better estimate of the sample's class distribution, due to the properties of neural network ensembles discussed earlier. Therefore, the forward predictions $f[\theta_j^f, x_j(t)]$ of the individual networks should equal these. The vector $I(t)$ can therefore be used as a target signal for the forward structure. An estimator that has to be minimised is therefore:

$$E^{(1)}(t) = \sum_{j=1}^{n} \|I(t) - f[\xi_j^f, x_j(t)]\|^2 \tag{4.4}$$

As the backward prediction $b[\theta_j^b, f[\theta_j^f, x_j(t)]]$ should be targeted to the sample pattern $x_j(t)$, this provides a second error measure:

$$E^{(2)}(t) = \sum_{j=1}^{n} \|x_j(t) - b[\theta_j^b, f[\theta_j^f, x_j(t)]]\|^2 \tag{4.5}$$

Further, a third measure of error is the deviation of the sample pattern with

the backward predictions using the result of the integration unit:

$$E^{(3)}(t) = \sum_{j=1}^{n} \|x_j(t) - b[\theta_j^b, I(t)]\|^2 \tag{4.6}$$

The training performs gradient descent learning on the combined error functions.

Although this network does not requires a labelling stage (as the Minimising Disagreement network), the network structure has to be designed manually. For this, knowledge on the number of classes is necessary. Crucial class information will be lost if too little output neurons (of the forward structure) are used. Using too many neurons will complicate the convergence to the class distribution pattern.

The main advantage over this network over the Minimising Disagreement network is the use of multi-layer perceptrons. These function-approximating networks are better at learning graded class probability densities for samples. This is useful for complex decision boundaries. It employs the same ensemble structure that is important for robust recognition. However, more than two networks can be combined in this system. This is at the expense of forming and/or training topological maps.

4.4 A multi-network system for sensory integration

A more psychologically motivated architecture for the integration of modality-specific information is proposed in [Reynaud, 2001]. The architecture of the system using two modalities is illustrated in figure 4.4.

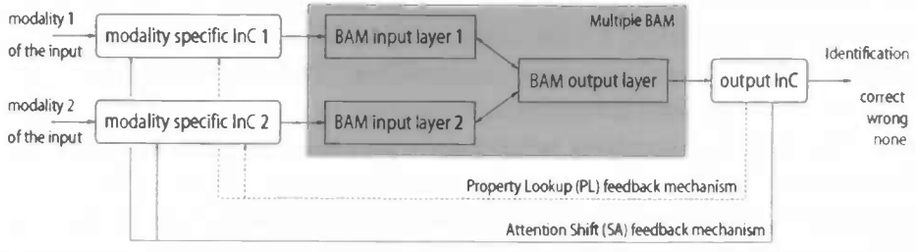

Figure 4.4: A multi-network system for sensory integration. The incremental classifiers InC provide prototypes to the combination unit, the multiple BAM. If the offered prototypes are too different from what it has learnt, one of the classifiers is asked to provide a runner-up for its prototype. [Reynaud, 2001]

The heart of this system is a modified bi-directional associative memory (m-BAM); it is responsible for the actual combining of different modalities. The model is inspired by the attention architecture of Kosslyn and Koenig [Kosslyn and Koenig, 1995]. Information from each modality is processed by an incremental classifier. Here, we use two modality specific classifiers called M_1 and M_2 (although the techniques can be generalised for more modalities). After the classifiers receive their input, they consecutively provide prototypes to the BAM. A prototype should be considered as a suggested solution of the network. The networks thus provide a number of possible solutions, their best matching solution p_{best}, second best solution p_{second}, etcetera.

The m-BAM is a recurrent network, where the different information streams originating from the different modalities are fused. During its learning phase, the BAM is fed explicit combinations of prototypes (from the modalities) that should be associated to an output class. In the testing phase, the modality specific networks incrementally perform classification on their input; providing their current best matching prototypes.

A *property lookup* phase follows, where the m-BAM recurrently updates the combination of prototypes. Each time it improves resemblance with a stored combination of patterns. The number of steps necessary to converge to a stored representation of patterns provides a distance or similarity value. If the convergence time exceeds a certain predetermined threshold value, the system performs an *attention shift*. This means that one of the modality specific networks is asked to provide a runner-up to the currently offered prototype. The new combination of prototypes is then subjected to another lookup phase. In case of a good (quick) match, the pattern is considered recognised.

