On-Line Learning with Structural Adaptation in a Network of Spiking Neurons for Visual Pattern Recognition

Simei Gomes Wysoski, Lubica Benuskova, and Nikola Kasabov

Knowledge Engineering and Discovery Research Institute,
Auckland University of Technology, 581-585 Great South Rd,
Auckland, New Zealand
{swysoski, lbenusko, nkasabov}@aut.ac.nz
http:www.kedri.info

Abstract. This paper presents an on-line training procedure for a hierarchical neural network of integrate-and-fire neurons. The training is done through synaptic plasticity and changes in the network structure. Event driven computation optimizes processing speed in order to simulate networks with large number of neurons. The training procedure is applied to the face recognition task. Preliminary experiments on a public available face image dataset show the same performance as the optimized off-line method. A comparison with other classical methods of face recognition demonstrates the properties of the system.

1 Introduction

The human brain has been modelled in numerous ways, but these models are far from reaching comparable performance. These models are still not as general and accurate as the human brain despite that outstanding performances have been reported [1] [2] [3]. Of particular interest to this research are the models for visual pattern recognition. Visual pattern recognition models can be divided in two groups according to the connectionist technique applied. Most of the works deal with the visual pattern recognition using neural networks comprised of linear/non-linear processing elements based on the neural rate-based code [4] [5]. Here we refer to these methods as traditional methods. In another direction, a visual pattern recognition system can be constructed through the use of brain-like neural networks.

Brain-like neural networks are networks that have a closer association with what is known about the way brains process information. The definition of brain-like networks is intrinsically associated with the computation of neuronal units that use pulses. The use of pulses brings together the definitions of time varying postsynaptic potential (*PSP)*, firing threshold (ϑ), and spike latencies (Δ), as depicted in Figure 1 [6]. Brain-like neural networks, despite being more biologically accurate, have been considered too complex and cumbersome for modeling the proposed task. However recent discoveries on the information processing capabilities of the brain and technical advances related to massive parallel processing, are bringing back the idea of using biologically realistic networks for pattern recognition. A recent pioneering work has shown that the primate (including human) visual system can analyze complex

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 61 – 70, 2006.

natural scenes in only about 100-150 ms [7]. This time period for information processing is very impressive considering that billions of neurons are involved. This theory suggests that probably neurons, exchanging only one or few spikes, are able to form assemblies, and process information. As an output of this work, the authors proposed a multi-layer feed-forward network (SpikeNet) of integrate-and-fire neurons that can successfully track and recognize faces in real time [7].

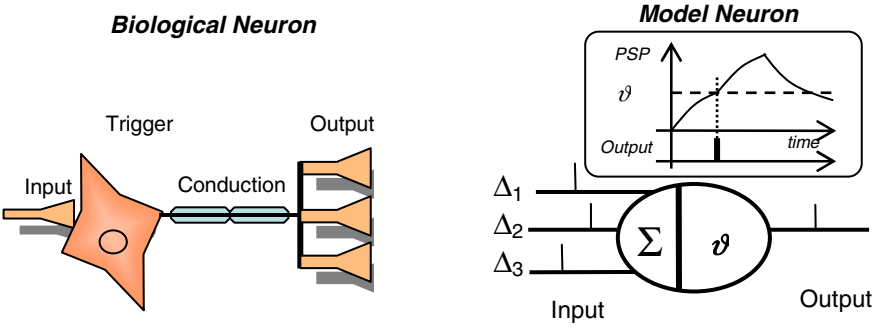

Fig. 1. On the left: Representation of biological neuron. On the right: Basic artificial unit (spiking neuron).

This paper intends to review the network model SpikeNet proposed in [8] and extend its applicability to perform on-line learning. In the next sections the spiking neural network model will be presented and the new learning procedure will be described. The new learning method is applied to the face recognition task. The results are compared with previous work and other models. Discussion and additional required analysis concludes the paper.

