ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 285-290

A Correlation-Based Network for Real-Time Processing

Jey Ngole
Department of Computer Systems
Uppsala University, Box 325, S - 751 05, Uppsala, Sweden
contact email: jeyn@DoCS.UU.SE

Abstract

The network architecture presented in this paper maximizes correlations between the activities of the hidden units in order to preserve the internal structure of a given data pattern in the high dimensional space while discounting, to a certain extent, factors that are irrelevant to recognition. Consequently, the method of updating the weights of the network is exclusively Hebbian. Each unit in the hidden layer attempts to match its input-driven bottom-up information, from the preceding layer, with the parameter-based top-down information from the layer above. The parameter-based top-down information effectively eliminates the need to propagate the error derivatives from the top layer to the bottom one. Simulation results, that demonstrate the feasibility of the approach, are also presented.

1. Introduction

This paper presents a predominantly feedforward architecture originally proposed in [1] for use in on-line linear pattern recognition. A significant incentive behind the current resurgence of interest in the subject of artificial neural networks (ANNs) lies in the evidence [2] that these nets could be implemented in parallel hardware which can be operated in real-time and which makes the nets computationally faster. The network uses a Hebb-type rule which is local, (that is, only information about the input to a unit is required in updating the weights on its input links), with a relatively low computational requirement; an attractive feature for a hardware implementation of any neurally-inspired architecture.

In section 2, the basic properties of the one-hidden layer model network will be introduced, together with the model neuron used in the hidden layer. Section 3 describes the learning rules for the model. The results from two experiments are reported in section 4, followed by some conclusions in section 5.

2. Network Model

The model network shown in figure 2.1 consists of an input layer, a hidden layer with a self-organizing character and an output layer. In general, self-organization relies on the observations of such properties as the mean, variance and correlation matrix of the input vector [3]. The network of figure 2.1 makes use of the correlation matrix to preserve, as far as possible, the internal structure of a given data pattern in the high dimensional space while discounting, to a certain extent, factors that are irrelevant to recognition. Hence the architecture is simply referred to as a correlation-based network (CBN). Each hidden unit h consists of a predictor, h_p, and a matched filter, h_f (see inserts). Temporal correlations between units h and j are mediated by the oscillator, m, which generates a short–term memory (STM) of unit j, r_{hj}, for unit h and a STM of unit h, r_{jh}, for unit j. The prediction s_h, (slightly delayed by the unit element D^{-1}), and the net STM r_h are then matched by the filter h_f which subsequently generates y_h, the filtered component of s_h. The output weight vector is reciprocal, but with its influence on the hidden units scaled down with a small positive constant α. The scaled feedback biases the unit h towards those regions of the input that cause the greatest error at the output since the magnitude of each weight vector is proportional to its contribution in

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 285-290

reducing the output error. The output z_k, of the linear output unit k, is a weighted sum of the filtered hidden predictions y_h^μ.

2.1 Model Neuron

The formulation of the CBN matched filter is based on the hypothesis that the transformation of an input vector $\boldsymbol{x}^\mu$ (upon projection on the vector $\boldsymbol{w}_h$) defined as

$$\boldsymbol{w}_h\boldsymbol{x}^\mu \rightarrow y_h^\mu + y_h^\mu J_h^\mu(h,j) \tag{2.1}$$

can be seen as its decomposition into a filtered output y_h^μ, for unit h, and a "byproduct" which is a *correlation* between y_h^μ and the cost, $J_h^\mu(h,j)$, of using the oscillator m to cooperate with unit j. Under the natural constraint that

$$\boldsymbol{w}_h\boldsymbol{x}^\mu \geq y_h^\mu + y_h^\mu J_h^\mu(h,j), \tag{2.2}$$

it is easy to see that the cost, $J_h^\mu(h,j)$, has a form of *inverse* relationship with the matched filter output y_h^μ, which is the solution of equation (2.1). In other words,

$$y_h^\mu \leftarrow \frac{\boldsymbol{w}_h\boldsymbol{x}^\mu}{1 + J_h^\mu(h,j)}, \qquad h = 1, 2, \ldots, H. \tag{2.3}$$

