ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 319-326

Trends in Unsupervised Learning

Colin Fyfe,
Department of Computing and Information Systems,
The University of Paisley, Scotland.
email: fyfe0ci@paisley.ac.uk

Abstract. We review the trends in unsupervised learning towards the search for (in)dependence rather than (de)correlation, towards the use of global objective functions, towards a balancing of cooperation and competition and towards probabilistic, particularly Bayesian methods.

1. Introduction

Artificial Neural Networks (ANNs) whose parameters are adjusted using unsupervised learning are often motivated by either

1. a desire to understand biological information processing or
2. a desire to emulate the powerful properties of biological information processors

The simplest networks consist of a set of input vectors $\mathbf{x}$ and outputs $\mathbf{y}$ connected by a weight matrix, W, where w_{ij} connects x_j to y_i. Then the problem in unsupervised learning is to find the values of the parameters, W, which will best solve the current problem.

There have been two major paradigms [10, 26, 11]used in unsupervised networks:

Hebbian Learning This is usually implemented by $\Delta w_{ij} = \eta x_j y_i$, where Δw_{ij} is the change in weight w_{ij} and η is a learning rate. Now we can substitute y into the learning rule to get $\Delta w_{ij} = \eta \sum_k w_{ki} x_k x_j$. It is this feature - the interaction between the input variables - which gives Hebbian learning its power to respond to second order correlations in the data set.

Oja [19] has shown that the special form of decay of the form

$$\Delta w_i = \alpha(x_i y - y^2 w_i)$$

not only stops the weights from growing too large, it also causes convergence to the first Principal Component direction of the input data. Extensions of the basic rule have been shown to be capable of finding other Principal Components[20, 21, 22]

Competitive Learning One of the non-biological aspects of the basic Hebbian learning rule is that there is no limit to the amount of resources which may be given to a synapse. This is at odds with

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 319-326

real neural growth in that it is believed that there is a limit on the number and efficiency of synapses per neuron. In other words, there comes a point during learning in which if one synapse is to be strengthened, another must be weakened. This is usually modelled as a competition for resources, followed by a weight update of

$$\Delta w_{ij} = \eta(x_j - w_{ij}), \text{ for the winning neuron i}$$

Two major variants on this rule are Kohonen's Self Organising Feature Map [16] and Grossberg's ART models [3].

These two means of adjusting networks weights have been the most influential though some authors would insert additional paradigms such as dynamical models (such as the Hopfield network) and perhaps within this, stochastic models such as the Boltzmann Machine.

2. Correlation and Independence

Hebbian learning, while generally accepted as a valid model for biological learning, is not a particularly accurate model of Long Term Potentiation if it is implemented as in Section 1.. [18] note that two small repeated inputs has the same effect as a single large input. Perhaps more importantly, the incorporation of nonlinearity has allowed ANNs to move from those seeking correlations (second order statistics) to those using higher order statistics to search for independence. There have been two main streams of interest:

1. Some networks are designed to uncover the underlying causes of a data set e.g. [6, 23, 25, 13]. Such a data set is often described as having been generated by a small number of hidden causes. Experiments are usually performed on a data set created by mixing the outputs of a small number of sources where the mixing parameters are often integers. Of interest in this paper, is the fact that a simple extension to Oja's Subspace Network [4] in which we perform a rectification of negative weights enables a PCA network to identify the independent causes of visual scenes. [5] will discuss this problem in more detail.
2. Other networks are designed to extract a single signal from a (linear) mixture of signals [2, 14, 15]. This is often called the "cocktail party problem" since we can use these networks to extract a single voice from a mixture of voices. Again it has been shown [14, 9] that nonlinear extensions of the PCA algorithms allow us to identify the independent sources which have been linearly mixed. Solutions to this problem have constituted one of the major trends in neural computing during the last three years.