2 Spiking Network Model

In this section we describe the steps of the biologically realistic model used in this work to perform on-line visual pattern recognition. The system has been implemented based on the SpikeNet introduced in [7] [8] [9] [10]. The neural network is composed of 3 layers of integrate-and-fire neurons. The neurons have a latency of firing that depends upon the order of spikes received. Each neuron acts as a coincidence detection unit, where the postsynaptic potential for neuron i at a time t is calculated as:

$$PSP(i,t) = \sum \mathrm{mod}^{order(j)} w_{j,i} \tag{1}$$

where mod $\in (0,1)$ is the modulation factor, j is the index for the incoming connection and $w_{j,i}$ is the corresponding synaptic weight. See [7] [9] for more details.

Each layer is composed of neurons that are grouped in two-dimensional grids forming neuronal maps. Connections between layers are purely feed-forward and each neuron can spike at most once on spikes arrival in the input synapses. The first layer cells represent the ON and OFF cells of retina, basically enhancing the high contrast parts of a given image (high pass filter). The output values of the first layer are encoded to

pulses in the time domain. High output values of the first layer are encoded as pulses with short time delays while long delays are given to low output values. This technique is called Rank Order Coding [10] and basically prioritizes the pixels with high contrast that consequently are processed first and have a higher impact on neurons' PSP.

Second layer is composed of eight orientation maps, each one selective to a different direction (0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°). It is important to notice that in the first two layers there is no learning, in such a way that the structure can be considered simply passive filters and time domain encoders (layers 1 and 2). The theory of contrast cells and direction selective cells was first reported by Hubel and Wiesel [11]. In their experiments they were able to distinguish some types of cells that have different neurobiological responses according to the pattern of light stimulus.

The third layer is where the learning takes place and where the main contribution of this work is presented. Maps in the third layer are to be trained to represent classes of inputs. See Figure 2 for the complete network architecture. In [7], the network has a fixed structure and the learning is done off-line using the rule:

$$\Delta w_{j,i} = \frac{\mathrm{mod}^{order(a_j)}}{N} \tag{2}$$

where $w_{j,i}$ is the weight between neuron j of the 2nd layer and neuron i of the 3rd layer, mod $\in$ (0,1) is the modulation factor, $order(a_j)$ is the order of arrival of spike from neuron j to neuron i, and N is the number of samples used for training a given class.

In this rule, there are two points to be highlighted: a) the number of samples to be trained needs to be known *a priori*; and b) after training, a map of a class will be selective to the average pattern.

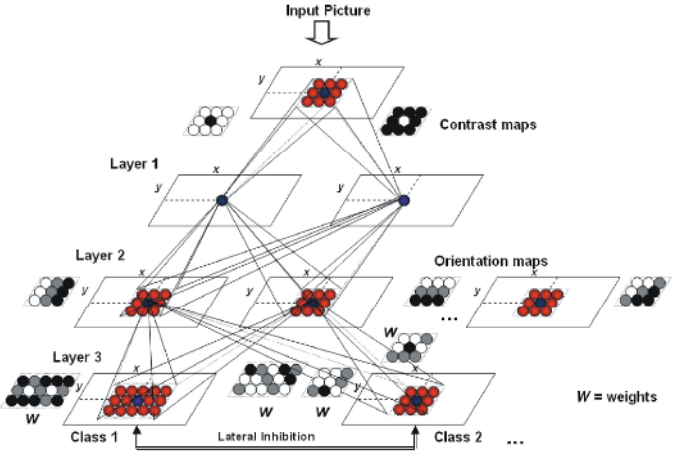

Fig. 2. Adaptive spiking neural network (aSNN) architecture for visual pattern recognition

There are also inhibitory connections among neuronal maps in the third layer, so that when a neuron fires in a certain map, other maps receive inhibitory pulses in an area centred in the same spatial position. An input pattern belongs to a certain class if a neuron in the corresponding neuronal map spikes first.

One of the properties of this system is the low activity of the neurons. It means that the system has a large number of neurons, but only few take active part during the retrieval process. In this sense, through the event driven approach the computational performance can be optimized [8] [12]. Additionally, in most cases the processing can be interrupted before the entire simulation is completed. Once a single neuron of the output layer reaches the threshold to emit a spike the simulation can be finished. The event driven approach and the early simulation interruption make this method suitable for implementations in real time.