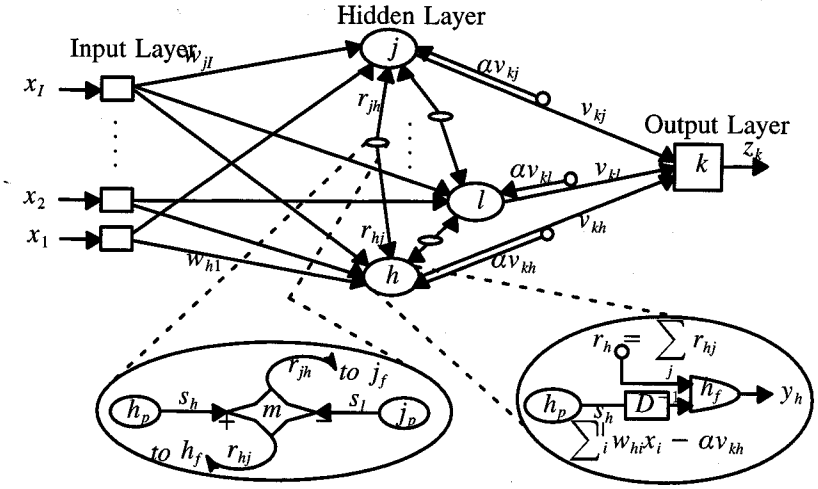

Figure 2.1 The CBN Model Architecture.

The objective here is to move hidden units towards regions of the input that contain the majority of the data. This can be achieved through a minimization of $J_h^\mu(h,j)$. The definition of $J_h^\mu(h,j)$ is based on the observations that biological neurons generate functional boundaries between clusters of high activities by making comparisons between their responses [4]. With this in mind, $J_h^\mu(h,j)$ is defined as follows:

$$\begin{aligned} J_h^\mu(h,j) &= \| (\boldsymbol{w}_h\boldsymbol{x}^\mu - \alpha v_{kh}) - q_h^\mu \sum_{j\neq h}^{n_h^\mu - 1} m_{hj}^\mu (\boldsymbol{w}_j\boldsymbol{x}^\mu - \alpha v_{kj}) \|^p \\ &= \left(s_h^\mu - r_h^\mu\right)^p \end{aligned} \tag{2.4}$$

where $\|.\|$ denotes a norm, p is an index that determines the degree of cooperation between units, $q_h^\mu = 1/n_h^\mu$, n_h^μ is the number of units in an instantaneous activity cluster

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 285-290

C_h^μ (see figure 2.2a) and m_{hj}^μ is the *correlation length* between units h and j [1]. Although distinct unit assemblies can be identified, there also exists *weak* links (between clusters) that convey intercluster interactions on presentation of a pattern μ. Overlaps between clusters provide a distributed representation for the data [5].

