Using Neural Networks to Learn Shape Decomposition by Successive Prototypication

Dr Nicholas Walker

Imperial Cancer Research Fund Laboratories, London

Abstract

I describe a neural-network which decomposes a set of inputs into a sequence of generative parameters. It uses a series of coupled parameter finding and removing networks and requires the input to be in a particular temporal format.

Introduction

If a multi-layer neural-network is required to develope its own representations, a good technique is to teach it to reconstruct its input at its output. The hidden layer has to code the inputs in some way and it is this code that can be used as a representation. Hidden layers taught by back-propagation tend to adopt all-or-nothing responses to one or more 'features' in the input. If the particular inputs are examples from a continuous set, as is the case in shape description, we actually want the hidden units to adopt a continuous coding using one or more parametric variables.

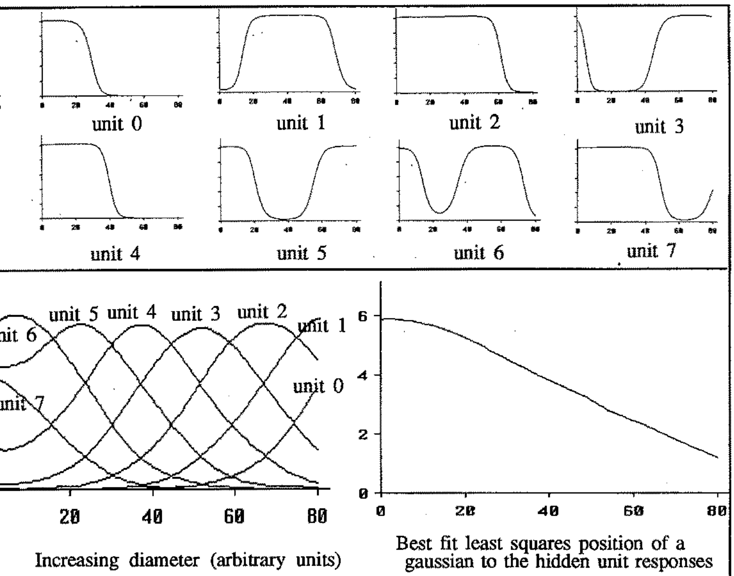
Learning to parameterise

There are several techniques which use a combination of competition between hidden units for activation (winner-takes-all) and adoption of similar activity in neighbouring units (where the neighbourhood relation is defined on some topology) and generate exactly these continuous parametric variables [1,2]. I have chosen Saund's technique as it incorporates back-propagation and so can integrate with hidden unit layers which are not being forced to parameterise. He uses an additional error, related to this mixture of competition for activation and spreading of activity between neighbours, which is added to the back-propagation error in the parameterising hidden units.
Figure 1 shows the hidden unit responses to a set of images of the pixel values of circles of varying diameter centred in the image. The first part of the figure shows the result of back-propagation alone, the units respond in an all or nothing fashion to one or more small ranges of circle diameter. Diameter is then coded by bands where a different set of units are switched on or off. It is not possible, just looking at the hidden unit responses, to know how similar two inputs are to each other, except where their coding is identical. The second part of the figure show a parametric coding - the units lie on a 1D array and their activity presents a single parameter which codes the inputs, in this case it corresponds to the diameter. A least-squares fit of a gaussian to the hidden unit responses shows a virtually linear relationship to diameter, except for very small circles where there is probably a quantisation problem.

Figure 1.

Hidden unit responses to a set of circles of varying diameter. Unit response is up the page, the diameter of the circles across the page. Only back-propagation has been used.

Hidden unit responses to the same input but with an extra error term added to force the units to parameterise the inputs.



Sequential parameterisation

Dealing with inputs requiring many parameters would only seem to involve extending the topology to the necessary dimensions. However there are problems with this even if we can cope with the computational costs.

- The isotropic nature of the function which spreads activation over the neighbourhood means there is an arbitrary relationship between the axes of the internal parametric space and any particular 'natural parameters' of the input set.
- The distribution of the the internal parametric space depends only on the similarity of the inputs and no account can be taken of relationships such as the fact that two inputs close together in time are likely to require similar internal representations (shape constancy).

An alternative is to perform a sequential parameterisation. The value of a single most explanatory parameter is found and this value is used to 'correct' the input for that parameter, i.e. transform the inputs in a way that corresponds to setting that parameter to a fixed, prototypical, value. For example undoing position means transforming the image so that an object is centred on the origin. Then the next most explanatory parameter is found and so on.

Leyton [3] suggests that shape perception consists of just such a sequential parametric decomposition. The parameters can be considered as the generative processes that have formed the shape.

The assumption that allows learning such a series of parameters is that they have a temporal ordering, with each parameter changing faster than those after it. Over some time interval a parameter can assume that the parameters following it will remain constant. We don't have to worry about the parameters before it in the ordering as they will have been removed by the prototyping networks.

Two networks are required to learn the parameterisation. Consecutive sets of inputs are formed from time slices of a larger input set. The first network, having used Saund's technique to distribute a parameter over the first of these sets then continuously modifies this distribution for the following sets. The parameterising hidden layer responses of this network are used to train a second, 3-layer back-propagation network working from the associated input. Eventually this network will generate the parameter over the full set of inputs.

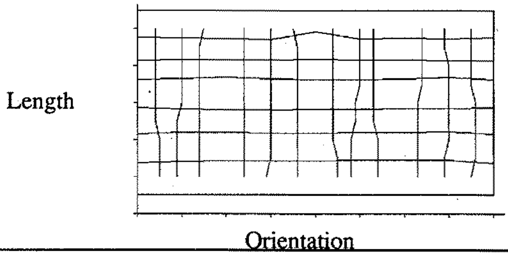
Another network takes this parameter value to effect the prototypication. The best way to construct a neural-network to perform coordinate axis transformations is not known, but Zipser and Anderson [4] use a 3-layer network and I have done the same. To teach it the transformation which will undo the parameter we have to decide whether it represents a prototypical value or not. If it is the associated input is transferred to a stored training image. The transforming network is taught to produce this stored image as output, with the original input plus the output of the parameter assigning network as input.

A two stėp parameterisation

This multiple network training scheme was used to prototype lines of different orientation and length. The orientation changed 100 times faster than the length of the lines. The parametric grid that the two stages produce is shown in figure 2.

Figure 2

Parametric grid for 2 stage parameterisation



Conclusion

The aim of this scheme is to allow a network to generate its own representations of a particular input but to know in advance the properties of the representation it will develope, so it can be used by other networks. The scheme appears to be viable, though only trivial inputs have so far been prototyped.
Its success depends on the data having the right temporal format. An Active Vision System, able to control the position of its sensors, could arrange this. For instance it could arrange that rotations along the axis of the camera are very much less frequent than panning motions of the camera. As the sequential parameters are found it could move its sensors in different ways to correctly format the sensory data.

References

1. Kohonen T. (1984). "Self organisation and Associative memory".
Springer Verlag, Berlin 1984.

2. Saund E. (1989). "Dimensionality Reduction using Connectionist Networks".
IEEE PAMI Vol 11 No 3, pp 304-314.

3. Leyton M. (1985). "Generative Systems of Analysers".
Computer Vision, Graphics and Image Processing 31, pp 201-241.

4. Zipser D. and Anderson R.A. (1988). "A back-Propagation Programmed Network that Simulates Response Properties of a Subset of Posterior Parietal Neurons".
Nature Vol 331, pp 679-684.