UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RAFAEL COIMBRA PINTO

A Neocortex Inspired Hierarchical Spatio-Temporal Pattern Recognition System

Undergraduate Thesis presented in partial
fulfillment of the requirements for the degree of
Bachelor of Computer Science

Prof. Dr. Paulo Martins Engel
Advisor

Porto Alegre, June 2009

“No great discovery was ever made without a bold guess.”
— Sir Isaac Newton

“I believe in intuition and inspiration. Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution. It is, strictly speaking, a real factor in scientific research.”
— Albert Einstein

“We rarely recognize how wonderful it is that a person can traverse an entire lifetime without making a single really serious mistake – like putting a fork in one’s eye or using a window instead of a door.”
— Marvin Minsky

THANKS

I thank my grandma Margarida, be she wherever she is, without who I would never have come this far. I also thank my parents for always believing in me.

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
SOM	Self-Organizing Map
RSOM	Recurrent Self-Organizing Map
PLSOM	Parameterless Self-Organizing Map
AntSOM	Anticipatory Self-Organizing Map
RecAntSOM	Recurrent Anticipatory Self-Organizing Map
HQSOM	Hierarchical Quilted Self-Organizing Map
SRN	Simple Recurrent Network
MLP	Multi-Layer Perceptron
HTM	Hierachical Temporal Memory
MPF	Memory Prediction Framework
SVM	Support Vector Machine
RPROP	Resilient Backpropagation

LIST OF FIGURES

ABSTRACT

Recently, inspired by our neocortex, an architecture for spatio-temporal pattern recognition has been proposed, the MPF (Memory-Prediction Framework). There already exists some attempts to implement it, like the HTM (Hierarchical Temporal Memory) and the HQSOM (Hierarchical Quilted Self-Organizing Map). The HQSOM lacks some parts of the MPF, which in turn makes it less powerful than it could be. This work aims to provide a step towards a complete implementation of the MPF by using some variants of a self-organizing map (SOM). In order to reach this goal, a new variant of the SOM is proposed: The LoopSOM. The LoopSOM extends the SOM-RSOM pair of the HQSOM with predictive and fallback abilities. The LoopSOM has been integrated to the HQSOM, making it closer to the MPF. Experiments were done in order to verify the results obtained with this modification.

Keywords: Pattern recognition, self-organizing maps, RSOM, HQSOM, hierarchical temporal memory, memory prediction framework, LOOPSOM, feedback, hierarchical, neural networks.

Um Reconhecedor de Padõres Espaço-Temporais Inspirado no Neocórtex

RESUMO

Recentemente, inspirado pelo nosso neocórtex, uma arquitetura para reconhecimento de padres espaço-temporais foi proposta, o MPF (Memory-Prediction Framework). Já existem algumas implementações dele, como a HTM (Hierarchical Temporal Memory) e o HQSOM (Hierarchical Quilted Self-Organizing Map). O HQSOM não possui algumas partes do MPF, o que o torna menos poderoso do que poderia ser. Este trabalho visa fornecer um passo em direção a uma implementação completa do MPF usando algumas variações do Mapa Auto-Organizável (SOM). Para alcançar este objetivo, uma nova variante do SOM é proposta: O LoopSOM. O LoopSOM estende o par SOM-RSOM do HQSOM com habilidades preditivas e de recuo. O LoopSOM foi integrado ao HQSOM, tornando-o mais próximo do MPF. Foram feitos experimentos a fim de verificar os resultados obtidos com esta modificação.

Palavras-chave: reconhecimento de padrões, mapas auto-organizáveis, memory prediction framework, neocórtex, predição, realimentação.

1 INTRODUCTION

Previously, Hawkins (2005) proposed a framework for computational modeling of the neocortex, the Memory-Prediction Framework (MPF). Basically it says the basic unit of the model must do spatial and temporal processing, such that the temporal portion sends predictions to the spatial portion. The output of the spatial part serves as input to the temporal part, which recognizes/classifies sequences. The temporal part sends its outputs to the next layer, and also sends predictions back to the spatial part. The cortex is built up from layers with many of these processing units.

1.1 Objectives

This work proposes to introduce feedback connections into the SOM-RSOM pair, allowing the RSOM to send predictions to the SOM, providing better pattern classification/recognition and solving ambiguities. Also, with this improvement, it perfectly fits into the MPF. This new algorithm was introduced by Pinto and Engel (2009) and is called LoopSOM. After that, the LoopSOM will be introduced into the HQSOM, providing a better implementation of the MPF.

1.2 Organization

The rest of this work is organized as follows. The next chapter explains the Memory-Prediction Framework, the HTM (Hierarchical Temporal Memory) algorithm (a Bayesian implementation of the MPF), the basic SOM (Self-Organizing Map) algorithm, the RSOM (Recurrent SOM) algorithm and HQSOM (Hierarchical Quilted Self-Organizing Map) algorithm (a neural implementation of the MPF). After that, the new LoopSOM algorithm is presented in Chapter 3, and some experiments are made. Having this new algorithm, Chapter 4 improves the HQSOM by replacing its SOM-RSOM pairs with LoopSOM's, and some more experiments are shown. Chapter 5 finishes this work with conclusions and future works.

2 RELATED WORKS

2.1 The Memory-Prediction Framework

Hawkins proposed a framework for computational modeling of the neocortex, the Memory-Prediction Framework (MPF). Basically it says the basic unit of the model must do spatial and temporal processing, such that the temporal portion sends predictions to the spatial portion. The output of the spatial part serves as input to the temporal part, which recognizes/classifies sequences. The temporal part sends its outputs to the next layer, and also sends predictions back to the spatial part. The cortex is built up from layers with many of these processing units.

As higher layers are examined, the concepts become more and more abstract (holding concepts like "person" or "ball"), and more concrete concepts are on the bottom, such as pixels of an image. Another important part of the MPF is that every sensory modality must use that same mechanism. The sensory inputs are what makes each modality unique.

It seems that simple neuron cells in our visual cortex perform spatial recognition (responding to edge patterns with a specific orientation and location in their receptive field) and complex neuron cells perform temporal recognition (responding to edge patterns with a specific orientation anywhere in their receptive field, that is, it's shift invariant). Since most movement we perceive is shifting, it is proposed as an explanation for how shifted edges might be associated together - because they occur near in time.

There are a few implementations of the MPF by now, like the Hierarchical Temporal Memory (HTM), proposed by Hawkins (2006), and the Hierarchical Quilted Self-Organizing Map (HQSOM) proposed by Miller (2006). The former is a Bayesian network implementation, while the later is a neural network implementation, but lacks prediction connections, as required by the MPF.

2.2 The Hierarchical Temporal Memory

HTMs are composed by a hierarchy of nodes performing the same learning algorithm. Figure 2.1 shows a simple HTM hierarchy. Sensory data enters at the bottom while the outputs exiting the top are a vector where each element represents a potential feature of the sensory data. Each node in the hierarchy performs the same function as the overall hierarchy. That is, each node recognizes spatio-temporal patterns of its inputs and learns to assign features to this input pattern. In other words, each node, no matter where it is in the hierarchy, discovers the features of its input. The outputs of nodes at each level become the inputs to the next level. Nodes at the bottom receive input from a small area of the sensory input, so the features they discover are relevant to a small part of the sensory input area. Higher layers receive input from multiple nodes below, and again discover

