Artificial Neural Networks, 2
I. Aleksander and J. Taylor (Editors)

Combining Two Methods of Recognizing Hand-Printed Digits

Geoffrey E. Hinton, Christopher K. I. Williams and Michael D. Revow
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 1A4

ABSTRACT

Hand-printed digits can be recognized quite well by a feedforward neural network that uses equality constraints between weights to achieve limited translational invariance. However, the net has no explicit model of what a digit looks like and this can lead it to make confident errors. An alternative approach, which incorporates much more prior knowledge, is to use explicit deformable models of the digits and to recognize a digit by finding which model fits best. We describe a system that uses learned digit models which consist of splines whose shape is governed by 8 control points. The elastic models are good at capturing shape knowledge, and the elastic matching process is good at rejecting parts of the image that are best explained as noise. However, the elastic matching is slow and can get trapped in local optima if the initial configuration of the elastic model is far from the actual data. So we are developing a hybrid system that combines the best aspects of both approaches. First, the slow elastic matching method is used to accurately label the training data with all the instantiation parameters of the correct digit. Then a feedforward network is trained to produce the fully instantiated digit, rather than just the class of the digit. After training, the neural net is used to initialize the elastic models, and the elastic matching is used to reject erroneous hypotheses of the neural network.

1 OVERVIEW

Given good bottom-up segmentation and normalization, a carefully constrained, feedforward neural network can recognize hand-printed digits in zip codes fairly accurately. (le Cun et al., 1990). After training, the network is very fast but it has a number of weaknesses that limit its final performance and lead it to require large sets of training examples:

1. The network must extract almost all its knowledge from training data because it has no prior knowledge about the shapes of the digits. It does not even know that digits are composed of one-dimensional strokes.
2. Even though each image contains a lot of information, each training example provides only $\log_2 10$ bits of information about the desired input-output relationship. For supervised learning, it is the amount of information in the *output* vector that is important in constraining the weights, not the amount of information in the input vector. So to constrain thousands of weights, many thousands of examples are needed.
3. Very occasionally, images that look nothing like a particular digit are confidently classified as that digit.

4. During recognition, the network does not segment the image into parts that constitute the digit and other parts that are noise. So it cannot be used to pinpoint segmentation errors. All it can do is indicate whether a particular segmentation led to confident recognition of a digit.

5. During recognition, the network does not explicitly decide on the instantiation parameters of the digit (i.e. its position, size, orientation, shear, elongation *etc.*). So it is unable to make use of consistencies between the instantiation parameters of neighbouring digits.

These weaknesses of feedforward neural network classifiers and the success of model-based approaches to shape recognition (e.g Burr, 1981a, 1981b; Lowe, 1991) led us to investigate a very different recognition method that uses deformable elastic models of the digits (Hinton, Williams and Revow, 1992). Training data can still be used to improve the models, but much less data is needed because we start with hand-designed models that are approximately correct and each model only has a small number of adaptive parameters. Elastic models of the type we use seem to be good at capturing all the possible variations of the shape of a digit with just a few parameters. But fitting the models to data takes a long time, especially if we use multiple different initial configurations of each model to circumvent local minima in the matching.

We believe that all five of the weaknesses we attribute to the standard way of using a neural network can be overcome by using a hybrid, two-stage recognition process. The first stage is a feedforward, multi-layer neural network that has a very large number of output units divided into 10 groups, one per digit class. Within each group we use a distributed pattern of activity over the output units to represent a particular instantiation of a digit (see section 6). By fitting a Gaussian to the activity pattern, we can explicitly extract the instantiation parameters, and they are then used to initialize the elastic model of the digit. Only a few iterations of elastic matching are then required to decide whether any nearby instantiation of the digit model can explain the image data.

2 ELASTIC MODELS

One technique for recognizing a digit is to perform an elastic match with many different exemplars of each known digit-class and to pick the class of the nearest neighbour. Unfortunately this requires a large number of elastic matches, each of which is expensive (though Simard & Le Cun, *pers. comm.*, 1992, have recently developed a relatively cheap way of finding the nearest neighbour of an intensity image when no distance cost is incurred for *slight* affine transformations of the stored templates). By using one elastic model to capture all the variations of a given digit we greatly reduce the number of elastic matches required. We describe a type of elastic model that is based on splines. Each elastic model contains parameters that define an ideal shape and a deformation energy for departures from this ideal.