3. Objective Functions

Becker [1] has pointed out that one advantage of deriving learning rules from objective functions is that this allows networks to be understood in

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 319-326

terms of their global behaviour. The derivation of a learning rule usually begins with a global function which the researcher believes is important for artificial systems to have and which biological systems are believed to exhibit. For example, Barlow's view of the neuron as a "suspicious coincidence detector" might lead to a desire to maximise the correlation between two sets of data[17]. Consider two sets of input data, $\mathbf{x}_1$ and $\mathbf{x}_2$. Then in classical CCA, we attempt to find that linear combination of the variables which gives us maximum correlation between the combinations. Let

$$\begin{aligned} y_1 &= \mathbf{w}_1\mathbf{x}_1 = \sum_j w_{1j}x_{1j} \\ y_2 &= \mathbf{w}_2\mathbf{x}_2 = \sum_j w_{2j}x_{2j} \end{aligned}$$

We wish to maximise the correlation $E(y_1 y_2)$ where $E()$ denotes the expectation which will be taken over the joint distribution of $\mathbf{x}_1$ and $\mathbf{x}_2$. Typically in CCA, we add the constraint that $E(y_1^2 = 1)$ and similarly with y_2. Using the method of Lagrange multipliers, this yields the constrained optimisation functions,

$$\begin{aligned} J_1 &= E(y_1 y_2) + \frac{1}{2}\lambda_1(1 - y_1^2) \text{ and} \\ J_2 &= E(y_1 y_2) + \frac{1}{2}\lambda_2(1 - y_2^2) \end{aligned}$$

These can be optimised independently by implicitly assuming that $\mathbf{w}_1$ is constant when we are changing $\mathbf{w}_2$ and vice-versa. We wish to find the optimal solution using gradient ascent and so we find the derivative of the instantaneous version of each of these functions with respect to both the weights, $\mathbf{w}_1$ and $\mathbf{w}_2$, and the Lagrange multipliers, λ_1 and λ_2. These yield respectively

$$\begin{aligned} \Delta w_{1j} \propto \frac{\partial J_1}{\partial \mathbf{w}_1} &= \mathbf{x}_1 y_2 - \lambda_1 y_1 \mathbf{x}_1 = \mathbf{x}_1(y_2 - \lambda_1 y_1) \\ \Delta\lambda_1 \propto \frac{\partial J_1}{\partial \lambda_1} &\propto (1 - y_1^2) \end{aligned}$$

which has been shown to find suspicious coincidences in data.

4. Cooperation and Competition

Now while cooperation and competition are individually powerful, there have been increasing attempts to use both simultaneously in ANNs. For example, many methods designed to find the independent causes of a data set use methods which contain a tension between cooperation and competition. For example, Foldiak [7] has a model in which each neuron tries to keep its probability of firing down by adjusting its own threshold. The mechanism does have some biogical plausibility in that neurons do

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 319-326

become habituated to inputs and stop responding so strongly to repeated sets of inputs.

$$y_i = f(\sum_{j=1}^{n} q_{ij}x_j + \sum_{j=1}^{m} w_{ij}y_j - t_i) \quad (1)$$

where q_{ij} is the weight of the feedforward connection from the j^{th} input x_j, w_{ij} is the weight of the lateral connection from the j^{th} output neuron to the i^{th} in that layer and t_i is the adjustable threshold for the i^{th} output neuron. Both sets of weights and the threshold are adjustable by competitive type learning:

$$\begin{aligned} \Delta w_{ij} &= \begin{cases} -\alpha(y_i y_j - p^2) \text{ if } i \neq j \\ 0 \text{ if } i = j \text{ or } w_{ij} < 0 \end{cases} \\ \Delta q_{ij} &= \beta y_i (x_j - q_{ij}) \\ \Delta t_i &= \gamma(y_i - p) \end{aligned}$$

where α, β, γ are adjustable learning rates. The feedforward weights, q_{ij}, use simple competitive learning. The lateral learning rule for the w weights will stabilise when $E(y_i y_j) = p^2$. i.e. each pair of units will tend to fire together a fixed proportion of the time. This rule will interact with the rule which changes the threshold: the long term effect of this rule should be to set the threshold t_i to a set value to ensure $E(y_i) = p$. By choosing the value of p appropriately we can determine the level of sparseness of the firing pattern.