3 On-Line Learning and Structural Adaptation

3.1 General Description

Our new approach for learning with structural adaptation aims to give more flexibility to the system in a scenario where the number of classes and/or class instances is not known at the time the training starts. Thus, the output neuronal maps need to be created, updated or even deleted on-line, as the learning occurs. In [13] a framework to deal with adaptive problems is proposed and several methods and procedures describing adaptive systems are presented.

To implement such a system the learning rule needs to be independent of the total number of samples since the number of samples is not known when the learning starts. Thus, in the next section we propose to use a modified equation to update the weights based on the average of the incoming patterns. It is important to notice that, similarly to the batch learning implementation of Equation 2, the outcome is the average pattern. However, the new equation calculates the average dynamically as the input patterns arrive.

There is a classical drawback to learning methods when, after training, the system responds optimally to the average pattern of the training samples. The average does not provide a good representation of a class in cases where patterns have high variance (see Figure 3). A traditional way to attenuate the problem is the *divide-and-conquer* procedure. We implement this procedure through the structural modification

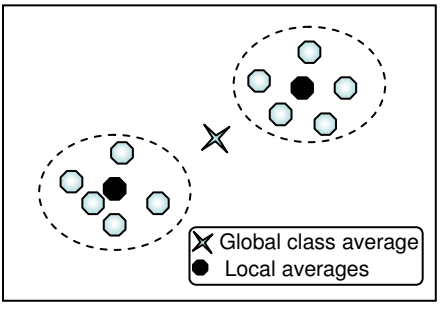

Fig. 3. *Divide and conquer* procedure to deal with high intra class variability of patterns in the hypothetical space of class *K*. The use of multiple maps that respond optimally to the average of a subset of patterns provides a better representation of the classes.

of the network during the training stage. More specifically, we integrate into the training algorithm a simple clustering procedure: patterns within a class that comply with a similarity criterion are merged into the same neuronal map. If the similarity criterion is not fulfilled, a new map is generated. The entire training procedure follows 4 steps described in the next section and is summarized in the flowchart of Figure 4.

3.2 Learning Procedure

The new learning procedure can be described in 4 sequential steps:

1. Propagate a sample k of class K for training into the layer 1 (retina) and layer 2 (direction selective cells – DSC);
2. Create a new map $Map_{C(k)}$ in layer 3 for sample k and train the weights using the equation:

$$\Delta w_{j,i} = \mathrm{mod}^{order(a_j)} \tag{3}$$

where $w_{j,i}$ is the weight between neuron j of the layer 2 and neuron i of the layer 3, mod $\in$ (0,1) is the modulation factor, $order(a_j)$ is the order of arrival of spike from neuron j to neuron i.

The postsynaptic threshold ($PSP_{threshold}$) of the neurons in the map is calculated as a proportion $c \in [0,1]$ of the maximum postsynaptic potential (PSP) created in a neuron of map $Map_{C(k)}$ with the propagation of the training sample into the updated weights, such that:

$$PSP_{threshold} = c \max(PSP) \tag{4}$$

The constant of proportionality c express how similar a pattern needs to be to trigger an output spike. Thus, c is a parameter to be optimized in order to satisfy the requirements in terms of false acceptance rate (FAR) and false rejection rate (FRR).
3. Calculate the similarity between the newly created map $Map_{C(k)}$ and other maps belonging to the same class $Map_{C(K)}$. The similarity is computed as the inverse of the Euclidean distance between weight matrices.
4. If one of the existing maps for class K has similarity greater than a chosen threshold $Th_{simC(K)}>0$, merge the maps $Map_{C(k)}$ and $Map_{C(K\text{similar})}$ using arithmetic average as expressed in equation:

$$W = \frac{W_{Map_{C(k)}} + N_{samples} W_{Map_{C(Ksimilar)}}}{1 + N_{samples}} \tag{5}$$

where matrix W represents the weights of the merged map and $N_{samples}$ denotes the number of samples that have already being used to train the respective map. In similar fashion the $PSP_{threshold}$ is updated:

$$PSP_{threshold} = \frac{PSP_{Map_{C(k)}} + N_{samples} PSP_{Map_{C(Ksimilar)}}}{1 + N_{samples}} \tag{6}$$