Each digit is modelled by a deformable spline whose shape is determined by the positions of 8 control points. Every point on the spline is a weighted average of four control points, with the weighting coefficients changing smoothly as we move along the spline. In computing the weighting coefficients we use a cubic B-spline and we treat the first and last control points as if they were doubled. To generate an ideal example of a digit we put the 8 control points at their home locations for that model. To deform the digit we move the control points away from their home locations. Currently we assume that, for each model, the control points have independent, radial Gaussian distributions about their home locations. So the negative

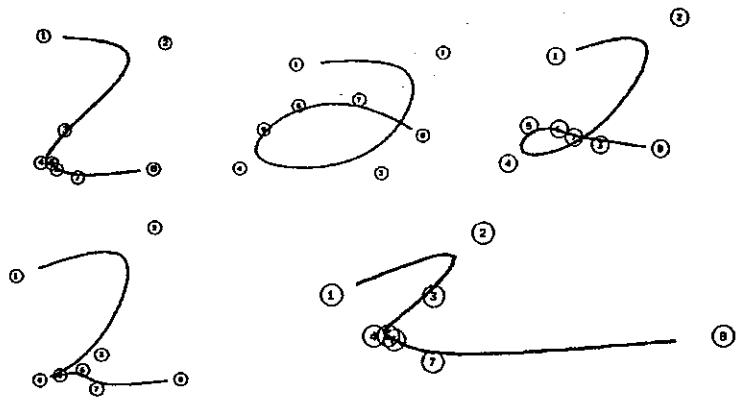

Figure 1: This shows the locations of the control points and the resulting splines when the 2 model is matched to some instances extracted from zip codes.

log probability of a deformation (its energy) is proportional to the sum of the squares of the departures of the control points from their home locations.

Using a spline it is easy to model topological variants of a digit. The loop of a 2, for example, can smoothly turn into a cusp or an open bend (see figure 1). These variants are produced by small changes in the relative locations of the relevant control points. This advantage of spline models is pointed out by (Edelman, Ullman and Flash, 1990) who use a different kind of spline that they fit to on-line character data by directly locating candidate control points in the image.

The deformation energy function only penalizes shape *deformations*. Translation, rotation, dilation, elongation, and shear do not change the shape of an object so we want the deformation energy to be invariant under these affine transformations.[1] We achieve this by giving each model its own "object-based frame" and computing the deformation energy relative to this frame. When we fit the model to data, we repeatedly recompute the best affine transformation between the object-based frame and the image (see section 4). The repeated recomputation of the affine transform during the model fit means that the shape of the digit is influencing the normalization. Having an explicit representation of the affine transformation of each digit should prove very helpful for recognizing multiple digits, since it will allow us to impose a penalty on differences in the affine transformations of neighbouring digits.

Although we use our digit models for recognizing images, it helps to start by considering how we would use them for generating images. The generative model is an elaboration of the probabilistic interpretation of the elastic net given by Durbin, Szeliski & Yuille (1989).

To generate a noisy image of a particular digit class, run the following procedure:

- Pick a deformation of the model (i.e. move the control points away from their home locations). This defines the spline in object-based coordinates. The probability of picking

[1] Currently we do not impose any penalty on extremely sheared or elongated affine transformations, though this would probably improve performance.

a deformation is proportional to $e^{-E_{def}}$.

- Pick an affine transformation from the model's intrinsic reference frame to the image frame (i.e. pick a size, position, orientation, slant and elongation for the digit).
- Map the spline into image coordinates and space circular Gaussian ink generators (beads) uniformly along its length. The number of beads on the spline and their variance can easily be changed without changing the spline itself.
- Repeat many times:
  Either (with probability π_{noise}) add a randomly positioned noise pixel
  Or pick a bead at random and generate a pixel from the Gaussian distribution defined by the bead.

