ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 215-220

Invariant Feature Maps for Analysis of Orientations in Image Data

Stephen McGlinchey and Colin Fyfe

Department of Computing and Information Systems,
University of Paisley,
High Street, Paisley PA1 2BE, Scotland
Tel: +44 141 848 3328, +44 141 848 3305,
fax: +44 141 848 3542.
Email: mcgl-ci0@wpmail.paisley.ac.uk,
fyfe-ci0@wpmail.paisley.ac.uk

Abstract

We present a method that uses competitive learning and a neighbourhood function in a similar way to the self-organising map (SOM) [3]. The network consists of a number of modules that are positioned in an array (normally in one or two dimensions) where each module performs a subspace projection and the rotation of these subspaces is weighted by the neighbourhood function. Non-linear activation functions are introduced so that each node performs non-linear PCA on the training data captured in its Voronoi region. We show that this network may be used for position invariant detection of bars at varying orientations.

1. Introduction

Recognition of patterns subject to transformations such as translation, rotation and scaling has been a difficult problem in artificial perception. The projection of objects on the retina is variable in position, size, orientation, luminance etc., yet we are still able to recognise objects with ease, regardless of such variations. A solution to this problem has been proposed by Kohonen et al. [3,4] called the Adaptive Subspace Self-Organising Map (ASSOM). The motivation for the ASSOM is that if different samples can be derived from each other by means of a linear transformation, then they span a common linear subspace of the input space and projecting them onto this subspace can then filter out their differences. Using common subspaces as filters also motivates the method presented in this paper.

By constructing a SOM where each module defines a subspace, we can observe organisation of clusters of different subspaces that each provides a different invariant filter. The neighbourhood relations in the map cause neighbouring modules to represent similar subspaces. The map consists of an array of modules, normally in one or two dimensions, although higher dimensional maps may be used. Each module of the map consists of the same number of nodes which defines the dimensionality of the subspace represented

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 215-220

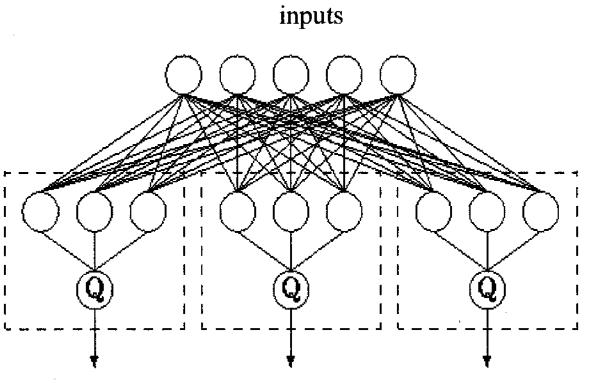


Figure 1: The modules of the subspace map. Each module is enclosed in a dotted box and has a group of neurons and another single output which is a quadratic function of the other neurons.

by the module and a single output which is a quadratic function of the outputs of these nodes. This is illustrated in figure 1.

Hubel and Wiesel [2] have shown that in the visual cortex, neurons whose receptive fields have similar orientation preferences are grouped together in columns which exhibit sequence regularity i.e. neighbouring columns respond to similar orientations and a systematic rotation is observed over the cortical surface. Other topographic mappings exist where the position of objects in visual space is corresponds with systematic firing of neurons on the 2D surface of the cortex. These findings have inspired the use of a topology preserving artificial neural network to detect the orientation of bars within images.

2 Training Algorithm

The projection of an input vector, $\mathbf{x}$, onto a subspace $L^{(i)}$ gives a projection $\hat{\mathbf{x}}^{(i)}$ and a residual $\tilde{\mathbf{x}}^{(i)}$. An episode S consists of a set of consecutive time instants and the inputs $\mathbf{x}(t_p)$ are taken from these sampling instants $t_p \in S$. These vectors are all projected onto the subspaces represented by the weight vectors of the modules. For each episode, we choose a representative winning node, c where the winner is constant for the duration of the episode. Since each node of the map represents a subspace $L^{(i)}$ of the input space, we can select the winning module, c, based on how closely its subspace, $L^{(c)}$, approximates the data in the episode. The module that is selected as the representative winner has the greatest energy of the orthogonal projections of the input vectors (1).

$$c_r = \arg\max_i \left\{ \sum_{t_p \in S} \|\hat{\mathbf{x}}^{(i)}(t_p)\|^2 \right\} \tag{1}$$