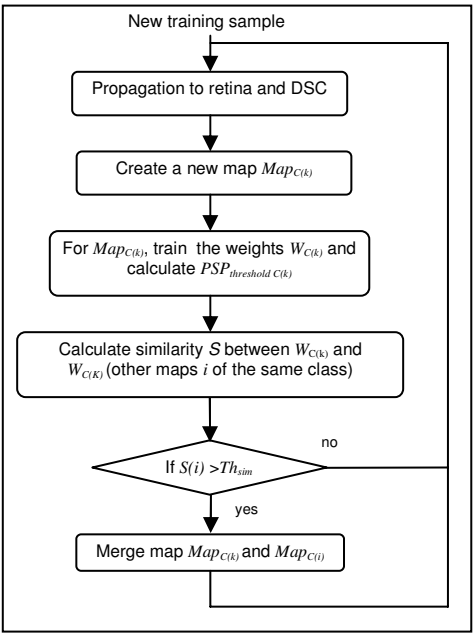

Fig. 4. On-line learning procedure flowchart

4 Experiments and Results

We have implemented the learning procedure proposed in the previous section in a network of spiking neurons as described in section 2. To evaluate the performance and compare with previous work, we used the same dataset as in [7], which is available from [14]. The dataset is composed of 400 faces taken from 40 different people. The frontal views of faces are taken in rotation angles varying in the range of [-30°, 30°].

4.1 Image Preparation

We manually annotated the position of eyes and mouth and used it to centralize the face images. The faces were rotated to align the right and left eyes horizontally. The boundaries of our region of interest (ROI) were then defined as a function of the inter-ocular distance and the distance between the eyes and mouth. The ROI is then normalized to the size 20 x 30 pixels in greyscale. The 2 dimensional array obtained has been used as input to the SNN. No contrast or illumination manipulation has been performed as previous work demonstrated the good response of the network under the presence of noise and illumination changes [7].

4.2 Spiking Network Parameters

The neuronal maps of retina, DSC and output maps have size of 20 x 30. The number of time steps used to encode the output of retina cells to the time domain is set to 100.

The threshold for the direction selective cells is set to 600, chosen in such a way that on average only 20% of neurons emits output spikes. The modulation factor mod $\in$ (0, 1) is set to 0.98. In this way the efficiency of the input of a given neuron is reduced to 50% when 50% of the inputs get a spike. The retina filters are implemented using a 5 x 5 Gaussian grid and direction selective filters are implemented using Gabor functions in a 7x7 grid. All these parameters were not optimized. Rather, we tried to reproduce as close as possible the scenario described in [7] for comparison purposes.

4.3 Results

Previous work demonstrated the high accuracy of the network to cope with noise, contrast and luminance changes, reaching 100% in the training set (10 samples for each class) and 97.5% when testing the generalization properties [7]. For the generalization experiment, the dataset was divided in 8 samples for training and the remaining 2 for test. With the adaptive learning method proposed here, we have obtained similar results for the training set.

In another experiment, to test the system ability to add on-line output maps for better generalization, we used only 3 sample images from each person for training. The remaining 7 views of each person were used for test. Among the dataset faces, we chose manually those samples taken from different angles that appeared to be most dissimilar. Thus, the training set was composed mostly of one face view taken from the left side (30°), one frontal view and one face view taken from the right side (-30°), as depicted in Figure 5. The results are shown in Table 1. In column 2 of Table 1, Th_{sim} is set in such a way that only one output map for each class is created. In such condition, the on-line learning procedure becomes equivalent to the original off-line learning procedure described by Equation 2. Tuning of Th_{sim} for performance, it can be clearly seen the advantage of using more maps to represent classes that contain highly variant samples, as the accuracy of face recognition increases by 6% with a reduction on the FAR.

