Image Labelling With A Neural Network

W.A.Wright *
Research Initiative in Pattern Recognition (RSRE)
St Andrews Road, Malvern
Worcs WR14 3PS.

The lack of contextual integrity in region labelling schemes for segmented visual images has been a long standing problem in computer vision. In the past it has been common to adopt some form of relaxation scheme, such as relaxation labelling, to remove both labelling ambiguities and labellings which, usually, are obviously contextually incorrect. So far no general region labelling system has been found.

A neural network has now been applied to the above problem. A simple multi-layer perceptron, trained on the relative positions of, and the unary features pertaining to, a set of regions obtained from image segmentations has been shown to be capable of finding roads in natural scenes.

The correct labelling of segmented images has been a difficult and generally unsolved problem in computer vision for many years (see [1,2]). Most labelling systems at the moment rely upon obtaining a tentative initial set of labels for the regions in a segmented image and then applying contextual knowledge to try and resolve the labelling ambiguities that inevitably occur. The tentative labelling schemes, such as Euclidean distance or K nearest neighbour clustering for example [3,4], usually act by comparing a measure of the unary features of a region (features internal only to the region) against the features of typical region types derived from a training set. The results from this method are quite often ambiguous: a region may have several possible labels. To try and rectify these ambiguities the contextual significance of a region's label with respect to the labels of the neighbouring regions is used. Schemes such as relaxation, or probabilistic relaxation labelling [5,6], use such contextual knowledge. Although these schemes work well with edge data in real images [7] and region segmentations of artificial images [2] they are found to be less robust with region segmentations of real imagery. This is because it proves exceptionally difficult to produce a complete and general set of rules to describe the contextual relationships between labelled regions for a real image [8].

Learning mechanisms such as those put forward by Michalski et al [9] and Quinlan [10] could be used to generate a suitable set of contextual rules directly from examples of region data. However, apart from a few exceptions, such as those given in references [11,12], very little work has been carried out in computer vision using these methods.

*Sowerby Research Centre, Advanced Information Processing Department, FPC 267., British Aerospace, PO Box 5., Bristol. BS12 7QW

An alternative method may be to use a neural network. The neural method proposed by Hopfield and Tank [13] has been used to optimise the tentative labellings of the regions present in an image [14]. However, the method only considers artificial images, and does not find the contextual rules relating regions in the image itself. The method, therefore, has the same drawbacks as the relaxation methods mentioned above. Other neural network methods which are capable of being trained could be more useful. Such a network is the multi–layered perceptron [15]. This has been shown to be robust against uncertain data [16] and able to extract the contextual relationships from real training data [17]. In fact recent work, related to the work presented here [18], has shown that a multi-layered perceptron is capable of forming a set of contextual relationships between the labelled regions in a segmented image by training it on correctly labeled images. Further, Sejnowski's and Rosenberg's [17] work shows that a network does not necessarily require a tentative labelling of the features present in the input data to be able to form contextual relationships. Unlike the conventional labelling methods, therefore, it should be possible to design a neural network implementation which only uses contextual and statistical features that are obtained directly from an image, and does not require any initial region labelling at all. This article describes the preliminary results of a study into the possibility of using such a neural network to label the regions obtained from real images of natural outdoor scenes, by first training it on such scenes.

Network

The type of network used in this implementation is known as the multi-layered perceptron [15]. To try and keep this article as brief as possible it has been assumed that the reader has a limited knowledge of the principles behind these ideas. However, for completeness, a limited description of this type of network is given here.

The multi-layered perceptron (MLP), as the name suggests, consists of layers of simple non-linear processing units. These units are highly connected and "threshold", usually with a sigma function, the sum of the unit's inputs to give an output. Each unit in a layer is connected to the units in layers adjacent to that layer via a weighted link (see figures 1 and 2).

The network *feeds-forwards* signals from its previous layer, via the non-linear function, to the next layer. The weights on the links together with the non-linear function on the unit allow non-linear relationships to be encoded between the network's layers and therefore between its input and output.