In the ASSOM, the basis vectors, $b_h^{(i)}$ of unit i are rotated for each sample vector $\mathbf{x}(t_p)$ of the episode S where $\alpha(t)$ is the leaning rate parameter at time t and the units $i \in N^{(c_r)}$, where $N^{(c_r)}$ is the set of modules in the neighbourhood of the representative winner (equation 2).

$$b_h'^{(i)} = \prod_{t_p \in S} \left[I + \alpha(t_p) \frac{\mathbf{x}(t_p)\mathbf{x}^T(t_p)}{\|\hat{\mathbf{x}}^{(i)}(t_p)\| \; \|\mathbf{x}(t_p)\|} \right] b_h^{(i)}(t_p) \tag{2}$$

One of the major differences between the ASSOM and the method that we now present is that this rule is replaced with a weight update method based on Oja's subspace algorithm (equation 3,4) [8]. The j^{th} output of $L^{(i)}$ is denoted by $y_j^{(i)}$ and is calculated by a weighted sum of the inputs.

$$y_j^{(i)} = \sum_{k=1}^{N} w_{jk}^{(i)} x_k \tag{3}$$

$$\Delta w_{jk}^{(i)} = \alpha \left(x_k y_j^{(i)} - y_j^{(i)} \sum_l w_{kl}^{(i)} y_l^{(i)} \right) \tag{4}$$

Therefore, for every module, $i \in N^{(c_r)}$, the weights are updated to shift towards a principal subspace of $\mathbf{x}(t_p \in S)$. This shift most strongly affects the winning module and the neighbourhood function determines the effect on its neighbouring modules. The neighbourhood function and competitive learning ensure that each subspace is trained on a subset of the training data and therefore, the weight vectors of each module form a principal subspace of a different class of input patterns. We have previously shown [7] that Oja's subspace algorithm is equivalent to a negative feedback network and this has already been used in a SOM architecture [5,6].

The neighbourhood function $h_{ci}(t)$ is a Gaussian or difference of Gaussians (Mexican hat) with a width that is decreased in time. The threshold of this function may also be narrowed so that the number of modules in $N^{(c_r)}$ is reduced.

The basis vectors of each node in the ASSOM are kept orthonormal by periodically resetting them using the Gram-Schmidt method, however, Oja's subspace algorithm guarantees that the basis vectors within each module will be orthonormal and therefore, there is no need to re-orthonormalise the weights.

3. Non-Linear Activation Functions

When the ASSOM was used for filtering of frequencies from speech data, it was necessary to dissipate small weights to prevent the wavelet type filters becoming tuned to multiple frequencies. This was performed by pruning weights that are less than ε to zero, where $\varepsilon (0 < \varepsilon < 1)$ is typically a very small term.[3,4]

A similar effect can be achieved by using the activation function shown in equation (6). Positive values of the κ term cause the function to be similar to the solid line in figure 2 and negative values shift this towards the shape of the dotted line. A negative κ value provides a function similar to the small weight dissipation used by Kohonen et al.

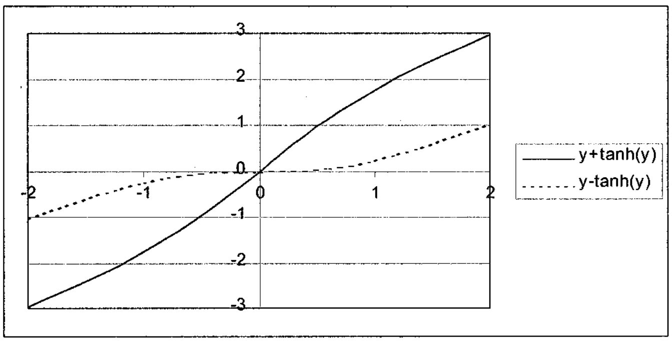


Figure 2: The non-linear functions.

$$f(y) = y + \kappa \tanh(y) \qquad (6)$$

This function was originally was proposed in [1] as a means to provide a more generalised independent component analysis transformation.

The weight update now gives a non-linear principal component analysis of the input space (equation 7.)

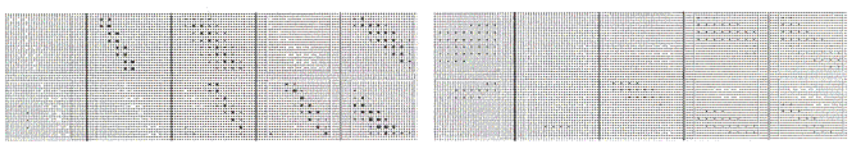
$$\Delta w_{jk}^{(i)} = \alpha \left(x_k f(y_j)^{(i)} - f(y)_j^{(i)} \sum_l w_{kl}^{(i)} f(y_l)^{(i)} \right) \qquad (7)$$

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 215-220

4. Application to Bars in Artificial Image Data

The network described in the previous sections was trained on artificial image data where bars of varying orientations are presented in different positions on an input grid (8x8 pixels). Eight different bar orientations were presented and each bar was in eight different positions. For example, for a vertical line, the line appeared in all eight columns. In an episode of input vectors, the orientation of the line was kept constant and the position was varied.