In Table 2 and Table 3 is presented the network performance for different values of PSP threshold that are calculated as a function of the proportionality constant c. In all the experiments the constant c is the same for all maps and chosen prior to the training start. In a batch mode operation the value of c can be optimized independently for each map after the training is completed using, e.g., Genetic Algorithms (GA).

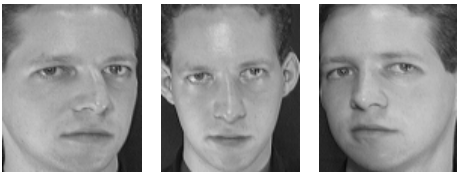

Fig. 5. Example of image samples used for training (30°, frontal and -30°)

Table 1. Results for the test set according to different similarity thresholds Th_{sim}.Three pictures of each class are used for training and the remaining seven for test.

Similarity threshold Th_{sim} (x10^{-3})	0.5	0.833	1.0	1.25	2
Number of output maps	40	47	80	109	120
Accuracy (%)	74.28	77.49	78.57	80.00	80.00
False Acceptance Rate (FAR) (%)	2.32	2.20	2.18	2.26	1.77
False Rejection Rate (FRR) (%)	0.00	0.00	0.00	0.00	0.00

Table 2. Accuracy for different values of c keeping Th_{sim} = 0.5x10-3. Output maps = 40

PSP threshold	c = 0.30	c = 0.35	c = 0.40	c = 0.45
Accuracy (%)	72.14	74.28	73.21	71.43
False Acceptance Rate (FAR) (%)	3.10	2.32	1.58	1.25
False Rejection Rate (FRR) (%)	0.00	0.00	0.00	2.50

Table 3. Accuracy for different values of c keeping Th_{sim} = 2.0x10-3. Output maps = 120

PSP threshold	c = 0.30	c = 0.35	c = 0.40	c = 0.45
Accuracy (%)	75.00	78.57	80.00	80.00
False Acceptance Rate (FAR) (%)	2.95	2.49	1.77	1.06
False Rejection Rate (FRR) (%)	0.00	0.00	0.00	3.57

In another comparison, to check how difficult the dataset is and to have a better idea of the performance of our learning algorithm, we compare the face recognition system using adaptive SNN (aSNN) with other three traditional methods of face recognition (Table 4). In these methods, PCA (principal component analysis) is used to extract facial features. The classification is done using SVM (support vector machine), MLP (multi layer perceptron) neural network and ECF (evolving classifier function). MLP and SVM are batch mode methods while ECF present similar adaptive learning characteristics as proposed in this work. ECF can be trained in both one-pass and recursive mode (several epochs)[13]. As expected, the batch mode algorithms over performed the one-pass on-line methods. The reason is that in the batch mode, the training samples are recursively presented to the classification method to minimize the output errors. In the one-pass on-line learning the adjustment of weights occurs only once at the time the training samples are presented to the network. Therefore, the performance of the batch methods can be considered roughly the target or the maximum accuracy that be reached. When comparing both one-pass online methods, the adaptive SNN presented better performance than ECF. Notice that, in this comparison we can not detect if the better performance is due to the learning method or to the different representation of the features.

Table 4. Comparison among different methods of face recognition (experiments using NeuCom [15])

Method	Accuracy (%)	Properties
PCA + SVM	90.7	Batch mode
PCA + MLP	89.6	Batch mode
PCA + ECF	74.0 (120 nodes)	One-epoch on-line method
Adaptive SNN	80.0 (109 maps)	One-pass on-line method

5 Discussion and Conclusion

A simple procedure to perform on-line learning in a network of spiking neurons has been presented. During learning, new output maps are created and merged based on the clustering of intra-class samples. Preliminary experiments have shown that the learning procedure reaches similar levels of performance of the previously presented work, and better performance can be reached in classes where samples have high variability. As a price, one more parameter needs to be tuned, e.g. Th_{sim}. In addition, more output maps require more storage memory.