Matching with different prototypes means iteratively searching for plausible solutions, each time shifting attention to other possible combinations. This approximates minimising a difference function of the offered combination of prototypes with the stored prototype combinations. The network was designed to integrate heterogeneous information, in contrary to some earlier discussed models. This algorithm simulates psychological high-level processing of integrating information. However, it does not resemble re-entrant mapping. It does not target the specialisation that occurs in neurons due to multi-modal information integration. Furthermore, a priori correct combinations of prototypes have to be specified.

	Darwin	MD	SSMLP	m-BAM
connecting groups	√			
topological maps	√	√		√
robust recognition	√	√	√	√
self-supervision	√	√	√	

Figure 4.5: a comparison of the resemblance of the discussed work with re-entrant mapping

4.5 Comparison

So far, we have discussed the algorithms with respect to the properties of re-entrant mapping. Here, we will summarize the differences between the discussed work and re-entrant mapping. The characteristics of re-entrant mapping are (deduced from chapter 3):

- information is processed in local topological maps
- neuron groups interconnect using co-occurrence, forming re-entrant structures
- re-entrant links improve the robustness of recognition
- the developmental impact of re-entrant links on the map

Figure 4.5 shows the compliance of the related works on these subjects. The Darwin project actually incorporates all of the properties. However, due to its neurobiological background, it does not focus on computational advantages. The best way to increase performance with features of re-entrant mapping, is using the Minimising Disagreement network. The self-supervised multi-layer perceptron is a close runner-up. Both algorithms let us incorporate re-entrant mapping in our ensemble techniques.

4.6 Conclusion

We discussed four related works with respect to the characteristics of re-entrant mapping. The comparison showed that the Minimising Disagreement network resembles re-entrant mapping most. The Minimising Disagreement and self-supervised multi-layer perceptron both incorporate the structure of a neural network ensemble. This way, we can benefit from increased performance due to ensemble techniques. Furthermore, their self-supervision techniques let us benefit from the properties of re-entrant mapping.

Chapter 5

An experiment: Hand-written digits

In order to test the usefulness of re-entrant mapping in practice, we developed a proof of concept. One algorithm will be benchmarked; this is the Minimising Disagreement algorithm (section 4.2). The Minimising Disagreement algorithm incorporates most important characteristics of re-entrant mapping. Furthermore, it focuses on achieving computational advantages.

To be of use for Océ, however, we need to test whether this algorithm can also be used in the domain of document analysis. In particular, we will examine if this theory can be applied in the context of an optical character recognition (OCR) problem. The objective in OCR is to extract the text written in an image. The problem of recognising handwritten numerals from a zip code, written on a letter, is such a problem. In the next section, we will discuss the used benchmark set. After this, we will consecutively discuss the used features and benchmark set-up. Finally, we will present the results.

5.1 Benchmark dataset

The well-known MNIST OCR dataset will be used to benchmark the discussed algorithms. This database has a training set of 60.000 handwritten digits and a test set of 10.000 digits. It is a subset of a larger set created by the National Institute of Standards Bureau (NIST). The digits are size-normalised and centred to fixed-size images.

5.2 Features

We will generate two feature streams from the dataset samples by performing the Winkel-Schnitt-Analyse (WSA) and Black-and-White runs. This will create two really complementary feature sets. This way, the networks are able to classify on their own, but with the help of each other could perform better. Their preprocessing steps will be described in the next subsections.

5.2.1 Winkel-Schnitt-Analyse

The WSA analysis takes slices of an image in a certain angle. Per slice, the number of black pixels is counted. The information from all slices is accumulated in a frequency histogram. Figure 5.1 illustrates the WSA.