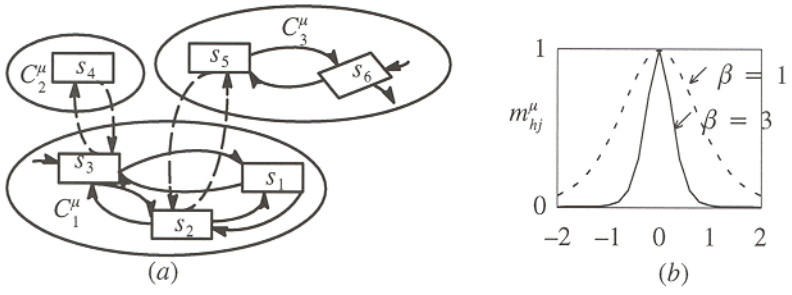

Figure 2.2 (a) Illustration of temporal cluster formation in a CBN. Dashed arrows denote weak links that convey intercluster interactions. Due to the presence of such overlaps, individual units have different $\boldsymbol{q}^\mu$ vectors. Here, for instance, $\boldsymbol{q}_1^\mu = (0, 1/2, 1/2, 0, 0, 0)^T$, $\boldsymbol{q}_2^\mu = (1/3, 0, 1/3, 0, 1/3, 0)^T$, $\boldsymbol{q}_3^\mu = (1/3, 1/3, 0, 1/3, 0, 0)^T$, $\boldsymbol{q}_4^\mu = (0, 0, 1, 0, 0, 0)^T$, $\boldsymbol{q}_5^\mu = (0, 1/2, 0, 0, 0, 1/2)^T$, $\boldsymbol{q}_6^\mu = (0, 0, 0, 0, 1, 0)^T$. Similarly, in (b), the elements m_{hj}^μ of the matrix of correlation lengths can vary considerably, for a given unit, or take unit values when the gains, β, of the mediating oscillators (not shown) are large.

The operation of the model neuron can be visualized by referring to figure (2.3a) and equation (2.4), from which s_h^μ can be interpreted as a positive feedback (for self–amplification) and the STM, r_h^μ, represents the negative feedback for stabilization (or synchronization). Thus, the interplay between recurrent inhibition and recurrent excitation (on presentation of a new example) allows aggregates of hidden activities to coalesce, disintegrate or fluctuate in size. Such a *re–wiring* scheme may not be far from that postulated by Poggio etc., as a basis for implementing the movable centres in a network with Gaussian units [6]. It can be seen, from figure 2.3a, that the output y_h^μ is linear up to the point where it coincides with the filter envelope $\Omega_h(s_h, s_j)$. This point is marked by the instantaneous STM, which therefore sets the diameter of the region across which the neuron will stay linear, (in response to any noticeable changes in $\boldsymbol{w}_h$). The output y_h^μ contracts, from both sides of r_h^μ, as $\boldsymbol{w}_h$ increases in magnitude, (effectively pushing the neuron into extreme specialization). On the other hand, small magnitudes of $\boldsymbol{w}_h$ lead to a dilation of y_h^μ, from both sides of r_h^μ, and hence exposing h to a wider range of the input features (represented in other hidden activities).

Minor changes in r_h^μ have the effect of altering the height of the linear region of the neuron output, and also its orientation with respect to other feature inputs. As shown in figure 2.3b, the activity of unit h, previously oriented at an angle $\theta_h \approx \tan^{-1}(s_h^\mu / r_h^\mu)$, (w.r.t. other activities) for a given input vector, can be steered to a new angle θ'_h by a shift in its STM to $r_h^{\mu'}$, relating to a new input vector. The larger the value of r_h^μ the wider is the receptive field of the unit. Therefore, overlaps between the receptive fields of neighbouring units may result in a continuum of receptive field properties. The shape of y_h^μ reveals that units for which $r_h^\mu > s_h^\mu$ will incure a higher cost than those for which

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 285-290

$r_h^\mu \leq s_h^\mu$. This phenomenon also has the desirable effect that a single unit is unlikely to dominate the wining process at the expense of other units.

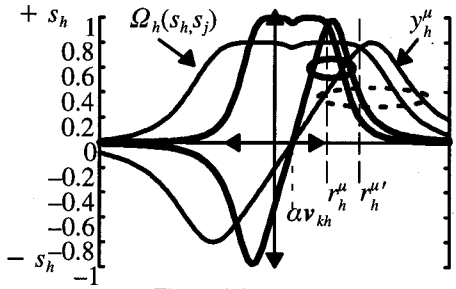

Figure 2.3a The density distribution of the CBN model neuron.

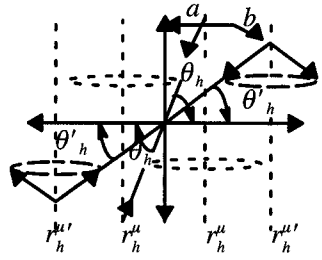

Figure 2.3b Effects of varying the parameters governing a CBN model neuron.

