http://dx.doi.org/10.1109/IJCNN.1992.287153

Automatic Extraction of Strokes by Quadratic Neural Nets.

M.D.Alder and Y. Attikiouzel
Centre for Intelligent Information Processing Systems
The University of Western Australia
Nedlands, WA 6009 AUSTRALIA

0. Introduction. In [1] we described how an algorithm for adapting quadratic neural nets, in which a single neuron was modelled by a positive definite quadratic form in the space of input patterns, was able to accomplish pattern classification tasks. In particular, when the data set to be classifed had any kind of internal structure such as the double spiral problem, our algorithm was about four orders of magnitude faster than back-propagation. We noted that the method was almost equivalent to well established statistical techniques for mixture modelling, [2], and that it could be compared with standard gaussian mixture modelling by the EM algorithm.

There are several reasons for using a quadratic form model. We may point out the attractions of reconciling neural net pattern classification with the main line of the statistical methods, or simply point to the speed with which classification can be done. Or we can argue that interaction terms are biologically plausible in real neurons. We might even argue that blob and edge detctors in the visual system of the vertebrate eye can be implemented simply by such a model. But the argument we wish to pursue here is that it allows the modelling of an important characteristic of biological learning, namely the inference of higher order structure.

This paper then consists of a preliminary exploration of some ideas from Syntactic Pattern Recognition [3] theory and some insights of Marr [4].

In order to give shape to the somewhat protean theory sketched in the present paper, we consider the concrete problem of Optical Character Recognition of hand written characters. It should be noted that this is a difficult problem; it has resisted all attempts to date to get close to human performance. Its history [5] is a study in naivety: every few years an optimist comes along with a 'new' approach which also doesn't work, and which usually turns out, on examination, to be a variant of an earlier approach. Typically, handwritten digits are used as a data base for training and testing a model and when the source of the digits is not known in advance, perhaps 95% may be correctly identified. Human error rates are far lower.

0-7803-0559-0/92 $3.00

To expect to do any better without careful analysis were foolhardy, and this paper does not attempt to accomplish such character recognition. What it does is to analyse some aspects of the problem with a view to accounting for (1) why it is hard, (2) how the human visual system might accomplish the task and (3) the features of an artifical neural net which might succeed where others have failed.

1. Higher Order Structure. In order to avoid overmuch generality and the confusion which attends it, we give two examples of higher order structure in the case of data sets in the plane where there are two categories of point. In general of course, a 'pattern' is some object such as a subset of the plane, and is represented by a point in some much higher dimensional space, but in order to clarify what we mean by *higher order structure* we reduce the dimension to two. We also deal with only two categories of point, which we call O and X respectively. Then a data set consists of some finite collection of points of these two categories. The pattern classification problem then, is, for a new point presented, to decide to which category it belongs. We observe that there is no sense in which one can guarantee a 'correct' answer to such a problem. What one can do however is to consider the decision which looks natural to the human eye and compare it with the result of existing algorithms.

Example 1. The Chess Board Structure.

```
  XX XX X   OO OO OO        P         OOO OOO O   XX XX XX
 XXX  XXX   OOO OOO O                  OOOOO O    XXXX XX
 O O O O   O XXX XX X  OO O  O OO O  XXX XX XX   OOO OO O
  OO O O O   XX XXXX X  OOO O OOO     XXX X XX   OOOOO OO
 OOO OOO      XX XX  X  OOOO O  O     X XXX XX     OOOOO O
 OOO O  OO    XXX XX X OOOO OO O    XXX XXXX    OOOOO OO
XX XX XXX   OOO OO OO  XXXXX  XX  O OOO  OOO   XX XXX X
XX XX XX  X  OO O OO O  XXX XX XX    OOOOO O     XXXX XX
```

Fig.1. The Chess Board Structure.

The usual algorithms such as the three layer neural net, statistical pattern recognition and k-nearest neighbours, when confronted with the point P in fig.1. will all declare P to be a O, essentially because its nearest neighbours are. The human eye overrides this by using information far less local, to perceive the higher order, global structure. Most human beings would classify P as a X.

It is plain that there exist data transformations which can easily classify the P as a X. To make this observation is to miss the point. The human eye, *on the basis of the data*, decides to apply some such transform. The eye does not have a benevolent engineer around to make decisions on how best to pre-process the data. It has to do it itself, and so should our algorithms.