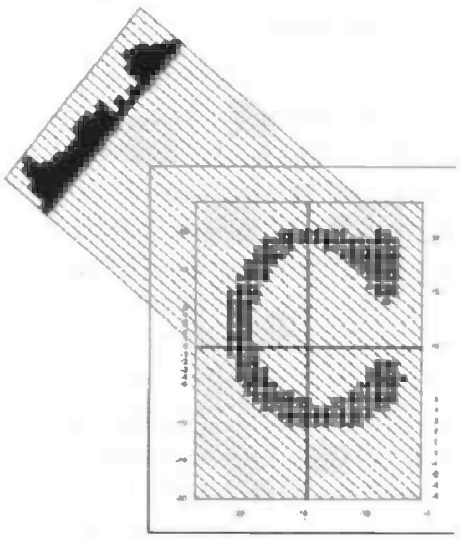

Figure 5.1: The Winkel-Schnitt-Analyse captures direction dependent density information in a frequency histogram. [Dolfing, 2004]

A WSA histogram provides shift-invariant information. When a character is moved along the projection axis, the frequency histogram will stay the same. A digit usually cannot be recognised using one projection. To capture the two-dimensional information from the image, a second projection is needed.

We will extract x- and y-projections (i.e. 0° and 90° slices) of the characters, as suggested by [Ho, 1992].

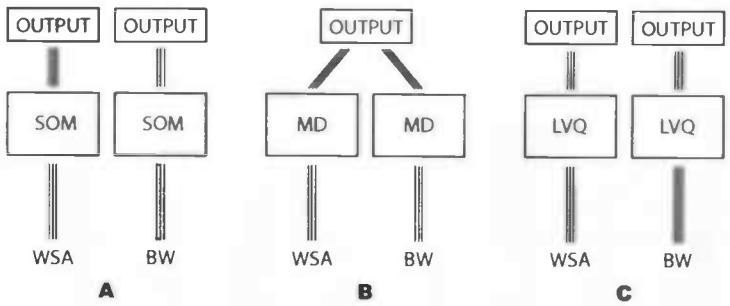

Figure 5.2: The set-up of the three experiments: (a) two self-organising maps (b) the Minimising Disagreement network (c) two Learning Vector Quantisation networks

5.2.2 Black-And-White runs

The Black-And-White runs, similar to the WSA, are also produced by cutting an image in slices. However, the presence of black pixels isn't counted, but rather the colour transitions. This is the amount of colour transitions from black to white, or vice versa. This is an indication for the number of intersecting pencil strokes. Note that, this is precisely the information we lose by taking a WSA. Again, we will use the projections along the x- and y-axis.

5.3 Set-up

Whether the Minimizing Disagreement algorithm can increase the classification accuracy of the classifiers, will be investigated with three experiments. Their set-up is depicted in figure 5.2. We will vary the network models used; these are self-organising maps, Learning Vector Quantisation, and of course Minimising Disagreement. Each network will use 600 neurons (the self-organising map a 30x20 two-dimensional lattice).

The first experiment processes the features with self-organising maps. The maps were labelled and benchmarked on performance. The second experiment initialises the Minimising Disagreement network with these codebook vectors and labels. It was then trained with merely unlabelled data from the training set. The last experiment trained the networks with the supervised LVQ2.1 algorithm (section 2.3).

The benchmark scores of the LVQ and SOM algorithm were calculated using the SOM_PAK and LVQ_PAK implementations, developed by the Helsinki University of Technology.

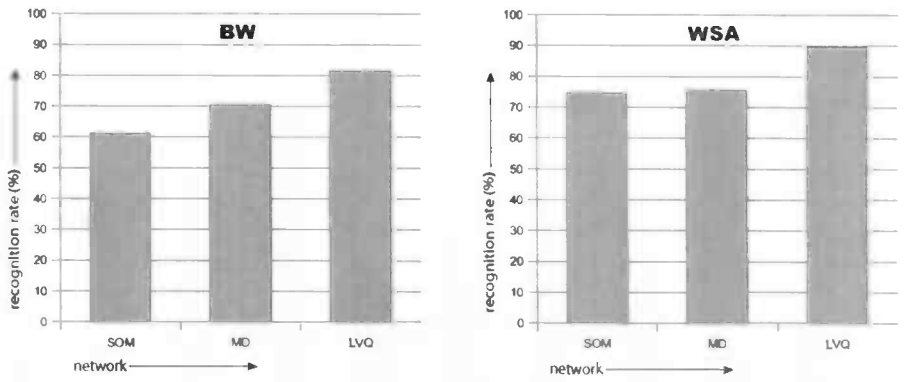

Figure 5.3: The recognition rates of respectively the SOM, LVQ, and MD networks on separate feature streams.