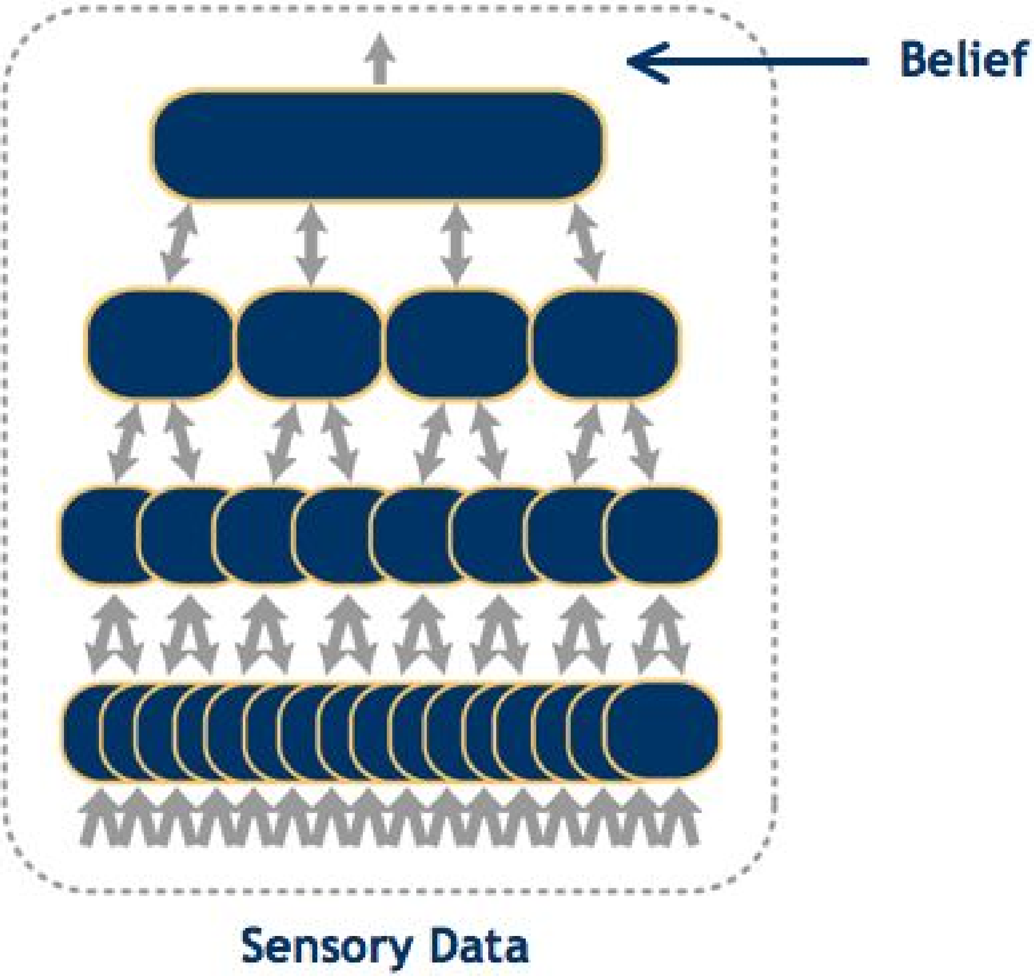

Figure 2.1: The HTM layered structure.

the features in this input. These features will be of intermediate complexity, occurring over larger areas of the entire input space. The node or nodes at the top of the hierarchy represent high level features that may appear anywhere in the entire sensory field. For example, in a visual inference HTM, nodes at the bottom of the hierarchy may discover low level features such as pixels, edges, lines, and corners in a small part of the visual space. Nodes at the top of the hierarchy will represent higher level complex features such as objects and faces, which can appear over the entire visual space or any sub-part of the visual space. Such structure can be seen as an hierarchical Bayesian network. Each layer sends predictions to lower layers, as well as temporal predictions inside single nodes are sent to their spatial processors. As can be seen, the HTM is a full implementation of the MPF. However, it's important to note that it currently doesn't support online learning, in contrast to batch learning.

2.2.1 Current Applications

The HTM was successfully applied to some applications, which will be shortly described here.

2.2.1.1 NuPIC

One such application is NuPic (Numenta Platform for Intelligent Computing), described in Numenta (2008), which performs invariant general image classifications. The algorithm had 66% accuracy (where the chance level is only 2%) in classifying images on the test set, and 99.73% in the training set, as shown by George (2007). There was a total of 9000 images in 48 categories.

2.2.1.2 Digit Recognition

Thornton (2006) has used the HTM for character recognition in the CEDAR database using the Buffalo Designed (BD) handwritten digit base. It had 96.32% accuracy against 92.28% of the Riedmiller's (1993) RPROP (Resilient Backpropagation) neural network algorithm.

2.2.1.3 Captcha Recognition

Hall (2007) has shown how an HTM performed a hard character recognition task on CAPTCHA images. There was an average of 89% accuracy when performed by human subjects. The HTM obtained 20% accuracy against 9% of a SVM (Support Vector Machine, Cristianini (2000)) neural network.

2.3 Self-Organizing Maps

A Self-Organizing Map is an unsupervised neural network intended for spatial clustering, vector quantization, dimensionality reduction and topology preservation. Topology preservation means that patterns close in input space produce patterns close in output (map) space. The neuron map forms a code-book of patterns from input space, composed by synapse weights. The learning takes place in a winner-take-all approach, by selecting the best matching unit (BMU) with the following equation:

$$\|\mathbf{x}(t) - \mathbf{w}_b\| = \min_{i \in V_O}\{\|\mathbf{x}(t) - \mathbf{w}_i\|\} \tag{2.1}$$

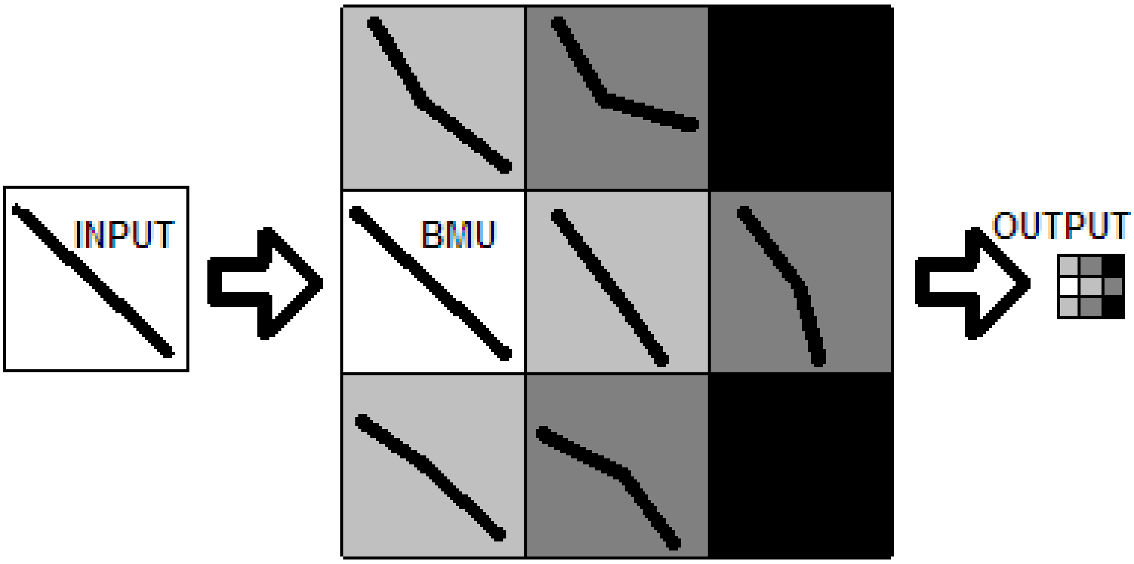

Figure 2.2: The input pattern is compared to each SOM unit, selecting the BMU. An activation vector may be produced from the distances vector from the input.

