Polar Backpropagation

Keith R. L. Godfrey and Yiannis Attikiouzel

Department of Electrical and Electronic Engineering
The University of Western Australia
Nedlands 6009 Australia

Abstract

This paper defines the general polar classification problem and proposes a modified backpropagation architecture for solving it. The topology of interconnections is constrained so that selected ones correspond with physical entities in the problem. With this in-built heuristic knowledge, a polar backpropagation network offers a more informative solution than a standard backpropagation network.

1 Introduction

Backpropagation has long been established as one of the more effective artificial neural network (ANN) paradigms [3]. It is conceptually easy to understand and has been shown to solve a wide variety of problems. Theorems exist to prove its ability to map arbitrary continuous functions [2, 5]. Backpropagation networks have found applications from ECG classification to bank mortgage processing. The present work arises from the classification of colours.

A class of problems has been found where the desired classification regions intersect at a central point called a *pole*. An example in two dimensions arises from plotting surface colours in the chromacity chart defined by the Commission Internationale d'Éclairage [1, 4, 6]. Points in this diagram represent different colours at the same brightness. The centre of the chart is achromatic (grey or white) and there is a smooth transition towards highly saturated colours at the boundary. Since the light reflected from an object or surface comprises a specular component as well as a coloured component, it follows that the loci of colours reflected from a surface will lie on a line radiating from the chromacity of the light source. If the chromacity of the light source is known, it can be plotted on the chromacity diagram and the calculation of saturation and hue becomes a polar transformation from that origin. A polar transformation from any other origin will not produce appropriate measures of hue and saturation.

The colour space is two-dimensional, but the general polar classification problem (PCP) can be defined as follows:

- The input space is N-dimensional.
- The input is defined in rectangular coordinates.
- There are M regions of interest in the space, corresponding to different input classes.
- The M regions converge at a central point called the *pole* and have classification boundaries which are approximately radial from the pole. This means that the distance from the pole has little or no effect on a point's classification. In two dimensions, the classification regions will be convex; in higher dimensions they need not be.
- If a polar transformation is made at the pole, the dimension of radial distance is redundant and the classification problem reduces to $N-1$ dimensions.
- The location of the pole $(p_1, \ldots, p_N)$ is not known *a priori*.

An artificial example of the polar classification problem in two dimensions is shown in Figure 1. A standard backpropagation network *can* find an approximation to the M classes in the N-dimensional space, but *cannot* locate the pole. The advantage of locating the pole is that it reduces the number of

dimensions in the classification problem and enables a more informative solution. Heuristic knowledge of the problem, i.e. existence of the pole, is useful in finding the solution.

The purpose of this paper is to present a modified backpropagation architecture which can solve the polar classification problem *and* locate the pole at the same time. The algorithm is called *polar backpropagation* (PBPN) in light of the problem that it solves. The 2-layer polar backpropagation network proposed in this paper is a special case of the general 3-layer backpropagation network [7, 9]. The advantage in using this network is that certain interconnection weights correspond to a physical quantity in the PCP, namely the coordinates of the pole. Heuristic knowledge of the problem is imposed in the constraints of the network.

2 Polar Backpropagation

The standard backpropagation algorithms are defined in [7] and [9]. The notation used here will be x_i for the input layer, z_j for the hidden layer and y_k for the output layer, where $1 \leq i \leq N$, $1 \leq j \leq H$ and $1 \leq k \leq M$. The weights from input to hidden layers are denoted w_{ji} and those from hidden to output layers are denoted u_{kj}. The polar backpropagation algorithm is developed in two steps:

1. By shifting the pole to the origin, the PCP reduces to a problem of boundaries passing through the origin. The shift (translation) of the pole is defined by:

$$x\prime_i = x_i - p_i \qquad (1 \leq i \leq N) \tag{1}$$

   where x_i is the original coordinate, $x\prime_i$ is the new coordinate and p_i is the coordinate of the pole. Since $p_1, \ldots, p_N$ are not known *a priori*, they are treated like interconnection weights, and optimised (learned) along with all the other weights.

2. The boundary planes created at the hidden layer are forced through the new origin by setting their threshold offsets w_{j0} to zero. This has the effect of making net_j linear rather than affine. The processing equation at the hidden layer is therefore:

$$net_j = \sum_{i=1}^{N} (w_{ji}.x\prime_i) \underbrace{+w_{j0}}_{0} \qquad (1 \leq j \leq H) \tag{2}$$

The activation function used in the hidden layer is the sigmoid:

$$z_j = f_s(net_j) \qquad \text{where} \qquad f_s(\gamma) = \frac{1}{1 + e^{-\gamma}} \tag{3}$$

2.1 Architecture

It is desirable to follow the standard backpropagation architecture as far as possible, so that the 'general delta rule' learning algorithm can be applied. The architecture proposed for 2-layer polar backpropagation is a special case of the 3-layer backpropagation architecture.