Example 2. The double spiral. For reasons of space we do not give a figure, but it is easy to imagine a spiral of X's with a spiral of O's on another arm, so that the two are interleaved. If there are subastantial gaps in one of the arms, the neighbouring points will be of the opposite category. The eye has no difficulty in interpolating or extrapolating the spiral structure and again using this to override a purely local classification.

There are many examples outside graphics, in the reading of text for example, where higher level (sometimes called 'contextual') information is applied to reverse a decision taken on a local basis. We take it that there is something fundamental to real neural information processing which acomplishes this.

2. Upwriting and downwriting. We model the process by supposing that at a first layer of processing the same kind of algorithms are employed as were described in [1], that is to say, the data set is modelled as being a union of convex regions which can be described to first order as ellipsoidal. Now one of the critical features about the use of ellipses or positive definite quadratic forms is that they define co-ordinate frames via the eigenbasis. It makes sense therefore to specify the states of the neurons at the lowest level in terms of relationships between them. We propose that the *relative* states of adjacent neurons on the first level be passed back to a second level. We refer to this process as *upwriting* by analogy with grammatical inference of a more traditional, string oriented, sort. The dual process of going from the relationships between the elements of the first order model back to the first order model we call *downwriting*. In the case of string grammars, this last is the rewrite operation on an intermediate symbol to turn it into a teminal symbol string (or another intermediate symbol string if the layering is of many levels).

We illustrate this process by showing how it can automatically extract the strokes out of which hand written cursive characters are constructed (and deconstructed!)

We conjecture, on the basis of examination of a number of OCR systems and their failures, that this process is crucial in human OCR.

3. Automatic Stroke Extraction. Some work has of course been done in trying to extract strokes from cursive text, but there has been no particular attempt to relate this to the information processing or neural aspects so far as we know. Our method is as follows. We first obtained binary TIFF files from a scanner if a number of cases of cursive handwritten characters. We treated the black pixels as data points of one category only and used the algorithm of [1] to fit ellipses to the data set. It was found that the simplest approach to deciding on the best number of ellipses to use was to start with an over supply and prune using ad hoc methods. We found that the EM algorithm for gaussian mixture modelling worked well after being initialised by the algorithm of [1], somewhat less well using more conventional initialisations. Examples of results obtained are shown in fig. 2.

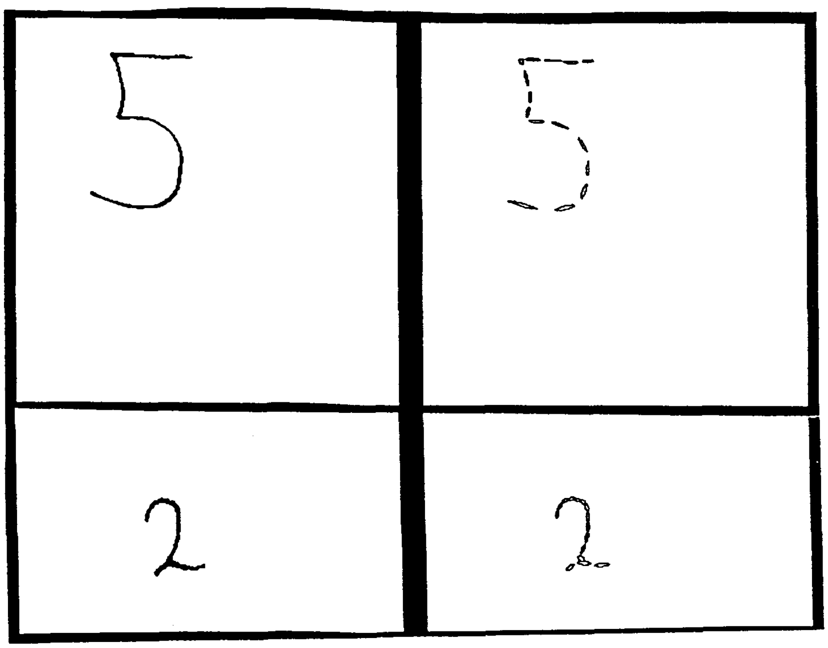

fig.2. Fitting of the first level model.

Having obtained the first level of modelling of the data set, we then went on to model the differences in the first order model at second level. Each ellipse was used as a co-ordinate frame to specify the location of its closest neighbours. In this case there were always precisely two such neighbours, cursive text being the shape it is, except at end points. We shall call them the forward and backward ellipses respectively.