where V_O is the output space, $\mathbf{x}$ is the input vector and $\mathbf{w}_i$ is the weight vector of the ith unit. After finding the BMU, its weights are updated, as well as the weights of its neighbors, according to the following update rule:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \gamma h_{ib}(t)(\mathbf{x}(t) - \mathbf{w}_i(t)) \tag{2.2}$$

where h_{ib} is a neighborhood function such as:

$$h_{ib}(t) = \exp\left(\frac{-\|\mathbf{I}_i - \mathbf{I}_b\|^2}{2\sigma(t)^2}\right) \tag{2.3}$$

where $\mathbf{I}_i$ and $\mathbf{I}_b$ are indices of the map units i and b, and $\sigma(t)$ is the gaussian standard deviation. Note that σ is dependent on time and normally is implemented into a cooling schedule. The problem with that approach is that it's not possible to do online learning. To fix it, Miller proposed to change the previous equation by the following one

$$h_{ib}(t) = \exp\left(\frac{-\|\mathbf{I}_i - \mathbf{I}_b\|^2}{\mu_b(t)\sigma^2}\right) \tag{2.4}$$

where $\mu_b(t)$ is the mean squared error of $\mathbf{w}_b$ compared to the input $\mathbf{x}$(t), and is given as follows:

$$\mu_b(t) = \frac{1}{N}\|\mathbf{x}(t) - \mathbf{w}_b\|^2 \tag{2.5}$$

where N is the length of the input vector. This enables the SOM for online learning by adjusting dynamically the neighborhood size. Other possible approaches for online learning are the PLSOM (Parameter-less SOM, as from Berglund (2003)) and its improved version by Berglund (2009), although Miller's simpler approach will be used for now. Additionally, an activation vector may be produced from the distances vector from the input x(t) with some function like a Gaussian:

$$y_i(t) = \exp\left(\frac{-\|\mathbf{x}(t) - \mathbf{w}_i\|^2}{2\rho^2}\right) \tag{2.6}$$

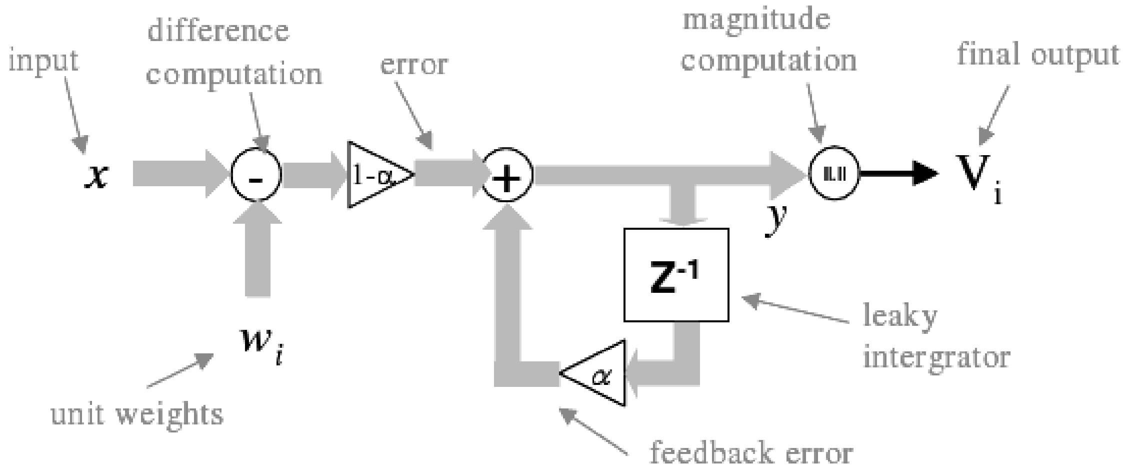

Figure 2.3: The RSOM structure.

where ρ is the standard deviation of the gaussian. Smaller ρ means more local coding while bigger ρ means denser coding (Foldiak, 1995). This vector will be useful later as input to another map for creating a SOM-RSOM pair. See figure 2.2 for an example.

2.4 The Recurrent SOM

The Recurrent SOM performs temporal clustering by using decayed traces of previous vector differences. At each time step a recursive difference vector $\mathbf{d}_i(t)$ is calculated as:

$$\mathbf{d}_i(t) = (1-\alpha)\mathbf{d}_i(t-1) + \alpha(\mathbf{x}(t) - \mathbf{w}_i(t)) \tag{2.7}$$

where $\alpha, 0 \leq \alpha \leq 1$ is the decay factor, **x** is the input vector and $\mathbf{w}_i$ is the weight vector of the ith unit. The memory becomes deeper as α gets closer to 0, being the original SOM a special case of the RSOM where α = 1 (no memory). Now the BMU can be found by using the following equation:

$$\mathbf{d}_b(t) = \min_{i \in V_O}\{\|\mathbf{d}_i(t)\|\} \tag{2.8}$$

and the new update rule is as follows:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \gamma h_{ib}(t)\mathbf{d}_i(t) \tag{2.9}$$

The result is a set of invariant representations for patterns which are correlated temporally.

2.5 The SOM-RSOM Pair

A problem with the RSOM is that if the input vectors are not orthogonal, some ambiguity will be created. To overcome this problem, Miller proposed to use the SOM output **y** (equation 6) as the input **x** for the RSOM (equation 7), creating a SOM-RSOM pair. So, the smaller the used ρ, more orthogonal will be the output and the RSOM will work better.

2.6 The Hierarchical Quilted Self-Organizing Map

The Hierarchical Quilted Self-Organizing Map (HQSOM) is a neural implementation of the MPF, using the RSOM organized into an hierarchy. The spatial SOM (an RSOM

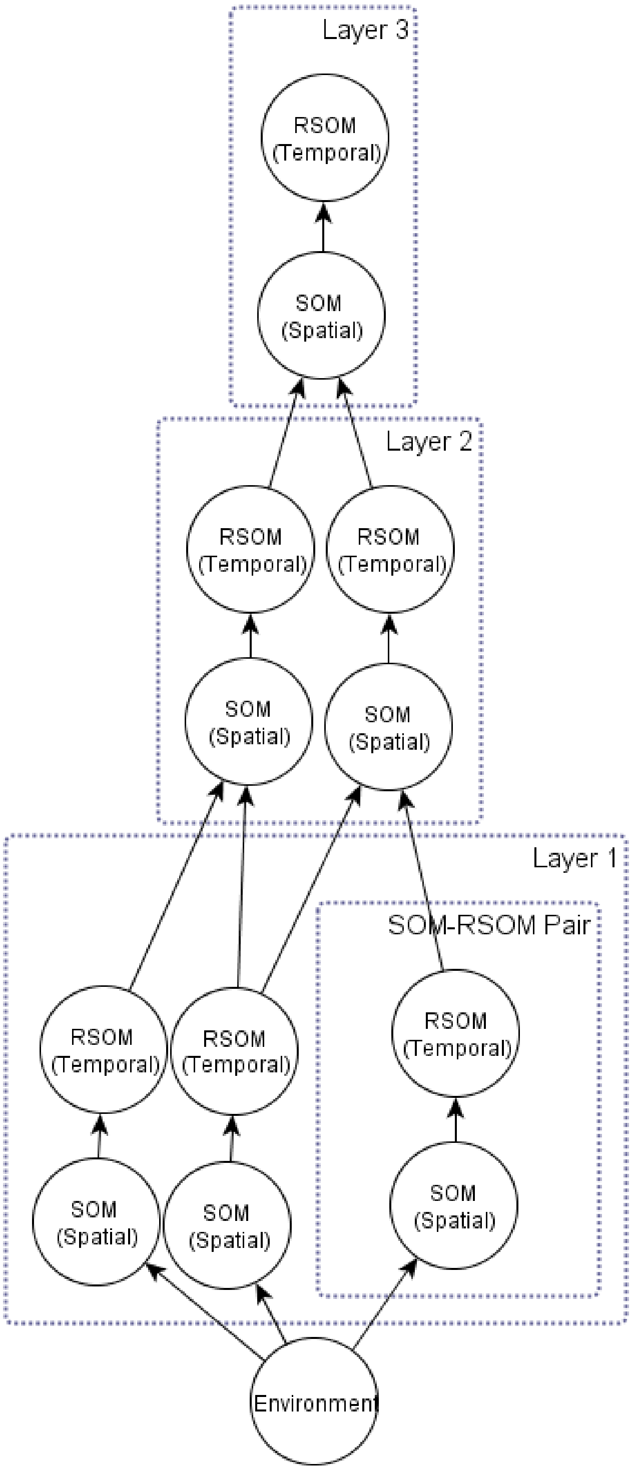

Figure 2.4: The HQSOM structure.