3 RECOGNIZING ISOLATED DIGITS

We recognize an image by finding which model-class is most likely to have generated it. Each possible model-class is fitted to the image and the one that has the lowest cost fit is the winner. The cost of a fit is the negative log probability of generating the image given the model-class.

$$E_{ideal} = -\log \int_{\substack{I \in model \\ instances}} P(I)\, P(image \mid I)\, dI \tag{1}$$

We can approximate this by just considering the best fitting model instance and ignoring the fact that the model should not generate ink where there is no ink in the image.[2]

$$E = \lambda\, E_{deform} - \sum_{\substack{inked \\ pixels}} \log P(pixel \mid best\ model\ instance) \tag{2}$$

The probability of an inked pixel is the sum of the probabilities of all the possible ways of generating it from the mixture of Gaussian beads or the uniform noise field.

$$P(i) = \frac{\pi_{noise}}{N} + \frac{1 - \pi_{noise}}{B} \sum_{beads} P_b(i) \tag{3}$$

where N is the total number of pixels, B is the number of beads, π_{noise} is the mixing proportion of the uniform noise field, and $P_b(i)$ is the probability of pixel i under Gaussian bead b.

4 FITTING A MODEL TO AN IMAGE

Every Gaussian bead in a model has the same variance. When fitting data, we start with a big variance and gradually reduce it as in the elastic net algorithm of Durbin and Willshaw (1987). This can be viewed as a continuation method in which the solution at one variance is used to initialize the search for a solution at a lower variance (Blake and Zisserman, 1987). Each iteration of the elastic matching algorithm involves three steps:

[2] If the inked pixels are rare, poor models sin mainly by not inking those pixels that should be inked rather than by inking those pixels that should not be inked.

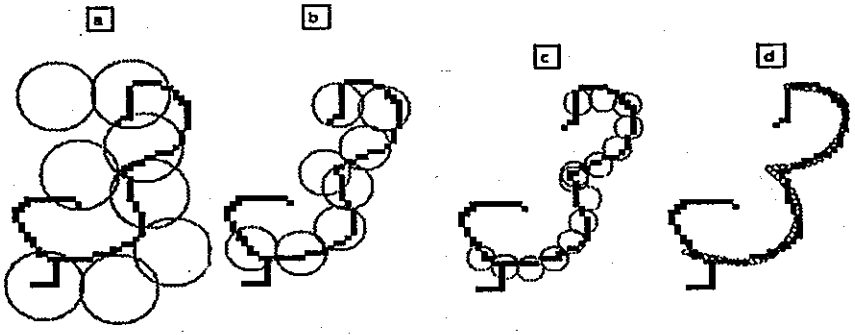

Figure 2: The sequence (a) to (d) shows some stages of fitting a model 3 to some data. The grey circles represent the gaussian beads, with the radius representing the standard deviation. (a) shows the initial configuration, with eight beads equally spaced along the spline. In (b) and (c) the variance is progressively decreased and the number of beads is increased. The final fit using 60 beads is shown in (d). We use about three iterations at each of five variances on our "annealing schedule". In this example, we used $\pi_{noise} = 0.3$ which makes it cheaper to explain the extraneous noise pixels and the flourishes on the ends of the 3 as noise rather than deforming the model to bring Gaussian beads close to these pixels.

- Given the current locations of the Gaussians, compute the responsibility that each Gaussian has for each inked pixel. This is just the probability of generating the pixel from that Gaussian, normalized by the total probability of generating the pixel.
- Assuming that the responsibilities remain fixed, as in the EM algorithm of Dempster, Laird and Rubin (1977), we invert a 16×16 matrix to find the image locations for the 8 control points at which the forces pulling the control points towards their home locations are balanced by the forces exerted on the control points by the inked pixels. These forces come via the forces that the inked pixels exert on the Gaussian beads.
- Given the new image locations of the control points, we recompute the affine transformation from the object-based frame to the image frame. We choose the affine transformation that minimizes the sum of the squared distances between the control points and their home locations. The residual squared differences determine the deformation energy.

Some stages in the fitting of a model to data are shown in figure 2. The search technique usually avoids local minima when fitting models to isolated digits. But, if we detect an unsatisfactory fit, we try alternative starting configurations for the models.

5 THE PERFORMANCE OF ELASTIC MODELS

We describe some preliminary results on the performance of our elastic models in Williams, Revow and Hinton (1992). The home locations of the control points in each elastic model are initially set by hand, but they are then revised by using discriminative of maximum likelihood learning. We have not yet performed a systematic comparison of this technique with other

techniques on a standard database, but it appears that the shape knowledge in the models is adequate: The correct model almost always fits better than the incorrect ones, provided the search finds the best fit of the model. Also, the correct elastic model can reliably reject as noise those parts of the image that do not depict the basic shape of the digit (see figure 2).