The network comprises five layers:

- an *input layer* of size N (standard)
- a *polar layer* of size N (with restricted input connections, no squashing function and no input weights except threshold offsets)
- a *hidden processing layer* of size H (standard processing equations but no threshold offsets)
- an *output processing layer* of size M (standard)
- a *training layer* of size M (standard)

The polar backpropagation network is shown in Figure 2. Weights w_{ji} and u_{kj} are connected in a fully feed-forward manner. There are threshold offsets u_{k0} in the output layer but not in the hidden layer. The threshold offsets p_i in the polar layer are the coordinates of the pole, with a change of sign effected by a constant input of -1 rather than 1. The polar layer behaves like a processing layer with restrictions. No squashing function is used, and the inputs are one-to-one connected from the input layer, rather than fully connected. Thus the number of elements in the polar layer must equal that of the input layer.

The advantages of the 2-layer polar backpropagation architecture over the 3-layer backpropagation architecture are:

1. The network has fewer interconnections, and therefore requires less computation.
2. The interconnection weights $p_1, \ldots, p_N$ have a physical meaning related to the problem, namely the location of the pole. The weights w_{ji} at the hidden layer contain angular information of the planes centred at the pole.

There is a larger number of weights in the standard 3-layer backpropagation network and they have no physical interpretation unless they happen to find the polar solution, which is very unlikely.

2.2 Processing Equations

The polar backpropagation processing equations are derived from equations 1 and 2 above:

$$\text{polar layer} \left\{ \quad x\prime_i = x_i - p_i \qquad (1 \le i \le N) \right. \tag{4}$$

$$\text{hidden layer} \left\{ \begin{array}{lll} net_j &=& \sum_{i=1}^{N} (w_{ji}.x\prime_i) \\ z_j &=& f_s(net_j) \qquad (1 \le j \le H) \end{array} \right. \tag{5}$$

$$\text{output layer} \left\{ \begin{array}{lll} net_k &=& \sum_{j=1}^{H} (u_{kj}.z_j) + u_{k0} \\ y_k &=& f_s(net_k) \qquad (1 \le k \le M) \end{array} \right. \tag{6}$$

2.3 The Training Algorithm

The standard backpropagation algorithm minimises an error function E with respect to the network weights u_{kj} and w_{ji} by steepest descent. The steepest descent principle is also applied here, except that there are the additional weights p_i. The function to be minimised is the mean squared error E for any particular training pattern:

$$E = \frac{1}{2} \sum_{k=1}^{M} (t_k - y_k)^2 \tag{7}$$

where t_k is the trained output at the k-th element. It can be shown that the partial derivatives of E with respect to the weights u_{kj}, u_{k0}, w_{ji} and p_i are:

$$\frac{\partial E}{\partial u_{kj}} = -(t_k - y_k).y_k.(1 - y_k).z_j \tag{8}$$

$$\frac{\partial E}{\partial u_{k0}} = -(t_k - y_k).y_k.(1 - y_k) \tag{9}$$

$$\frac{\partial E}{\partial w_{ji}} = -\left(\sum_{k=1}^{M} (t_k - y_k).y_k.(1 - y_k).u_{kj} \right).z_j.(1 - z_j).(x_i - p_i) \tag{10}$$

$$\frac{\partial E}{\partial p_i} = -\left(\sum_{j=1}^{H} \left(\sum_{k=1}^{M} (t_k - y_k).y_k.(1 - y_k).u_{kj} \right).z_j.(1 - z_j) \right).(-w_{ji}) \tag{11}$$

Using the steepest descent rules:

$$\Delta u_{kj} = -\alpha \frac{\partial E}{\partial u_{kj}} \qquad \Delta u_{k0} = -\alpha \frac{\partial E}{\partial u_{k0}} \qquad \Delta w_{ji} = -\alpha \frac{\partial E}{\partial w_{ji}} \qquad \Delta p_i = -\alpha \frac{\partial E}{\partial p_i} \tag{12}$$

the weight update law follows the 'delta rule':

$$\text{output layer} \begin{cases} \delta_k &= (t_k - y_k).y_k.(1 - y_k) \\ \Delta u_{kj} &= \alpha.\delta_k.z_j \\ \Delta u_{k0} &= \alpha.\delta_k \end{cases} \tag{13}$$

$$\text{hidden layer} \begin{cases} \delta_j &= z_j.(1 - z_j).\sum_{k=1}^{M}((t_k - y_k).u_{kj}) \\ \Delta w_{ji} &= \alpha.\delta_j.x\prime_i \end{cases} \tag{14}$$

$$\text{polar layer} \begin{cases} \delta_i &= \sum_{j=1}^{H}(\delta_j.(-w_{ji})) \\ \Delta p_i &= \alpha.\delta_i \end{cases} \tag{15}$$