5.4 Results

Although classification scores of the Minimising Disagreement network, inherent to the used algorithm, never exceed the LVQ score (section 4.2), there is no reason to assume that the lower boundary is strict. However, the results show an overall improvement compared to the self-organising maps. This means unlabelled data is useful in the development of topological maps.

It shows that both classifiers benefit from participating in the Minimising Disagreement network. The weakest classifier, which uses the Black-and-White runs, improved most. This shows how knowledge can be transferred from one network to another. Furthermore, it does not use labelled data for training. This is very important in some situations.

An overview of the results is shown in figure 5.3. A more extensive overview of these class-dependent results is provided by figure 5.4 and 5.5 .

5.5 Conclusion

The experiments clearly showed the benefits of the minimizing disagreement. By cooperation of two networks, a strict improvement for both networks was attained. The performance is better than the self-organising map, but worse than Learning Vector Quantisation. Thus, we can increase classification accuracy with unlabelled data.

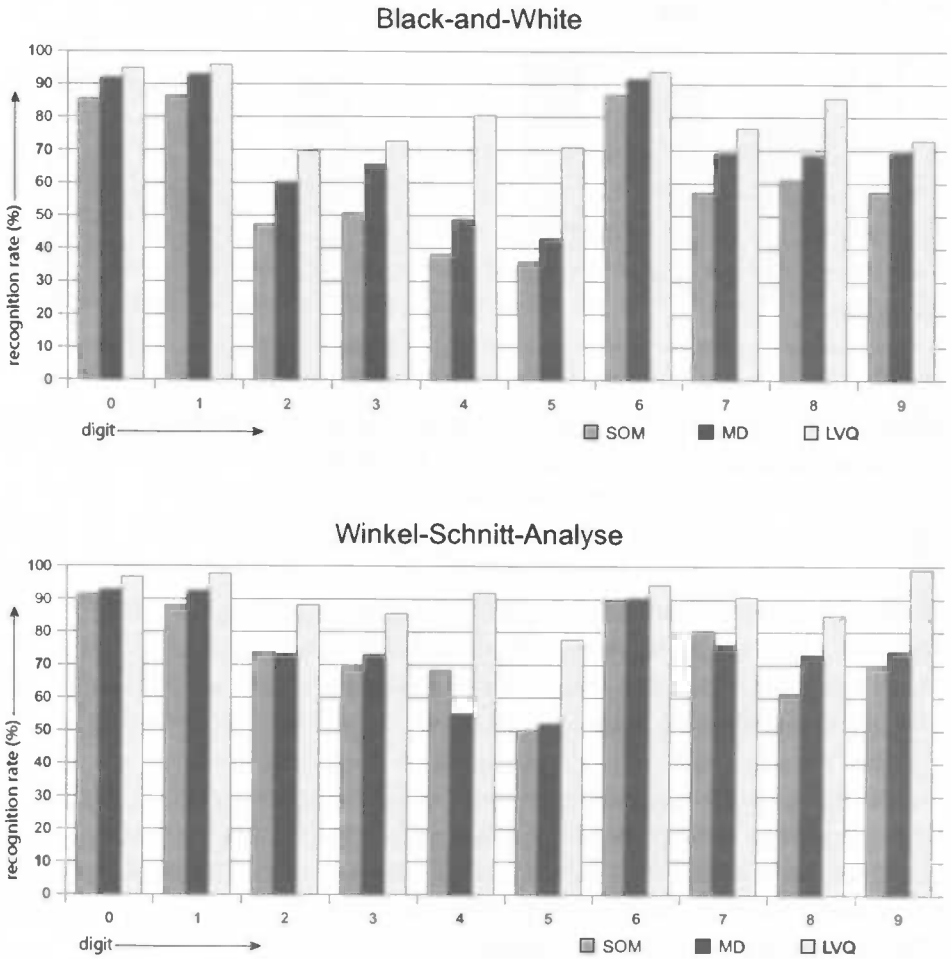

Figure 5.4: A bar chart diagram of the results per category.