A major weakness of the elastic model approach is the slowness of the search and the fact that it occasionally fails to find the global optimum, even if restarted from several different configurations. If we could be sure that the elastic matching started from a nearly correct configuration, we would only need to run a few iterations at low variance, and we would be very unlikely to get trapped by local optima. This suggests a hybrid method that uses neural nets to get close to a solution and elastic matching to confirm the fit.

6 REPRESENTING INSTANTIATED DIGITS WITH RBF's

Using our elastic models, the instantiated digits of a particular class lie in a 22-dimensional space. Six of these dimensions jointly represent the position, size, orientation, shear and elongation of the digit instance, and the 16 remaining dimensions represent the deformation of the digit.

The obvious way to extract the 22 instantiation parameters is to use a separate, real-valued output unit for each parameter. This method should be tried, but we anticipate that it will be difficult for a network to explicitly extract the parameters in this way because, in the presence of noise, the image data does not provide separate evidence for each parameter. Instead, combinations of nearby pixels provide evidence for quite high-order combinations of underlying parameter values. So we have decided to use an output representation in which each instantiated digit is represented by the centre of gravity of a pattern of activity over many class-specific output units. Each output unit is a radial basis function that knows where it is centered in the 22-dimensional instantiation space. Notice that in determining the desired activity of an output unit, we treat it as an RBF in instantiation space, but in determining its actual activity we can treat it as a standard sigmoid unit that combines evidence coming from other units.

The desired representation of a particular digit instance corresponds to activating each output unit by a Gaussian function[3] of its distance, in instantiation space, from the location of that instance. If this desired activity can be achieved, there will be a Gaussian blob of activity over the 22-dimensional space, with the centre of the blob representing the instantiation parameters, and the area under the blob representing the probability that the digit is present. We can locate the centre of gravity of the blob by fitting a mixture of a Gaussian and a uniform distribution in instantiation space. The mean of this Gaussian makes explicit the neural network's estimate of the 22 instantiation parameters. The second stage of the hybrid system then uses this information to initialise the elastic models of probable digits and matches these models to the image data.

When using a coarse-coded representation of this type, there are several important issues which we only have space to mention briefly. The first issue concerns the *accuracy* with which the centre of gravity of the Gaussian blob of activity encodes the instantiation parameters. A good impression of the accuracy can be obtained by encoding a particular digit instance as the desired pattern of activity on the output units and then reconstructing the digit instance from the centre of gravity of the activity pattern (see figure 3).

[3]It might be better to use non-Gaussian, non-radial local basis functions (Ballard, 1987).

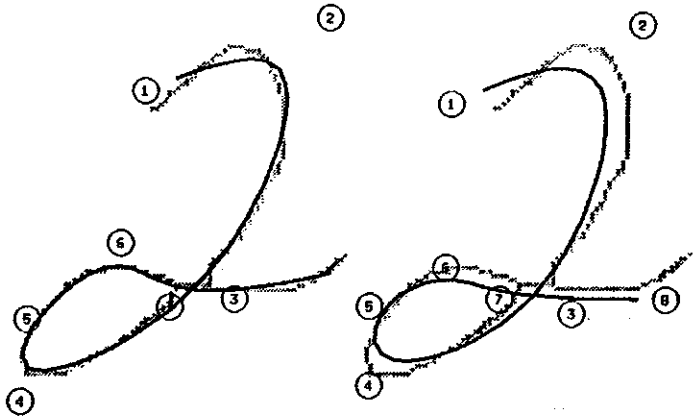

Figure 3: After fitting the 2 model to an image, we encode the 22 instantiation parameters as a pattern of activity over 200 Gaussian RBF units. Then we take the center of gravity of this pattern and reconstruct the spline. The figure shows this reconstructed spline superimposed on the inked pixels in the image.

One way to avoid needing more RBF's as the dimensionality increases is to make the width of each RBF proportional to the square root of the dimensionality. However, increasing the width of the RBF's maintains accuracy at the cost of decreasing the resolution of the representation (see below). For any given dimensionality, we can also reduce the *linear* error in the centre of gravity by a factor of $\sqrt{N}$ by increasing the number of RBF's by a factor of N. This does not affect the resolution.