3 Use of Polar Coordinates

For any processing element in the hidden layer, the weights $w_{j1}, \ldots, w_{jN}$ defining its boundary plane could be represented in polar coordinates. In two dimensions these would be:

$$g_j = \sqrt{w_{j1}^2 + w_{j2}^2} \tag{16}$$

$$\theta_j = \arctan(w_{j2}, w_{j1}) \tag{17}$$

The term g_j is equivalent to the steepest gradient of the plane net_j with respect to x_1 and x_2. The inverse of the polar transformation is:

$$w_{j1} = g_j \cos(\theta_j) \tag{18}$$

$$w_{j2} = g_j \sin(\theta_j) \tag{19}$$

Substituting these functions for w_{j1} and w_{j2} into the processing and training equations leads to a much more complicated formula:

$$\begin{aligned} net_j &= w_{j1}.x\prime_1 + w_{j2}.x\prime_2 \\ &= g_j.(x\prime_1.\cos(\theta_j) + x\prime_2.\sin(\theta_j)) \end{aligned} \tag{20}$$

By fixing g_j, the problem reduces to one dimension θ_j (or $N-1$ dimensions in the general case). The disadvantage is that the derivatives add a great computational burden and loss of generality of the delta rule. The derivative in the two-dimensional case is:

$$\frac{\partial net_j}{\partial \theta_j} = g_j.(x\prime_2.\cos(\theta_j) - x\prime_1.\sin(\theta_j)) \tag{21}$$

Since the purpose of polar coordinates is to describe the angular displacements of classification regions about the pole, it is only applicable at the ***solution*** of the PCP. It follows that there is no benefit (and the significant disadvantage described above) in using polar coordinates during training. Thus the proposed polar backpropagation algorithm uses rectangular coordinates. If polar coordinates are required, the transformation should be made *after* the solution has been found.

4 Results

Figure 1 is an example of an artificial PCP in two dimensions. Figures 3 and 4 show the solutions found by 2-layer backpropagation and 2-layer polar backpropagation networks, while the solutions are developing. Both networks have 6 processing elements in the hidden layer and learning rates of $\alpha = 0.1$, deliberately chosen to be low [10]. The plots are shown for 50, 100, 200 and 900 thousand iterations of training points picked at random from inside the square. It can be seen that the standard backpropagation network learns faster than the polar backpropagation network, but they both achieve a similar solution. The fundamental difference between the two solutions is that the polar network has determined the pole's location.

5 Conclusion

A polar backpropagation network has been developed for solving the polar classification problem. The architecture is a contrained version of general 3-layer backpropagation, such that selected weights correspond to physical entities in the problem, namely the coordinates of the pole. Heuristic knowledge of the problem is implicit in the network constraints. It must be stressed that the network is not as general as backpropagation, but is useful at solving a specific kind of problem which backpropagation can only partially explain. The polar backpropagation network is not suitable to general classification problems, but is useful at solving the polar classification problem because it locates the pole.

References

[1] R. W. Burnham, R. M. Hanes and C. J. Bartleson, *"Color: a guide to basic facts and concepts"*, Report of the Inter-Society Colour Council (ISCC) Subcommittee for Problem 20: Basic Elements of Color Education, Wiley, 1963, 249 pages.

[2] R. Hecht-Nielsen, *Kolmogorov's Mapping Neural Network Existance Theorem*, Proc. 1987 International Conference on Neural Networks, Vol 3, 1987, pp 11-14.

[3] R. Hecht-Nielsen, *"Theory of the Backpropagation Neural Network"*, Invited Paper at the 1988 INNS Annual Meeting, 1988.

[4] K. R. L. Godfrey and Y. Attikiouzel, *"Automatic Full-Colour Image Segmentation with Artificial Neural Networks"*, Proc. ACNN'90, Sydney, January 1990.

[5] B. Irie and S. Miyake, *"Capabilities of Three-Layered Perceptrons"*, Proc. 1988 IEEE International Conference on Neural Networks, Vol 1, 1988, pp 641-648.