In order to specify the forward ellipse, we simplified by supposing that all were of the same size and aspect ratio, something which experimentally turned out to be approximately the case. Then to specify the forward ellipse we needed two numbers to describe its centre in the co-ordinate frame of the given ellipse, and one number to specify the amount of rotation of the longer axis. Thus the neighbour became a point in a three dimensional space, and similarly a backward neighbour became another. Because we did not orient the ellipses except locally, both forward and backward neighbours were represented by almost identical values in this three dimensional space.

There are a number of interesting cases to consider: the simplest is that of the straight line. Here, the angle was close to zero, and if we call the longer axis the x-axis, x values for the centre were positive and y-values close to zero. Since this happens for all the interior ellipses on a straight line segment, twice, we obtained in the upwrite space a cluster of points representing the neighbourhood structure, itself a roughly ellipsoidal cluster centred at about the point (1,0,0), where the co-ordinates are (x,y,θ).

Another case which is simple enough to study easily is when the ellipses fall on an arc of constant curvature, whereupon again the forward and backward neighbours exist and their locations in the upwrite space comprise a roughly gaussian cluster.

Given the upwrite structure of the line segment or the constant curvature arc, a number of things can be done. One for instance is *smoothing*. Suppose that for a particular ellipse on a straight line, the forward neighbour in the upwrite space is displaced from the point (1,0,0). Then we can move it back to (1,0,0). In doing so, we shift the forward neighbour in the base space. In other words, we can use the mean behaviour expected on the basis of other ellipses, perhaps trained on different data sets, to modify what we actually observe into conformity with what we feel we ought to observe. Anybody who has failed to find typos while proof reading may be familiar with this propensity of human information processing.

Another is *prediction*. From the upwrite space, given an ellipse, we may expect to find a forward and a backward neighbour. At the end points of a line segment, such a neighbour will be missing. We can however compute where is should be, and given a gaussian distribution on the cluster of points centred on (1,0,0) in the upwrite space, we can assign a likelihood function to the prediction. We can also compute an *entropy* when our neighbour is displaced from the predicted position.

Now it is easy to see how to do the chunking of parts of the cursive character. We use the upwrite space to do a forward and/or backward prediction. If the prediction is confirmed up to some tolerance, we may perhaps regularise the next ellipse, or interpolate, and proceed there in the base space. We now make another prediction in the same direction. We proceed in this way to travel along a curve until the prediction derived from the upwrite space fails. This represents some unacceptably high entropy in the prediction process. At this point, our chunk terminates. It is bounded by entropy maxima, just as in English text the white spaces represent the entropy maxima of a local letter level predictor of the next character. It is easy to apply this to extract line segments and constant curvature parts of a cursive image. In a subsequent paper we shall describe how more complex entities can be identified as 'chunks', elements which themselves become eligible for the same process as was applied to pixels in the base space.

It is noteworthy that the data in the upwrite space when stored has applications to other data of a quite different provenance. In particular, vertical lines on one scale allow for smoothing, interpolation and extrapolation to horizontal lines on a different scale. Allowing for variations in the density of ellipses, circles of one size allow for the prediction smoothing and interpolation for circles of a different size in a different location. We have effected a transfer of learning.

4.Summary and Conclusions. That chunking is a crucial property of human cognitive processes is a cliché of Psychology. That human OCR of cursive script entails both upwriting and downwriting into strokes and presumably other structures is eminently plausible, as an examination of the differences between human and machine OCR makes clear. And finally that this is accomplished by arrays of neurons in the central nervous system is indisputable. In this paper we have outlined algorithms for accomplishing the task with model neurons.

5. References.

[1] Lim,S.G., Alder,M.D., Hadingham, P and Attikiouzel,Y: *Adaptive Quadratic Neural Nets*. Proceedings IJCNN '91.

[2] Titterington,Smith and Markov *Statistical Analysis of Finite Mixture Distributions*. John Wiley, 1985

[3] Fu, K.S. *Syntactic Pattern Recognition*

[4] Marr, D. *A theory for cerebral neocortex* Proc.Roy.Soc Lond.B 176,161-234 (1970)

[5] Tappert, et al:*The state of the art in on-line handwriting recognition*. IEEE Trans. Pattern Recog. and A.I.Vol.12 No.8 August 1990 pp787-808