with $\alpha = 1$) mimics groups of simple cells, while the temporal RSOM mimics corresponding groups of complex cells. In some sense, it is similar to Fukushima's Neocognitron (1980), Riesenhuber's HMAX (1999), and Behnke's NAP (1998), consisting of layers of simple-complex cells (SOM-RSOM pairs). The input is parsed into overlapping receptive fields, each of which is sent to a SOM-RSOM pair in the first layer. Each layer is a quilt of SOM-RSOM pairs with overlapping receptive fields (overlapping receptive fields are not mandatory for it to work, but it may increase the fault tolerance of the system). The output of the RSOM in each SOM-RSOM pair is combined with the outputs of the other pairs in the same layer to compose a feature vector which is presented to the next higher layer. In this work (as well as Miller's), the indices of the BMUs from the RSOMs are used as outputs. The signals passed from layer to layer represent topological maps of features, with increasing size and complexity after each layer. When the hierarchy gets to a single SOM-RSOM pair at the top, the output of the entire hierarchy is a single integer pair (BMU position) a compressed and highly abstracted invariant representation of the image presented to the lowest layer. Other algorithms, such as reinforcement learning or supervised learning, might use the outputs of other layers as well for its inputs. At the top of the hierarchy, sensory patterns are compressed into representations which are invariant to any transformation (e.g. shift, scale, rotation) as long as related patterns occur near in time. The HQSOM implements only feed-forward connections. However, there is a massive amount of feedback connections in our cortex, and their utility would be to predict future activity of lower layers and bias their activity, resolving ambiguities and increasing robustness. In the next chapter, a replacement for the SOM-RSOM pair will be proposed, so there will be feedback connections doing exactly that.

3 THE LOOPSOM

The SOM-RSOM pair works very well for spatio-temporal pattern processing, but it doesn't use all of its potential and doesn't conform totally to the MPF, because of the lack of feedback connections carrying predictions from the RSOM to the SOM. Such connections may solve ambiguities on the spatial level and provide higher noise tolerance, resulting in a more robust unit. A new implementation, called LoopSOM, aims to provide a way to implement such connections. In the LoopSOM, the weight vector $\mathbf{w}_b$ of the RSOM BMU is used as an activation prediction $\mathbf{p}(t)$ for the SOM, as shown in figure 3.2. So, the prediction is as follows:

$$\mathbf{p}(t) = \mathbf{w}_{bt}(t-1) \tag{3.1}$$

The current SOM activation vector **y**(t) will be combined with the last RSOM prediction **p**(t-1), and this new equation will be used to compute the highest activation and find the BMU:

$$y_b(t) = \max_{i \in V_O} \left(\frac{\xi_s(t) y_i(t) + \xi_t(t-1) p_i(t)}{\xi_s(t) + \xi_t(t-1)} \right) \tag{3.2}$$

where $\xi_s(t)$ is the spatial SOM output confidence computed as follows:

$$\xi_s(t) = 1 - \frac{1}{2} \left\| \frac{\mathbf{x}_s(t)}{||\mathbf{x}_s(t)||} - \frac{\mathbf{w}_{bs}(t)}{||\mathbf{w}_{bs}(t)||} \right\| \tag{3.3}$$

where $\mathbf{x}_s(t)$ is the input vector and $\mathbf{w}_{bs}(t)$ is the weight vector of the BMU. Note that the input must be processed with the original BMU equation (1) in order to get the confidence value, and the SOM learning rate must be kept at 0 while doing it, to avoid interfering with the new BMU calculation. In the same way $\xi_t(t-1)$ can be computed as:

$$\xi_t(t-1) = 1 - \frac{1}{2} \left\| \frac{\mathbf{x}_t(t-1)}{||\mathbf{x}_t(t-1)||} - \frac{\mathbf{w}_{bt}(t-1)}{||\mathbf{w}_{bt}(t-1)||} \right\| \tag{3.4}$$

being the variables analogue to the ones on the previous equation. Here the values from the previous pattern presentation must be used, since they are used to predict the current activations. Analyzing these 2 factors, some extreme cases can be shown: when both confidences are similar, current observations and predictions will weight near the same. When $\xi_s > 0$ and ξ_t equals 0, the original SOM-RSOM pair without feedback is obtained. When ξ_s equals 0 and $\xi_t > 0$, pure prediction is obtained and could even generate the most likely sequences decoupled from the environment. This can be seen as a form of fall back behavior, as proposed by Cohen (1998), so that the classification can fall back to pure spatial classification when temporal classification is sub-optimal.

The two confidence values may be used in other ways too. For instance, the learning rate γ of the RSOM can be a function $\gamma(\xi_s(t))$ (it will learn less if there is error on the spatial SOM).

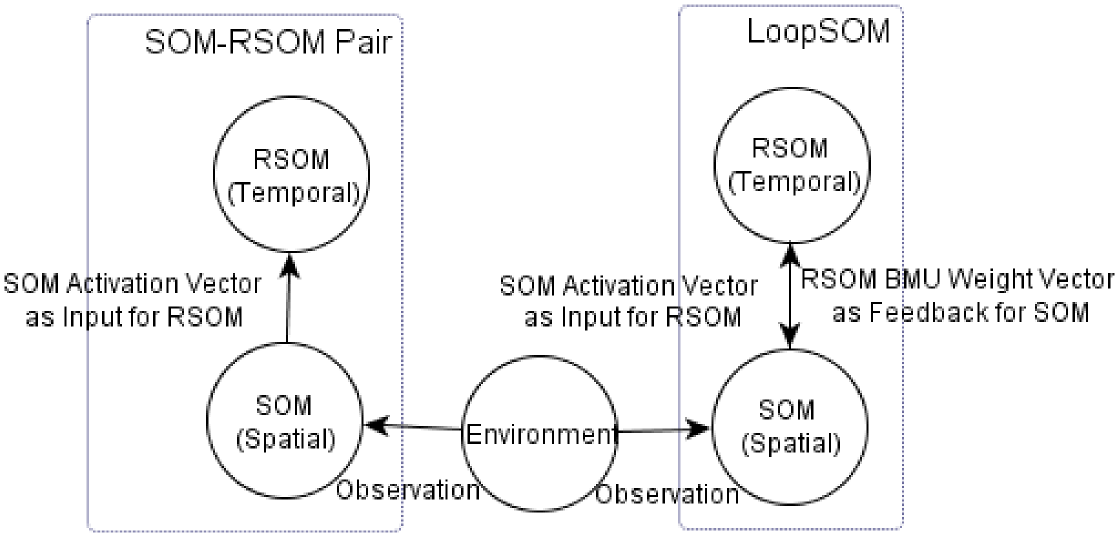

Figure 3.1: The SOM-RSOM structure compared to the new LoopSOM structure.

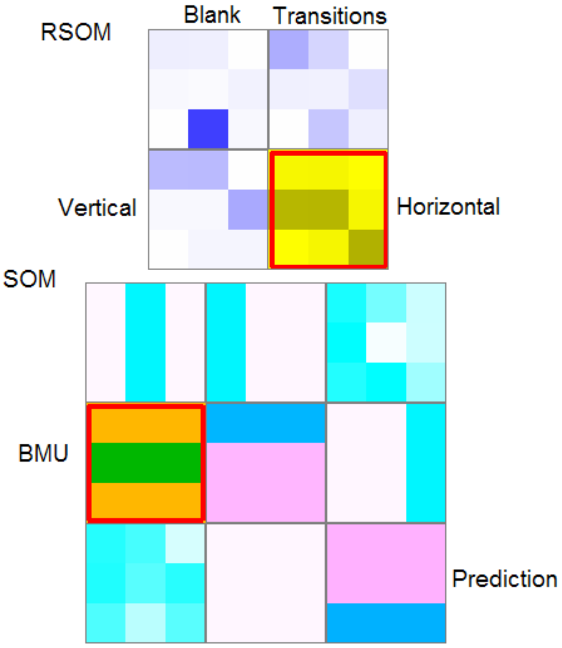

Figure 3.2: The LoopSOM. Spatial SOM bellow, temporal RSOM above. Activations in yellow, predictions in purple, coincident activations and predictions in red/orange. The BMU is highlighted in red. The used inputs are the ones described in section 6.2.