[6] E. H. Land, *"The Retinex Theory of Colour Vision"*, Scientific American, December 1977, pp 108-129.

[7] R. P. Lippmann, *"An Introduction to Computing with Neural Nets"*, IEEE ASSP Magazine, April 1987, pp 4-22.

[8] D. E. Rumelhart and J. L. McClelland, *"Parallel Distributed Processing"*, Volume 1: Foundations, MIT Press, 1987, 547 pages.

[9] D. E. Rumelhart, G. E. Hinton and R. J. Williams, *"Learning Internal Representations by Error Propagation"*, in [8], pages 318-362.

[10] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink and D. L. Alkon, *"Accelerating the Convergence of the Back-Propagation Method"*, Biological Cybernetics, Springer-Verlag 1988.

Acknowledgement

The Authors are grateful for the support of ATERB and OTC.

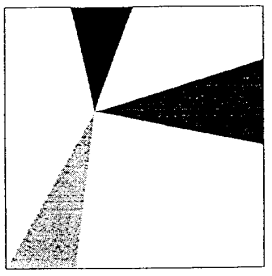

Figure 1: Example of the polar classification problem (PCP) in two dimensions.

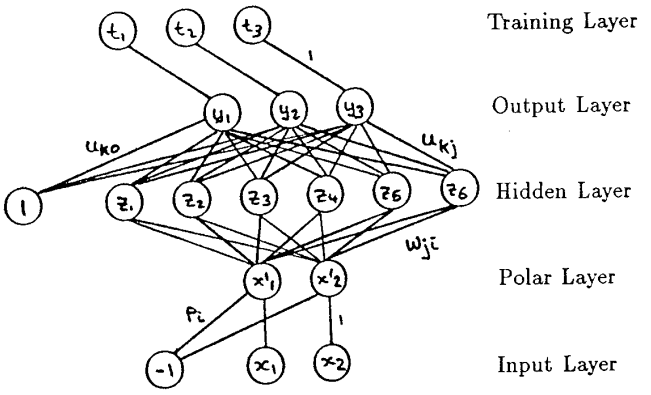

Figure 2: The Polar Backpropagation Architecture.

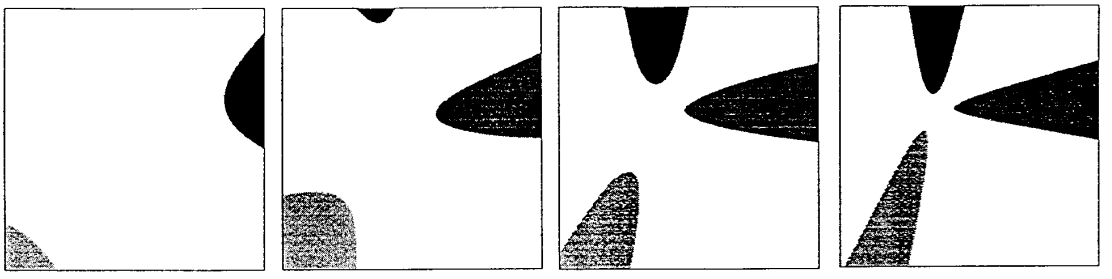

Figure 3: Solution by Backpropagation with $H = 6$ and $\alpha = 0.1$ for 50, 100, 200 and 900 thousand training points.

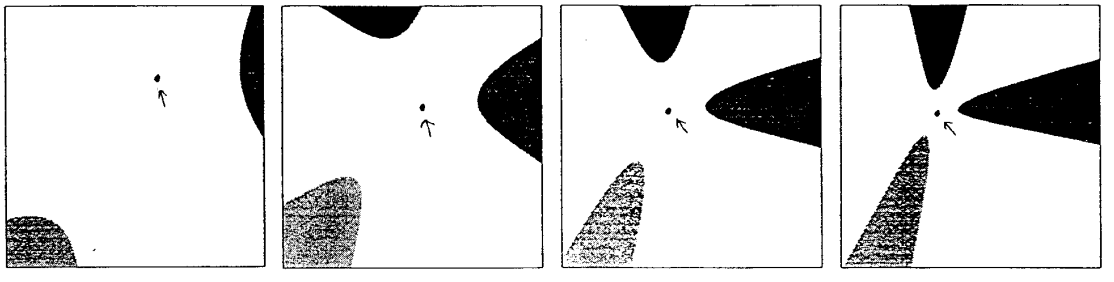

Figure 4: Solution by Polar Backpropagation with $H = 6$ and $\alpha = 0.1$ for 50, 100, 200 and 900 thousand training points. Pole locations marked.