In terms of normalization, the rank order codes are intrinsically invariant to changes in contrast and input intensities, basically because the neuronal units compute the order of the incoming spikes and not the latencies itself [7]. This can be a reason why adaptive SNN present better result than PCA+ECF as the feature extraction using PCA can degrade performance with illumination changes.

The adaptive SNN doesn't cope well with patterns rotation. In all the experiments presented in this work we aligned the samples in the image preparation stage. Alternatively, a certain degree of rotation invariance can be reached with the use of additional neuronal maps, in which each map need to be trained to cover different angles. In this case, the learning procedure described here, can automatically generate the new maps when it's required.

With respect to the overall system, the computation with pulses, contrast filters and orientation selective cells finds a close correspondence with traditional ways of image processing such as wavelets and Gabor filters [16] that already have proven to be very robust for feature extraction in visual pattern recognition problems. From the biological perspective, despite still being a very simplified representation of what effectively happens in the brain, the use of pulses is a starting point.

In our future work, aiming to improve the use of biologically realistic neural networks for pattern recognition, we intend to add adaptation to layer 1 and layer 2. It has been experimentally proven [17] that neural filters adaptively change to increase the information carried by the neural response. As a result, the contrast and direction selective cells are optimized filters to describe natural scenes. We intend to explore how to adaptively obtain optimal filters in different types of data.

Acknowledgments

The work has been supported by the NERF grant X0201 funded by FRST (L.B., N.K.) and by the Tertiary Education Commission of New Zealand (S.G.W.).

References

1. Fukushima, K.: Active Vision: Neural Network Models. In Amari, S., Kasabov, N. (eds.): Brain-like Computing and Intelligent Information Systems. Springer-Verlag (1997)
2. Mel, B. W.: SEEMORE: Combining colour, shape, and texture histrogramming in a neurally-inspired approach to visual object recognition. Neural Computation 9 (1998) 777-804

3. Wiskott, L., Fellous, J. M., Krueuger, N., von der Malsburg, C.: Face Recognition by Elastic Bunch Graph Matching: In Jain, L.C. et al. (eds.): Intelligent Biometric Techniques in Fingerprint and Face Recognition. CRC Press (1999) 355-396
4. Haykin, S.: Neural Networks - A Comprehensive Foundation. Prentice Hall (1999)
5. Bishop, C.: Neural Networks for Pattern Recognition. University Press, Oxford New York (2000)
6. Gerstner, W., Kistler, W. M.: Spiking Neuron Models. Cambridge Univ. Press, Cambridge MA (2002)
7. Delorme, A., Thorpe, S.: Face identification using one spike per neuron: resistance to image degradation. Neural Networks, Vol. 14. (2001) 795-803
8. Delorme, A., Gautrais, J., van Rullen, R., Thorpe, S.: SpikeNet: a simulator for modeling large networks of integrate and fire neurons. Neurocomputing, Vol. 26-27. (1999) 989-996
9. Delorme, A., Perrinet, L., Thorpe, S.: Networks of integrate-and-fire neurons using Rank Order Coding. Neurocomputing. (2001) 38-48
10. Thorpe, S., Gaustrais, J.: Rank Order Coding. In: Bower, J. (ed.): Computational Neuroscience: Trends in Research. Plenum Press, New York (1998)
11. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol, 160 (1962) 106-154
12. Mattia, M., del Giudice, P.: Efficient Event-Driven Simulation of Large Networks of Spiking Neurons and Dynamical Synapses. Neural Computation, Vol. 12 (10). (2000) 2305-2329
13. Kasabov, N.: Evolving Connectionist Systems: Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines. Springer-Verlag (2002)
14. http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html
15. http://www.aut.ac.nz/research/research_institutes/kedri/research_centres/centre_for_novel_methods_of_computational_intelligence/neucom.htm
16. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision, 2nd edn. (1998)
17. Sharpee, T. *et al*.: Adaptive filtering enhances information transmission in visual cortex. Nature, Vol. 439 (2006) 936-942