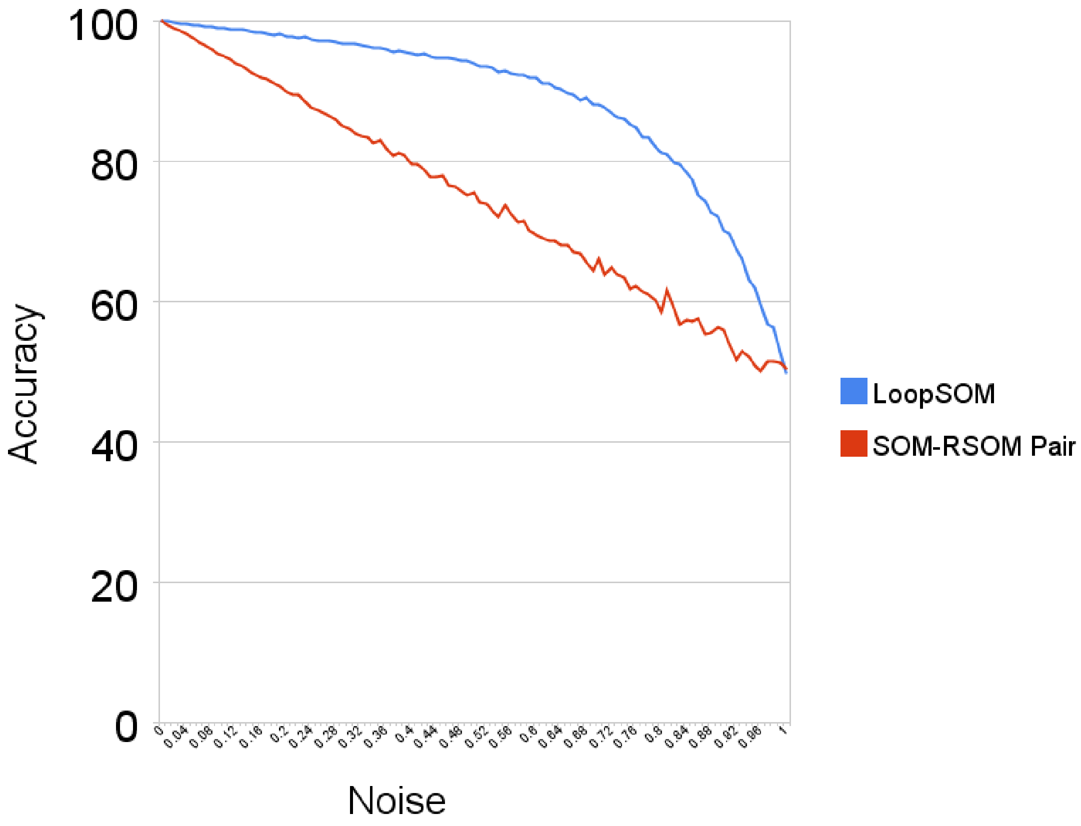

Figure 3.3: Results comparing the SOM-RSOM pair and the LoopSOM accuracy.

3.1 Possible Alternatives

Other possible replacements for the SOM-RSOM pair are Swarup's Anticipatory SOM (2005) and Recurrent AntSOM (2005b). The former performs simple predictions using activation counters and uses only a conventional SOM, while the later makes predictions using Simple Recurrent Networks (SRN, as from Elman (1990)). Possible advantages of the LoopSOM over them are the adaptive weighting and the explicit invariant representation, but it's out of the scope of this work to compare such algorithms.

3.2 Experiments

To compare the SOM-RSOM pair and the LoopSOM performances, two simple experiments were created.

3.2.1 Four Points in 2D Space

Four points in 2D space were presented to both algorithms: $\{(0,0),(0,1),(1,0),(1,1)\}$. They are grouped by their first term: group0 = $\{(0,0),(0,1)\}$, group1 = $\{(1,0),(1,1)\}$ and are presented in a way that points in the same group are more likely to show in the next step (90% chance), resulting in 50% for each group. After 5000 steps the weights in the SOM and RSOM are frozen and each SOM unit is labeled with 0 or 1 according to the best matches for each of them. Then the same points are presented, but now there's some chance of showing a totally ambiguous point (0.5,0.5). The tests involved chances within the range from 0% to 100%, with 1% increments. The parameters used were:

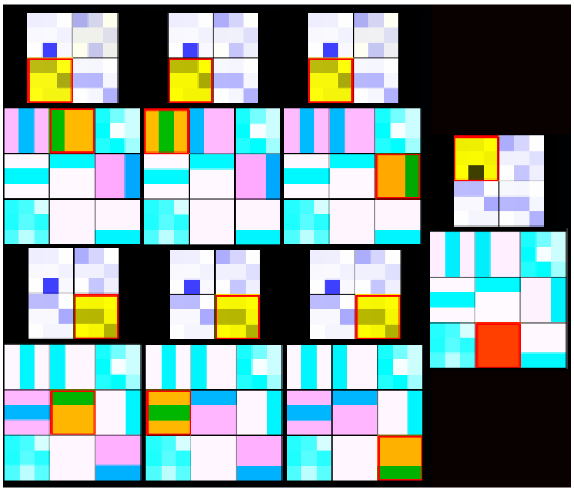

Figure 3.4: All 7 inputs shown to the LoopSOM and its respective states. Vertical on the top , horizontal on the bottom and blank on the right. Note how each group has only 1 invariant representation on the RSOM.

- SOM: size = 2x2, $\rho = 0.125$, $\sigma = 1$, γ = 0.1 (fixed, not adaptive);
- RSOM: size = 2x1, $\rho = 0.125$, $\alpha = 0.3$, $\sigma = 0.7$, γ = 0.01 (fixed, not adaptive);
- For the SOM-RSOM pair, the same implementation was used but with fixed $\xi_s = 1$ and $\xi_t = 0$.

Results are shown in Figure 3.3. The LoopSOM dominates the SOM-RSOM pair, although they are similar on extreme cases (0% and 100%). Note that the errors on the SOM-RSOM pair approximate the a posteriori probabilities of showing an ambiguous pattern with equal priors for each group (50%). The LoopSOM adds information to this prior probabilities, increasing accuracy by moving it from maximum likelihood to maximum a posteriori.

3.2.2 2D Lines

In the next experiment, 7 different 2D visual patterns with 3x3 pixels are presented: 3 horizontal lines, 3 vertical lines and 1 empty pattern (Figure 3.4). They are grouped as horizontal, vertical and blank, and patterns in the same group has 90% chance of showing in next step, resulting in nearly 33% for each group. After 20000 steps the weights in the SOM and RSOM are frozen and each SOM unit is labeled according to the best matches for each group. Then the same patterns are presented, but now there's some chance of showing a totally ambiguous pattern (a cross). Chances within the range from 0% to 100%, with 1% increments were tested. The used parameters were:

- SOM: size = 3x3, $\rho = 0.1875$, $\sigma = 0.3$, γ = 0.1 (fixed, not adaptive);
- RSOM: size = 2x2, $\rho = 0.125$, $\alpha = 0.2$, $\sigma = 0.7$, γ = 0.01 (fixed, not adaptive);
- For the SOM-RSOM pair, the same implementation was used but with fixed $\xi_s = 1$ and $\xi_t = 0$.