Black-and-White	0	1	2	3	4	5	6	7	8	9	Average
SOM	85.5%	86.5%	47.3%	50.7%	38.3%	36.0%	86.9%	57.1%	61.0%	57.3%	61.1%
MD	91.8%	92.9%	60.1%	65.5%	48.8%	42.9%	91.6%	69.2%	68.6%	69.3%	70.5%
LVQ	94.8%	95.9%	69.8%	72.7%	80.5%	70.7%	93.7%	76.7%	85.8%	72.8%	81.5%

Winkel Schnitt Analyse	0	1	2	3	4	5	6	7	8	9	Average
SOM	91.5%	88.0%	73.8%	69.4%	68.3%	50.3%	89.8%	80.4%	61.4%	69.9%	74.7%
MD	92.8%	92.5%	73.3%	73.1%	55.1%	52.1%	90.4%	76.2%	73.1%	74.0%	75.7%
LVQ	96.6%	97.8%	88.3%	85.5%	92.0%	77.7%	94.4%	90.8%	85.1%	99.1%	89.8%

Figure 5.5: Category dependent results of the different algorithms in numbers.

Chapter 6

Conclusions and future directions

We have shown that a neurobiological theory, re-entrant mapping, can be incorporated in neural network ensembles. The related implementations of this theory conform to the techniques of neural network ensembles. Troubles with these techniques were discovered in earlier studies [Swier, 2001]. Because of the strict resemblance, none of the current problems concerning ensembles can be solved with the discussed re-entrant mapping algorithms. Nevertheless, these algorithms increase their robustness by applying ensemble techniques.

The theory on re-entrant mapping, however, bares other applications. Ensemble techniques are usually employed for combining trained classifiers. The re-entrant mapping techniques, on contrary, also use the ensemble techniques in the *development* of neural networks. Labelled samples train the networks in a traditional manner. However, this research has shown that unlabelled samples can also contribute to the development of the networks. This is done by incorporating the ensemble techniques in our teaching methods.

The ability of networks to devise supervision intern, also called self-supervision, means the end of separate learning and testing phases. As networks never finish learning, their application domain shifts from off-line to on-line learning. The time required for off-line learning can therefore be replaced with training on the job. This is a major advantage in situations where labelled data is scarce.

The considered algorithms provide no advantage over the classical supervised ensemble techniques when enough labelled data is available. In that case, these algorithms are exactly the same and thus can not beat supervise based performance. Although training with unlabelled data does not provide performance equal to supervised learning, it does improve classification accuracy. As

unlabelled data usually is available in abundance, these improvements are free.

6.1 Future directions

Re-entrant mapping plays a major role in autonomous behaviour. In order to elaborate on this subject, it is important to detect cross-modal objects automatically. The current implementation still requires human intervention, by either connecting category-related neurons or specifying the number of categories. Categories are very important structures for learning; discriminative behaviour can only be developed with the notion of a target.

Future experiments can elaborate on the notion of a global mapping to enable robust recognition. When detectors fail or provide confusing information, information from other maps can be used to fill in the blanks. This is more like the experiment on controlling the user interface by eye movement [Salojärvi et al., 2003]. Here, eye movement cannot directly be used for controlling the user interface. However, during conditioning experiments it might show that certain eye movements are correlated with controlling certain user interface elements. This obtained information can then be used to improve robustness. Note that this is more like the McGurk effect; because there is a relation between lip movement and speech, we learn to use lip movement in perceiving speech.

While it is important to be able to learn at all time, the discussed algorithms use a decreasing learning rate. This means networks will not be able to adjust to radical boundary shifts in time. Other algorithms might provide better online learning properties. The Adaptive Resonance Theory (ART) is designed for this property; there are some recent implementations on a modified version that enables self-supervision [Butz and Ray, 2003]. Further, other semi-supervised networks might be more suited for demanding categorisation tasks. For example, the semi-supervised support vector machine [Bennett and Demirz, 1998] has shown some promising results in this area. The principles of self-supervision as depicted in this thesis, can easily be extended for other networks.