[24, 23, 6, 13] all have models for tackling this particular problem. It is of interest that in each of these models there is a balancing of competing criteria: e.g. cooperation between outputs (finding several causes per input) is balanced with competition (separation of responsibilities for coding different independent sources). This seems to be an essential part of each solution but we still seem to be lacking an overall rationale for this observation.

5. Probabilistic Methods

Probabilistic methods are usually based on Bayes Theorem. Many probabilistic methods too have this tension between two competing criteria which we met in the last section; they also often have explicit objective functions as seen in Section 3.. Many of the models contain both

- a recognition model which takes e.g. an image as input and attempts to infer the underlying causes of the image using bottom-up connections
- a generative model which has top down connections from underlying reasons for the input image i.e. the top down connections create the image from an abstraction of the image.

[12] states that "Visual perception consists of inferring the underlying state of the stochastic graphics model using the false but useful assumption that the observed sensory input was generated by the model."

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 319-326

Learning is done by maximising the likelihood that the observed data came from the generative model. The simplest probabilistic model is probably the Mixtures of Gaussians in which

- Each data point has an associated probability of being generated by a mixure of Gaussian distributions.
- Given the current parameters of the model, we calculate the probability (*the posterior probability*)that any data point came from the distributions.
- The learing process adjusts the parameters of the model - the means, variances and mixing proportions (weights) of the Gaussians - to maximise the likelihood that the model produced the points.

In order to generate a data point,

- Pick a hidden neuron (the underlying cause of the data). Give it a state of 1, set all other hidden neurons' states to 0.
- Each hidden neuron will have probability of being picked of π_j - *a prior probability.*
- Feed back to input weights through weight vector $\mathbf{g}_j$. The $\mathbf{g}_j$ is the center of the Gaussian $= \{g_{j1}, g_{j2}, ..., g_{jn}\}$.
- Add local independent zero mean Gaussian noise to each input.
- This means that each data point is a Gaussian cloud with mean $\mathbf{g}_j$ and variance σ_i^2.

$$p(\mathbf{d}) = \sum_j \pi_j \prod_i \frac{1}{\sqrt{2\pi}\sigma_i} e^{-(d_i - g_{ji})^2/2\sigma_i^2} \tag{2}$$

We may then hope to interpret the data. (This is the Expectation Step of the EM algorithm).

1. Compute the probability density for each data point (assuming the model is correct.)

$$p(\mathbf{d}|s_j = 1) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_i} e^{-(d_i - g_{ji})^2/2\sigma_i^2} \tag{3}$$

2. Weight these with the prior probabilities π_j.
3. Use Bayes theorem to calculate the probability of the data.

$$p(s_j = 1|\mathbf{d}) = \frac{\pi_j p(\mathbf{d}|s_j = 1)}{\sum_k \pi_k p(\mathbf{d}|s_k = 1)} \tag{4}$$

We have now calculated the posterior probabilities of the hidden states given the data ("perceptual inference") - the E-step. We may treat learning as Expectation Maximisation:

$$\begin{aligned}
\mathbf{g}_j &= \frac{E\{p(s_j = 1|\mathbf{d})\mathbf{d}\}}{E\{p(s_j = 1|\mathbf{d})\}} \\
\sigma_i^2 &= \frac{E\{p(s_j = 1|\mathbf{d})(d_i - g_{ji})^2\}}{E\{p(s_j = 1|\mathbf{d})\}} \\
\pi_j &= E\{p(s_j = 1|\mathbf{d})\}
\end{aligned}$$

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 319-326

We have now a means of maximising the expectation - the M-step which may be done using online learning.

$$\Delta g_{ji} = \epsilon p(s_j = 1|\mathbf{d})(d_i - g_{ji}) \tag{5}$$

We may view competive learning models as methods of fitting Gaussian parameters. But " They are usually inefficient because they do not use a full M-step and slightly wrong because they pick a single winner among the hidden units instead of making the states proportional to the posterior probabilities."