When the network was trained on this data with a zero κ value (linear network), each node showed wide bars of strong weights, and a systematic ordering of angles over the whole map (figure 3a). Multiple bar positions were also learned, indicating a similar finding to that of Kohonen where the wavelet type filters became tuned to multiple frequencies. Increasing κ to a positive value widened these bars so that they became distributed across most of the weights and there was a smearing over the whole vector (figure 3b.)



(a) (b)

Figure 3: Weight vectors from a subspace map trained with (a) $\kappa = 0$ and (b) $\kappa = 0.8$

A negative value of κ had the effect of reducing the width of the bars learned until there were thin lines of strong weights and all other weights were small or zero. When negative values close to -1 were used, individual bars were learned, but the topographic ordering was compromised. Since Oja's subspace algorithm guarantees orthonormal weights vectors, when small weights are reduced, larger weights grow and this effect can become stronger than the effect of the neighbourhood function. Therefore, it is a good strategy to choose a high initial κ value and then reduce it during training. Figure 4 shows the weights of a map where the κ parameter was set to 0.2 and then reduced by 0.05 every 50 training episodes to a minimum value of -1.

5. Conclusions

We have described a subspace network that uses competitive learning and neighbourhood relations to train each module on a different subset of the training data where smooth ordering of subspaces is a result of the self-organising process. This method has been compared with the ASSOM which also gives a smooth ordering of subspaces using a different training algorithm.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 215-220

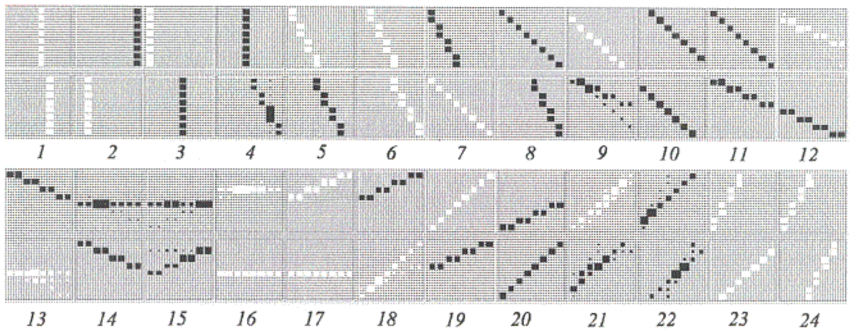


Figure 4: The weight vectors of a converged map showing a systematic ordering of angles.

The use of non-linear activation functions in a subspace self-organising map has been demonstrated to show that each node of the map converges to give a non-linear principal component analysis of the data close to its Voronoi region. This has been effective in extracting the direction of a bar from a sequence of frames of artificial data that all have a bar with the same orientation, but in different positions. Future work will concentrate on applying this method to real data.

References

[1] Girolami, M and Fyfe, C. "*Extraction of Independent Signal Sources using a Deflationary Exploratory Projection Pursuit Network with Lateral Inhibition*" I.E.E Proceedings on Vision, Image and Signal Processing , Vol 14, No 5, pp 299 - 306, 1997.

[2] Hubel D.G., Wiesel T.N. *"Sequence Regularity and Geometry of Orientation Columns in the Monkey StirateCortex"* J.Comp. Neurol. 158:267-293

[3] Kohonen T., "*Self-Organizing Maps*" 2nd Edition Springer 1997.

[4] Kohonen T., Kaski S., Lappalainen H., *"Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM"* Neural Computation 9,1321-1344, 1997.

[5] McGlinchey S., Fyfe C., *"Automatic Extraction of Phase and Frequency Information from Raw Speech Data"* International Conference on Artificial Neural Networks and Genetic Algorithms, Springer, April 1997.

[6] McGlinchey S., Fyfe C., *"An Angular QuantisingSelf-Organising Map for Scale Invariant Classification"* Workshop on Self-Organizing Maps, June 1997.

[7] McGlinchey S., Fyfe C., *"Local Models for Coding"* ISCS Symposium on Soft Computing, September 1997.

[8] Oja E., *"Neural Networks, Principal Components and Subspaces"* International Journal of Neural Systems, 1:61-68, 1989