When the oscillator gain, β, is a large, the activity patterns to which h and j belong form an orthonormal set when:

$$m_{hj} = \begin{cases} 1 & \text{if } h = j \\ 0 & \text{if } h \neq j \end{cases} \tag{2.5}$$

Since the activity of a hidden unit is a "number representation" of the importance of the corresponding input vector, units with small responses (irrelevant features) are *forced* (through lateral inhibition) to relinguish their roles to those with stronger responses.

3 Learning in a CBN

The mapping performed by the CBN neuron can be quantified by taking advantage of the consequences of optimizing the system of equations in (2.4), one system per neuron, rather than having to accept an objective function that is explicitly (or implicitly) given by a particular self–organizing algorithm [7]. The CBN is therefore trained by converting the STMs into the LTM vectors $\boldsymbol{w}_h$, where (from equation (2.4))

$$\frac{\partial J_h^\mu(h,j)}{\partial w_{hi}} = \frac{\partial J_h^\mu(h,j)}{\partial r_h^\mu}\frac{\partial r_h^\mu}{\partial w_{hi}} = \eta_i^\mu x_i^\mu y_h^\mu \left(s_h^\mu - r_h^\mu\right)^{(p-1)} \tag{3.1}$$

The update rule is Hebbian, with a forgetting (STM) term. The parameter η_i^μ is a learning rate that varies with time. Note that the appearance, in equation (3.1), of the filter output, y_h^μ, implies that individual *weights are continuously normalized*, during the learning process, in addition to the forgetting term.

In the experiments reported in the next section, the output weights were trained through a minimization of the following cost function:

$$E_\mu = \frac{1}{2N}\sum_\mu \left(d_k^\mu - z_k^\mu\right)^2 \tag{3.2}$$

where N is the number of training patterns,

$$z_k^\mu = \psi\left(\sum_h v_{kh} y_h^\mu + v_{k0}\right), \tag{3.3}$$

v_{k0} is the threshold of unit k and ψ is a linear function.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 285-290

4. Experiments and Results

In this paper, the idea is not to compare network architectures and/or algorithms, but rather, to demonstrate the feasibility of performing computations in real time with little computational complexity. Two experiments were carried out in this light, with $p = 4$ (see equation (3.1)). The first involved the use of the computer generated time series data set (one of the data sets used in the Santa Fe Time Series Competition). The following brief description has been summarized from [8], where further details may be obtained. This particular univariate time series data set was generated (100, 000 points in total) by numerically integrating the equations of motion for a damped, driven particle

$$\frac{d^2x}{dt^2} + \gamma\frac{dx}{dt} + \Delta V(x) = F(t) \qquad (4.1)$$

in an asymmetric four-dimensional four-well potential

$$V = a_4(x_1^2 + x_2^2 + x_3^2 + x_4^2)^{\frac{1}{2}} - a_2(x_1^2x_2^2) - a_1x_1 \qquad (4.2)$$

With a forcing period $F(t) = F\sin(\omega t)$ in the x_3 direction, and a dissipation of $-\gamma$ velocity. The value of a_1 has a small drift produced by integrating a Gaussian random variable and the observable saved is

$$\{(x_1^2 + 0.3)^2 + (x_2^2 + 0.3)^2 + x_3^2 + x_4^2\}^{\frac{1}{2}} \qquad (4.3)$$

The data used in this experiment is labeled D1 and D2 in the database [8]. Of the 16666 patterns that were generated from the D1 set, only 5000 were used for training. Figures 4.1 and 4.2 show the test results and the training error distribution respectively.

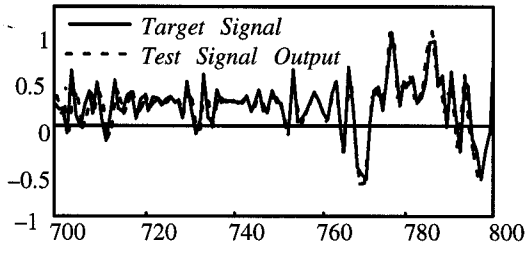

Figure 4.1 A plot of the target and network outputs for the time series test set.