Results are shown in Figure 3.5. They're similar to the previous experiment, except that the prior probabilities are nearly 33% (3 groups) now.

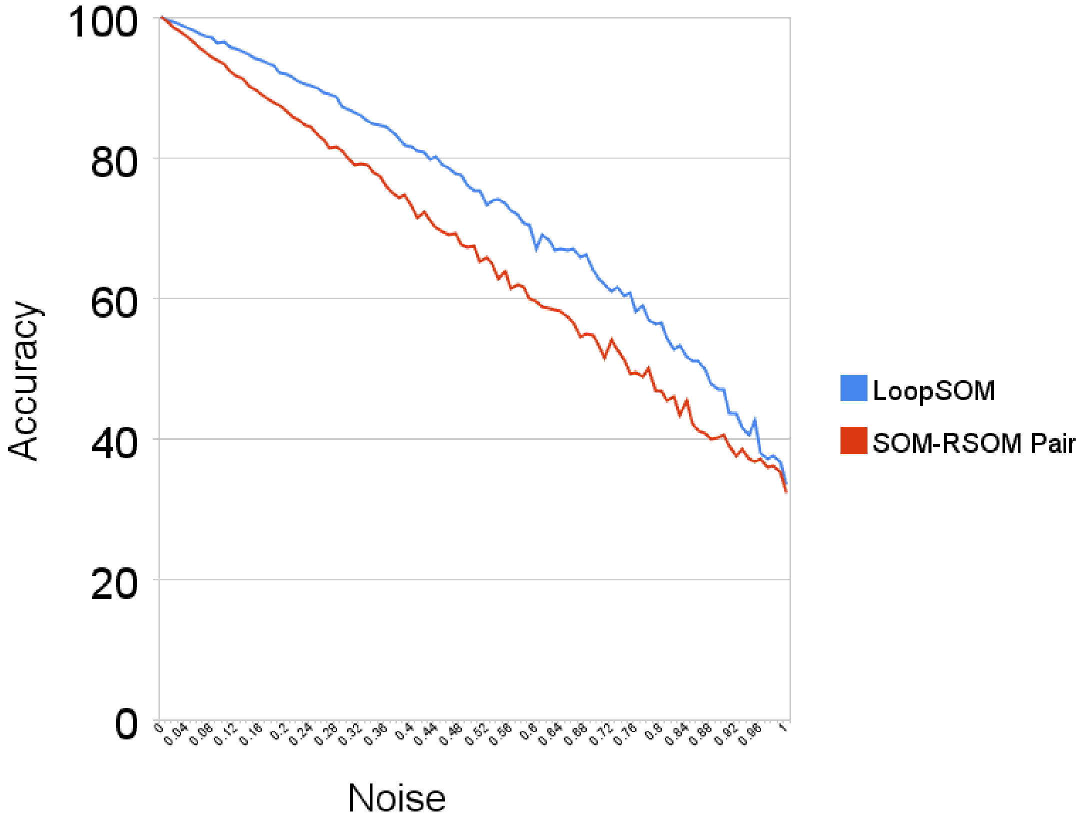

Figure 3.5: Results comparing the SOM-RSOM pair and the LoopSOM accuracy on a 2D line classification task.

4 IMPROVING THE HQSOM WITH LOOPSOM: THE HQSOM+

Simply replacing the SOM-RSOM pair for a LoopSOM in the HQSOM isn't sufficient, since the LoopSOM improves only the spatial SOM, while the HQSOM uses the temporal RSOMs outputs as inputs for the higher layers. To reflect the LoopSOM improvements in the RSOM too, the SOM activation vector isn't adequate as input for the RSOM. Instead, the combination of activation and prediction, y_b in equation 3.2, will be used as the RSOM input. Various different combination formulas were tryied and deserve a deeper analisys in the future, but 3.2 will be used for now. The improved version is called HQSOM+.

4.1 Experiments

To evaluate the HQSOM+ an experiment with 2D shapes is set, similar to the experiment conducted by Miller with the HQSOM. There are 3 pattern groups: square, x and diamond. Each one is composed by 5x5 2D shapes with scale and shift transformations. The shapes are parsed into 4 3x3 overlapped inputs for 4 LoopSOMs in the first layer. The second layer is composed by only 1 LoopSOM, which receives the combined BMU coordinates from the first layer as inputs. As with the LoopSOM experiments, an ambiguous pattern will be presented with some probability in the range from 0% to 100%.

As can be seen in figure 4.1, while the HQSOM starts do degrade at 60% ambiguity, the HQSOM+ resists until 80% without any accuracy loss. The problem is that this result requires very specific parameter values to actually take advantage from the LoopSOMs. The parameters used were:

- Layer 1 SOMs: size = 3x3, $\rho = 0.1875$, $\sigma = 0.3$, γ = 0.1 (fixed, not adaptive);
- Layer 1 RSOMs: size = 2x2, $\rho = 0.125$, $\alpha = 0.2$, $\sigma = 0.7$, γ = 0.01 (fixed, not adaptive);
- Layer 2 SOM: size = 2x2, $\rho = 0.125$, $\sigma = 0.7$, γ = 0.02 (fixed, not adaptive);
- Layer 2 RSOM: size = 2x2, $\rho = 0.125$, $\alpha = 0.9$, $\sigma = 0.7$, γ = 0.01 (fixed, not adaptive);

However, the results were very unstable, and more study is necessary to find a good parameter value set.

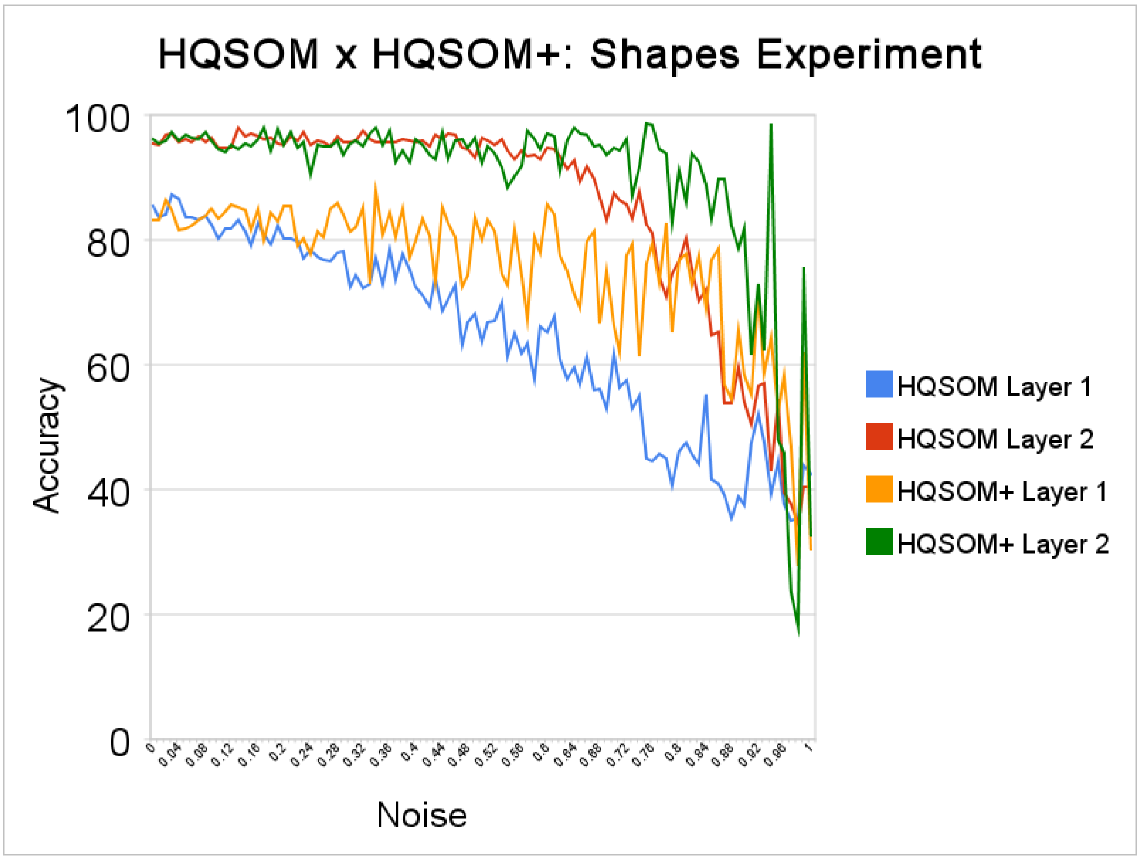

Figure 4.1: Results comparing the HQSOM and the HQSOM+ accuracy on a 2D shape classification task.