6. Conclusion

The field of unsupervised learning is a wide one which cannot be covered in a brief paper. However we have discussed some of the recent trends in this area, though not mentioned others: for example, we are now increasingly insisting that real world problems are tackled. We also recognise the need for multilayered networks for solving higher order problems. Finally we have discussed these trends as though they were separate which is not the case: in many new networks we are seeing an intertwining of the above trends to give an even richer set of analysable networks.

References

[1] S. Becker. Mutual information maximization: Models of cortical self-orgainization. *Network:Computation in Neural Systems*, 7:7–31, 1996.

[2] A. J. Bell and T. J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[3] Gail Carpenter and Stephen Grossberg. *Pattern Recognition by Self-Organizing Neural Networks*. The MIT Press, 1991.

[4] D. Charles and C. Fyfe. Modelling multiple cause structure using rectification constraints. *Network: Computation in Neural Systems*, 1998.

[5] D. Charles and C. Fyfe. Noise to extract independent causes. In *European Symposium on Artificial Neural Networks*, April 1999.

[6] P. Dayan and R. S. Zemel. Competition and multiple cause models. *Neural Computation*, 7:565–579, 1995.

[7] P. Földiák. *Models of Sensory Coding*. PhD thesis, University of Cambridge, 1992.

[8] M. Girolami and C. Fyfe. Stochastic ica contrast maximisation using oja's nonlinear pca algorithm. *International Journal of Neural Systems*, 1999.

[9] Simon Haykin. *Neural Networks- A Comprehensive Foundation*. Macmillan, 1994.

[10] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing, 1992.

[11] G. E. Hinton and Z. Ghahramani. Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society, B*, 1997.

[12] G.E. Hinton, P. Dayan, and M. Revow. Modelling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, 8(1):65–74, Jan 1997.

[13] C. Jutten and J. Herault. Blind separation of sources,part 1: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[14] J. Karhunen, E. Oja, L. Wang, R. Vigário, and J. Joutsensalo. A class of neural networks for independent component analysis. *IEEE Transactions on Neural Networks*, 1997. (in press).

[15] Tuevo Kohonen. *Self-Organising Maps*. Springer, 1995.

[16] P. L. Lai and C. Fyfe. Canonical correlation analysis using artificial neural networks. In *European Symposium on Artificial Neural Networks, ESANN98*, 1998.

[17] C. W. Lee and B. A. Oslhausen. A nonlinear hebbian network that learns to detect disparity in random-dot stereograms. *Neural Computation*, 8:545–566, 1996.

[18] E. Oja. A simplified neuron model as a principal component analyser. *Journal of Mathematical Biology*, 16:267–273, 1982.

[19] E. Oja. Neural networks, principal components and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[20] E. Oja, H. Ogawa, and J. Wangviwattana. Principal component analysis by homogeneous neural networks, part 1: The weighted subspace criterion. *IEICE Trans. Inf. & Syst.*, E75-D:366–375, May 1992.

[21] T.D. Sanger. Analysis of the two-dimensional receptive fields learned by the generalized hebbian algorithm in response to random input. *Biological Cybernetics*, 1990.

[22] E. Saund. A multiple cause mixture model for unsupervised learning. *Neural Computation*, 7:51–71, 1995.

[23] Jurgen Schmidhuber and Daniel Prelinger. Discovering predictable classifications. Technical Report CU-CS-626-92, University of Colorado, 1992.

[24] R. H. White. Competitive hebbian learning: Algorithm and demonstration. *Neural Networks*, 5:261–275, 1992.

[25] Jacek M. Zurada. *Introduction to Artificial Neural Systems*. West Publishing Company, 1992.