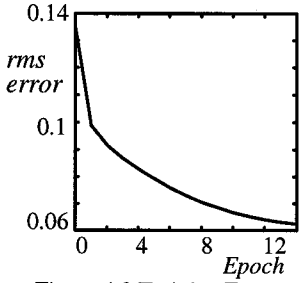

Figure 4.2 Training Error distribution

The second experiment involved the problem of link admission control (LAC) in the asynchronous transfer mode (ATM) telecommunication network [9].

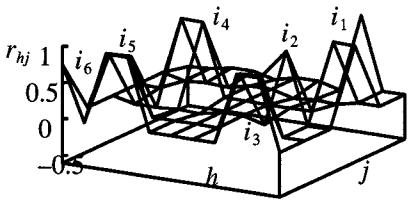

Figure 4.3 All 6 inputs inputs in the LAC problem coded in the STMs of the network

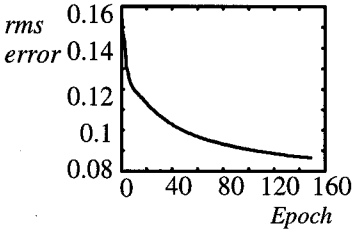

Figure 4.4 Training Error Curve

A CBN of 6 input units, 9 hidden units and 1 output unit was trained as a function approximator. Figure 4.3 shows the distribution of the network's STM of all six inputs.

The rms error over 150 presentations of the entire set the 500 training patterns is also shown in figure 4.4. The hit rate (or number of correct decisions) for this experiment was 78%. This is lower than that which is usually obtained with the conventional MLP [9]. With 12 hidden units, the CBN increases its hit rate to approximately 84%.

5. Conclusions

The suitability of a correlation–based network for real–time processing has been demonstrated. Lateral interactions between the hidden layer units serve as a means of preprocessing the input to the network before it reaches the output layer that is only capable of forming linear discriminant functions. The model neuron used in the hidden layer ensures that only important features of the input are responded to. Thus, as a unit learns it becomes more specialized to only a particular subregion of the input space. The postsynaptic predictors are linear and so, it can be stated that the subspace spanned by the active units will correspond to the space of the principal component eigenvectors of the input. The simulation results on two real–world data also confirm the feasibility of the approach.

Acknowledgements

This work was financed by NUTEK, the Swedish National Board for Industrial and Technical Development.

References

[1] J. Ngole: Correlation–Based Networks for Implementation in CMOS Analogue VLSI, PhD Thesis, Newcastle Uni., Uk, (1995).

[2] J. J. Hopfield: Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proceedings of the National Academy of Science USA, Vol. 79, pp. 2554–2558, (1982).

[3] S., Haykin: Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Comapany, New York, (1994).

[4] H. B., Barlow: Conditions for Versatile Learning, Helmholtz's unconcious ineference, and the task of perception. Vision Research, 30:1561–1571, (1990).

[5] G. F., Harpur and R. W., Prager, "Experiments with simple Hebbian–based learning rules in pattern classification tasks", CUED/F–INFENG/TR 168, Cambridge University Engineering Dept., England, (1994).

[6] T., Poggio and F., Girosi, "Networks for approximation and Learning", Proc. of the IEEE, Vol. 78, No. 9, 1990.

[7] G. J., Goodhill S., Finch and T., Sejnowski 1995. "Optimizing cortical mappings." To appear in Advances in Neural Information Processing Systems, 8, eds. David S. Touretzky, Michael Mozer & Michael E. Hasselmo, MIT Press: Cambridge, MA.

[8] A., Weigend, CU Time series Resipository: 1995. http://www.cs.colorado.edu/~andreas/T ime–Series/TSWelcome.html

[9] H. Brandt et. al.: A Hybrid Neural Network Approach to ATM Admission Control, Proc. of the Int. Switching Symposium (ISS'95), p. b6, Berlin, April 1995.