5 CONCLUSIONS

This work has shown how to create a more robust variant of the SOM-RSOM pair with little modification. By using the RSOM weight vector of its BMU, a good predictor for the SOM activations is obtained. Also, it is closer to the MPF than the previous model. The resulting algorithm, the LoopSOM, was applied to a hierarchical SOM algorithm, the HQSOM, making it more robust too, and leaving it just one step away from a complete implementation of the MPF.

There is still room for many improvements over the LoopSOM, for instance:

- improve the prediction combination formula, preferably with a strong probabilistic basis
- replace the current predictor (RSOM BMU weight vector) by a SRN;
- replace the RSOM or the entire SOM-RSOM pair by a Voegtlin's Recursive SOM (2002) / RecSOM;
- use the PLSOM adaptive parameters;

Also, the HQSOM+ may be improved by some modifications:

- find a better parameter value set
- replace the LoopSOM with an improved version, with the improvements cited above;
- replace the LoopSOM for the AntSOM or the RecAntSOM, if they prove to be better then the final LoopSOM version;
- introduce feedback connections between layers, increasing robustness and finishing the MPF implementation;

All of these items will be explored in future works, as well as more complex experiments and a deeper analysis of the formulas and results.

Appendix

APPENDIX A SUMÁRIO EM PORTUGUÊS (PORTUGUESE SUMMARY)

O que segue é uma versão reduzida deste trabalho no idioma português, para fins de melhor compreensão e alcance.

A.1 Introdução

Anteriormente, Hawkins propôs um framework para modelagem computacional do neocórtex, o Memory-Prediction Framework (framework memória-predição). Basicamente ele define unidades básicas de processamento executando o mesmo algoritmo em todo modelo, independentemente de modalidade sensorial ou nível hierárquico.

Já existem algumas implementações do MPF, tais como o algoritmo HTM (Hierarchical Temporal Memory ou Memória Hierárquica Temporal) e o HQSOM (Hierarquical Quilted Self-Organizing Map).

Este trabalho propõe introduzir conexões de realimentação no algoritmo HQSOM. Para tanto, suas unidades básicas de processamento, os pares SOM-RSOM, são substituídas por um novo algoritmo, o LoopSOM.

A.2 Trabalhos Relacionados

A.2.1 Memory-Prediction Framework

O MPF define unidades básicas de processamento executando o mesmo algoritmo em todo modelo, independentemente de modalidade sensorial ou nível hierárquico. Cada unidade executa um processamento espacial e temporal sobre seus sinais de entrada, e repassa seus resultados para camadas superiores de unidades. É feita uma analogia entre o processamento espacial e as células simples do neocórtex, bem com entre o processamento temporal e as suas células complexas. Assim como o ambiente serve de entrada para as unidades do primeiro nível, as saídas das unidades do primeiro nível servem de entradas para o segundo nível, e assim sucessivamente.

A.2.2 Hierarchical Temporal Memory

O HTM é uma implementação Bayesiana do MPF. De fato, ele pode ser visto como uma Rede Bayesiana Hierárquica com propagação de crenças. Cada unidade básica de processamento computa sua própria probabilidade em relação aos sinais de entrada, com ajuda de algoritmos de clusterização. As probabilidades resultantes tornam-se entradas para as camadas subsequentes, que as processam exatamente da mesma forma.

O algoritmo HTM já obteve algumas aplicações de sucesso como em um classificador geral de imagens (NuPic), um reconhecedor de dígitos manuscritos e um reconhecedor de captchas (imagens usadas para bloquear acesso automatizado a sites na web).

Uma desvantagem básica do HTM em relação ao HQSOM é o fato de que, ao menos atualmente, ele suporta apenas aprendizado em lote.

A.2.3 Hierarchical Quilted Self-Organizing Map

O HQSOM é uma implementação neural do MPF. Suas unidades básicas de processamento são compostas por mapas auto-organizáveis (SOM). SOMs simples são usados para o processamento espacial, enquanto SOMs recorrentes (RSOMs)s são usados para o processamento temporal. Os índices dos neurônios vencedores de cada par SOM-RSOM são passados como entradas para suas respectivas camadas subsequentes. Além disso, alguns mecanismos de ajuste de parâmetros foram usados para permitir que o algoritmo funcione de forma online, ou seja, não existe distinção entre fase de aprendizado e execução.

A.3 LoopSOM

O LoopSOM consiste em um melhoramento do par SOM-RSOM. Tal modificação é obtida com uma realimentação do RSOM para o SOM, fornecendo predições a respeito dos padrões espaciais. Com isso, certos casos de ambiguidade passam a ser resolvidos, como por exemplo, quando o SOM chega a um empate entre 2 ou mais neurônios vencedores. É feito um desempate entre os neurônios com base no que o processamento temporal espera que seja a entrada seguinte. O sinal de realimentação é simplesmente o vetor correspondente ao último neurônio vencedor do RSOM, o que coincide com um vetor de ativações do SOM. As ativações correntes são combinadas com a realimentação de forma que as ambiguidades puramente espaciais são desfeitas. Tal combinação foi obtida de forma empírica neste trabalho e merece um estudo mais aprofundado.

A.3.1 Experimentos

O LoopSOM obteve excelentes resultados em relação ao par SOM-ROM em 2 experimentos básicos.

A.3.1.1 4 Pontos 2D

No primeiro experimento, 4 pontos em um espaço 2D são apresentados com entradas e os algoritmos devem classificá-los como sendo do lado esquerdo ou direto. A sequência de apresentaç ao é feita de forma que padrões das mesmas classes apareçam próximos no tempo. Após o treinamento, um quinto ponto totalmente ambíguo (mesma distância até os outros 4 pontos) é apresentado com uma frequência cada vez maior. O par SOM-RSOM teve uma queda linear de 100% a 50% conforme o aumento da frequência, correspondendo exatamente ao cálculo de probabilidade a posteriori com valores a priori idênticos para ambas as classes. Já o LoopSOM teve uma queda bem mais lenta, sendo aproximadamente linear entre 100% e 64% de frequencia de ambiguidade, e com uma queda exponencial no restante do intervalo, até 50%. Tal resultado coincide com a introdução de valores a priori relevantes para as 2 classes em questão, obtidos pela predição.

A.3.1.2 Linhas 2D

No segundo experimento, 7 padr oes 3x3 diferentes s ao apresentados: 3 linhas horizontais com diferentes deslocamentos, 3 linhas verticais com diferentes deslocamentos e 1 padrão vazio. Os algoritmos devem classificar estes 7 padr oes em 3 classes: horizontal, vertical e vazio. Mais uma vez, padr oes das mesmas classes são apresentados próximos no tempo, o que ocorreria naturalmente em nosso campo visual ao observarmos o deslocamento de certos objetos. Após o treinamento, um padrão ambíguo é apresentado com frequência gradativamente maior. Tal padrão consiste em uma cruz, que tem distância igual entre uma linha horizontal central e uma linha vertical central, resultando em um empate entre essas 2 categorias. O par SOM-RSOM obteve uma degradação linear na sua taxa de classificações corretas, de 100% a 33%, mais uma vez conforme o que seria esperado em cálculos probabilsticos com valores a priori idênticos para 3 categorias. O LoopSOM teve uma queda mais suave, porém não tanto quanto a do experimento anterior. Vários fatores podem ter alguma influência sobre este resultado, tais como a adição de 1 categoria, a ambiguidade entre apenas 2 dessas 3 categorias, maior número de padrões e maior complexidade dos padrões.