The *resolution* of the representation (*i.e.* its ability to discriminate between simultaneously presented instantiations) is a much more complicated issue, which is relevant even if only one digit instance is present in the image. The output representation needs to act like a Hough space which non-linearly combines evidence from feature detectors in the previous layer of units. When several different feature detectors all suggest the same digit instance we want that instance to be represented, but when different detectors all suggest different instances, we would like the output representation to remain inactive. So we require sufficient resolution to detect agreement between the outputs of feature detectors. If this resolution can be achieved, the use of multiple layers of RBF spaces is a very natural alternative implementation of the TRAFFIC system described by Zemel, Mozer and Hinton (1990). In TRAFFIC, weight matrices store the fixed linear coordinate transforms from parts to wholes, and the non-linearities of units are used to decide if multiple parts agree sufficiently well for a new whole to be instantiated. So the linear part of a standard neuron does coordinate transforms, and the subsequent non-linear stage does model-based segmentation.

The idea of using multiple output units to allow a neural net to represent alternative instantiations of a phoneme or digit is not new (Lang, Waibel, and Hinton, 1990; Keeler, Rumelhart and Leow, 1991). But previous systems have attempted to train the net by back-propagating class information through a final integrator that simply adds up the activities

of the instantiation-tuned units. This makes the learning much harder than it is if each instantiation-tuned unit is explicitly given a desired activity level. We are able to provide this richer desired output by fitting our elastic models (slowly) to get the instantiation parameters.

We hope to achieve fast, accurate recognition by using an elegant but slow generative model to train and filter the outputs of a neural network, but we do not yet know whether this approach will actually work.

Acknowledgements

This research was funded by Apple and by the Ontario Information Technology Research Centre. We thank Allan Jepson, Richard Durbin, Rich Zemel and Yann Le Cun for helpful discussions. Geoffrey Hinton is the Noranda Fellow of the Canadian Institute for Advanced Research.

References

Ballard, D. H. (1987). Interpolation Coding: A Representation for Numbers in Neural Models. *Biological Cybernetics*, 57:389–402.

Blake, A. and Zisserman, A. (1987). *Visual Reconstruction*. MIT Press.

Burr, D. J. (1981a). A dynamic model for image registration. *Comput. Graphics Image Process.*, 15:102–112.

Burr, D. J. (1981b). Elastic matching of line drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 3(6):708–713.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Proc. Roy. Stat. Soc.*, B-39:1–38.

Durbin, R., Szeliski, R., and Yuille, A. L. (1989). An analysis of the elastic net approach to the travelling salesman problem. *Neural Computation*, 1:348–358.

Durbin, R. and Willshaw, D. (1987). An analogue approach to the travelling salesman problem. *Nature*, 326:689–691.

Edelman, S., Ullman, S., and Flash, T. (1990). Reading cursive handwriting by alignment of letter prototypes. *Internat. Journal of Comput. Vision*, 5(3):303–331.

Hinton, G. E., Williams, C. K. I., and Revow, M. D. (1992). Adaptive elastic models for hand-printed character recognition. To appear in *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson and R. P. Lippmann eds, San Mateo, CA: Morgan Kaufmann.

Keeler, J. D., Rumelhart, D. E., and Leow, W.-K. (1991). Integrated Segmentation and Recognition of Hand-Printed Numerals. In Lippmann, R. P., Moody, J. E., and Touretzky, D. S., editors, *Advances in Neural Information Processing Systems 3*, pages 557–563.

Lang, K. J., Waibel, A. H., and Hinton, G. E. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3:23–43.

le Cun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan Kaufmann.

Lowe, D. G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-13:441–450.

Williams, C. K. I., Revow, M. D., and Hinton, G. E. (1992). Hand-printed digit recognition using deformable models. To appear in *Spatial Vision in Humans and Robots*, L. Harris and M. Jenkin eds, Cambridge University Press.

Zemel, R. S., Mozer, M. C., and Hinton, G. E. (1990). TRAFFIC: Recognizing objects using hierarchical reference frame transformations. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems, Vol. 2*, pages 266–273. Morgan Kaufmann, San Mateo, CA.