6.2 Acknowledgements

As my supervisors, I would like to thank Jan Jacobs and Sebastian de Smet of Océ-Technologies and Jos Nijhuis of the university of Groningen for their support during my master's degree project.

As a last remark, I would like to thank everyone who reviewed my thesis for

the feedback.

Bibliography

[Edelman, 1993] Gerald M. Edelman, 1993, Bright Air, Brilliant Fire (book)

[de Sa, 1994] Virginia Ruth de Sa, 1994, Unsupervised Classification Learning from Cross-modal Environmental Structure

[Reynaud, 2001] Emanuelle Reynaud, 2001, A Multisensory Identification System for Robotics

[Crépet et al., 2000] Agnès Crépet, Hélène Paugam-Moisy, Emanuelle Reynaud & Didier Puzenat, 2000, A Modular Neural Modal for Binding Several Modalities

[Yamauchi et al., 1998] K. Yamauchi, M. Oota, N. Ishii, 1998, A self-supervised learning system for pattern recognition by sensory integration

[Westermann, 2001] Gert Westermann, 2001, A Model of Perceptual Change by Domain Integration

[Butz and Ray, 2003] Martin V. Butz, Sylvian Ray, 2003, Bidirectional ARTMAP: An Artificial Mirror Neuron System

[Swier, 2001] Miranda Swier, Marvin Brünner, Ruud Janssen, 2001, Multiple Classifier Systems

[Jain et al., 1999] Anil K. Jain, Robert P.W. Duin, Jianchang Mao, 1999, Statistical Pattern Recognition: A Review

[Körding and Kördig, 2000] Konrad Körding, Peter Kördig, 2000, Two sites of synaptic integration: relevant for learning?

[Kohonen, 1997] T. Kohonen, 1997, Second Edition, Self-organising Maps

[McGurk and Mac Donald, 1976] H. Mc Gurk, J. Mac Donald, 1976, Hearing lips and seeing voices

[Krichmar et al., 2000] Jeffrey L. Krichmar, James A. Snook, Gerald M. Edelman, Olaf Sporns, 2000, Experience-Dependent Perceptual Categorization in a Behaving Real-World Device

[Ho, 1992] Tin Kam Ho, 1992, A Theory of Multiple Classifier Systems and its application to visual word recognition

[Yu, 2003] Stella X. Yu, 2003, Computational Models of Perceptual Organization

[Sajda, 2001] Paul Sajda, 2001, Neural Networks – Encyclopedia of the Human Brain, Volume 1

[Singer, 1999] Wolf Singer, 1999, Neuronal Synchrony: A Versatile Code Review for the Definition of Relations?

[Haykin, 1998] Simon Haykin, 1998, Neural Networks, Second Edition

[Hebb, 1949] D.O. Hebb, 1949, The organization of behavior: A neuropsychological theory

[Salojärvi et al., 2003] Jarkko Salojärvi, Ilpo Kojo, Jaana Simola, and Samuel Kaski, 2003, Can relevance be inferred from eye movements in information retrieval?

[Weiss, 1997] Gerhard Weiss, 1997, Towards the synthesis of neural and evolutionary learning

[Kandel et al., 2000] Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, 2000, Fourth Edition, Principles of Neural Science

[de Sa and Ballard, 1993] V.R. de Sa, D.H. Ballard, 1993, A Note On Learning Vector Quantization

[Zrehen and Gaussier, 1994] Stephane Zrehen, Philippe Gaussier, 1994, Why Topological Maps Are Useful for Learning in an Autonomous Agent

[Dolfing, 2004] Henrico Dolfing, 2004, Determination of Expert Weights for the Adaptive Combination of OCR Result Strings

[Christensen, 2003] Wayne Christensen, 2003, Self-directedness, integration and higher cognition

[Bennett and Demirz, 1998] Kristin P. Bennett, Aykan Demirz, 1998, Semi-Supervised Support Vector Machines

[Kosslyn and Koenig, 1995] S. M. Kosslyn and O. Koenig, 1995, Wet mind: the new cognitive neuroscience