A.4 HQSOM+

Substituindo os pares SOM-RSOM por LoopSOMs no HQSOM, o algoritmo HQSOM+ é obtido. Porém, a simples substituição não é suficiente: as entradas dos RSOMs são modificadas para não consistirem apenas nos mapas de ativação dos SOMs mas também nas predições. Ou seja, o resultado combinado utilizado para defazer ambiguidades deve também ser usado como entrada para os RSOMs. Tal modificação é necessária pois o LoopSOM foi criado com o objetivo de melhorar a classificação de sua porção espacial, mantendo a espacial intacta. Porém, o que é passado entre as camadas do HQSOM são as saídas dos RSOMs e portanto eles precisam ser beneficiados com suas próprias predições para que o HQSOM+ possa usufruir delas também.

A.4.1 Experimentos

Apenas 1 experimento foi executado para o HQSOM+. Foram apresentadas 4 categorias de padrões visuais 5x5: Quadrados, losangos, X's e o padrão vazio. Cada categoria é composta por formas com transformações de escala e translação. Tais padrões foram divididos em 4 regiões com sobreposição, sendo cada uma distribuída a 1 unidade de processamento da primeira camada. As saídas dos 4 RSOMs são concatenadas e usadas como entradas para a única unidade de processamento na segunda camada. Após o treinamento, 3 padrões ambíguos são apresentados com frequência gradativamente maior. Cada padrão é um intermediário entre 2 das 3 categorias não vazias. Não foi possível obter um conjunto ótimo de parâmetros para este experimento, mas foi demonstrado empiricamente que, com um ajuste adequado, é possível obter os resultados esperados após observar os resultados do LoopSOM. Enquanto o HQSOM se mantém próximo aos 100% de classificações corretas até 60% de frequência de ambiguidades, o HQSOM+ só começa a decair quando chega aos 80% de frequência ambiguidade, ambos encerrando em 25%. A análise deste caso é mais complexa devido Ã presença de replicação de dados e padrões maiores e mais complexos. Tal análise, bem como um melhor ajuste de parâmetros a fim de obter resultados mais estáveis, são deixados para trabalhos futuros.

A.5 Conclusões

Este trabalho apresentou uma variante do par SOM-RSOM, o algoritmo LoopSOM. Demonstrou-se empiricamente a superioridade do novo algoritmo sobre seu antecessor em 2 casos básicos. Os pares SOM-RSOM do HQSOM foram substituídos por LoopSOMs e o resultado do único experimento realizado demonstrou diversas dificuldades com o ajuste de parâmetros do algoritmo, mas foi possável apresentar resultados satisfatórios. Futuramente, as fórmulas utilizadas receberão um tratamento mais formal, e experimentos mais complexos serão realizados.

REFERENCES

BEHNKE, S.; ROJAS, R. Neural abstraction pyramid: a hierarchical image understanding architecture. In: NEURAL NETWORKS PROCEEDINGS, 1998. IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE. THE 1998 IEEE INTERNATIONAL JOINT CONFERENCE ON, 1998. **Anais. . .** [S.l.: s.n.], 1998. v.2.

BERGLUND, E. Improved PLSOM algorithm. **Applied Intelligence**, [S.l.], p.1–9, 2009.

BERGLUND, E.; SITTE, J. The parameter-less SOM algorithm. In: ANZIIS, 2003. **Proceedings. . .** [S.l.: s.n.], 2003. p.159–164.

COHEN, L. An information-processing approach to infant perception and cognition. **The development of sensory, motor, and cognitive capacities in early infancy**, [S.l.], p.277–300, 1998.

CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to support vector machines**. [S.l.]: Cambridge university press, 2000.

DEAN, T. **A Computational Model of the Cerebral Cortex**. 2005. 938p. v.20, n.2.

ELMAN, J. Finding structure in time. **Cognitive science**, [S.l.], v.14, n.2, p.179–211, 1990.

FOLDIAK, P.; ENDRES, D. Sparse coding. , [S.l.], 2008.

FOLDIAK, P.; YOUNG, M. Sparse coding in the primate cortex. **The handbook of brain theory and neural networks**, [S.l.], p.895–898, 1995.

FUKUSHIMA, K. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, [S.l.], v.36, n.4, p.193–202, 1980.

GEORGE, D.; JAROS, B.; INC, N. The HTM learning algorithms. **March**, [S.l.], v.1, p.2007, 2007.

GUIDE, N. Numenta Platform for Intelligent Computing. , [S.l.], 2008.

GUIMARAES, G. A taxonomy of Self-organizing Maps for temporal sequence processing. **Intelligent Data Analysis**, [S.l.], v.7, n.4, p.269–290, 2003.

HALL, Y. J.; POPLIN, R. E. Using Numentas hierarchical temporal memory to recognize CAPTCHAs. , [S.l.], 2007.

HAWKINS, J. **On Intelligence**. [S.l.]: Owl Books, 2005.

HAWKINS, J.; GEORGE, D. Hierarchical Temporal Memory–Concepts, Theory, and Terminology. **Whitepaper, Numenta Inc, May**, [S.l.], 2006.

KOHONEN, T. Self-organization and associative memory. , [S.l.], 1989.

KOHONEN, T. The self-organizing map. **Neurocomputing**, [S.l.], v.21, n.1-3, p.1–6, 1998.

KOSKELA, T.; VARSTA, M.; HEIKKONEN, J.; KASKI, K. **Temporal sequence processing using recurrent SOM**. 1998. v.1.

MILLER, J.; LOMMEL, P. Biomimetic sensory abstraction using hierarchical quilted self-organizing maps. In: SPIE, 2006. **Proceedings...** [S.l.: s.n.], 2006. v.6384, p.63840A.

MINSKY, M.; PAPERT, S. **Perceptrons: an introduction to computational geometry**. [S.l.]: MIT press Cambridge, Mass, 1969.

PINTO, R.; ENGEL, P. LoopSOM: a robust som variant using self-organizing temporal feedback connections. In: TO APPEAR ON PROCEEDINGS OF THE VIII ENIA - BRAZILIAN MEETING ON ARTIFICIAL INTELLIGENCE, 2009. **Anais...** [S.l.: s.n.], 2009.

RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In: IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, 1993. **Anais...** [S.l.: s.n.], 1993.

RIESENHUBER, M.; POGGIO, T. Hierarchical models of object recognition in cortex. **nature neuroscience**, [S.l.], v.2, p.1019–1025, 1999.

SWARUP, S.; LAKKARAJU, K.; KLEMENTIEV, A.; RAY, S.; SASSEN, E.; MUNCHEN, G. An anticipatory self-organized map for robust recognition. In: IASTED INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND APPLICATIONS (AIA'05), INNSBRUCK, AUSTRIA, 2005. **Proceedings...** [S.l.: s.n.], 2005.

SWARUP, S.; LAKKARAJU, K.; SMITH, N.; RAY, S. A Recurrent Anticipatory Self-Organized Map for Robust Time Series Prediction. In: THE WORKSHOP ON SELF-ORGANIZING MAPS (WSOM), PARIS, FRANCE, SEP 5-8, 2005., 2005. **Anais...** [S.l.: s.n.], 2005.

THORNTON, J.; GUSTAFSSON, T.; BLUMENSTEIN, M.; HINE, T. Robust Character Recognition Using a Hierarchical Bayesian Network. **Lecture Notes in Computer Science**, [S.l.], v.4304, p.1259, 2006.

VOEGTLIN, T. Recursive self-organizing maps. **Neural Networks**, [S.l.], v.15, n.8-9, p.979–991, 2002.