AVC 1989 doi:10.5244/C.3.39

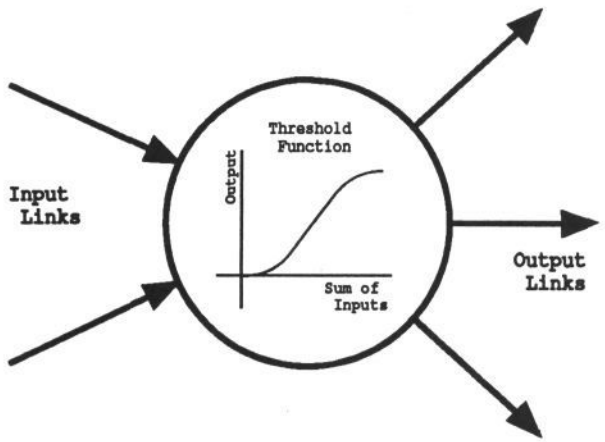

Figure 1: Typical neural element with a sigma function threshold

A set of weights for a given problem may be *learnt* by training the network, on known examples, with an algorithm known as *error-back propagation* [15]. Error-back propagation is a gradient descent algorithm. The algorithm allows the weights on the network to be altered in proportion to an error generated by taking the Euclidean distance of the network's actual output, for a given input, from what is considered to be the true output of the network for that input. How the network is tailored to the particular labelling problem discussed here is is explained in the next section. A more complete description of the ideas behind neural networks may be found in Rumelhart and McClelland [19].

Implementation

To ease the amount of computation during this initial study the network was only required to look for one type of region ie. *roads*. Although this is a simplification the implementation is still relevant for other types of region such as fields or vehicles for instance. The implementation, therefore, differs from that of Pomerleau [20] which is tailored solely for the detection of roads.

A three layer multi–layer perceptron [15] was designed, as described below, to act on a region segmentation of an image and output a description (a label) of the regions in the image. The aim was to label every region (correctly) as *road* or *not-road*. A typical example of an image and its associated region segmentation is given in figures 3 and 4. The segmentation was obtained by processing the image with a region based segmentation algorithm, *coalesce*, designed at British Aerospace [21]. However, in principle any other reasonably efficient segmentation algorithm could have been used.

The region features in the segmented image were presented to the network on a region by region basis. The

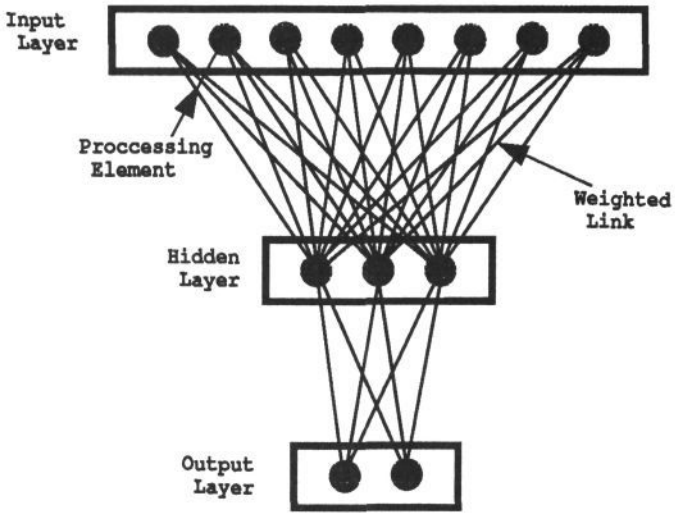

Figure 2: Schematic diagram of a three layered MLP

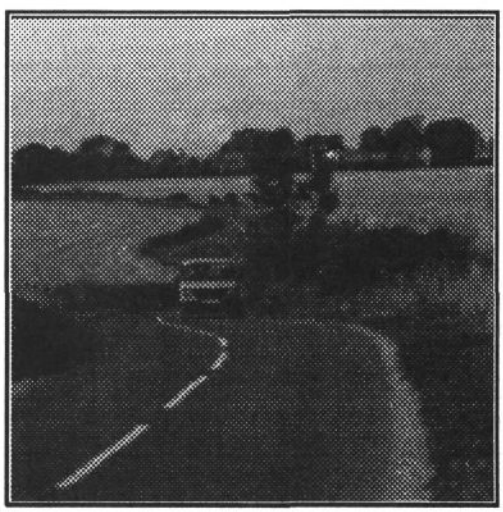

Figure 3: Test image

network was configured to allow this to happen as follows:

layer 1: 89 Input units. 80 of these units coded the position and statistical features, as described below, of up to 8 regions adjacent to the region to be classified (the *central region*). The features from the 8 adjacent regions allowed contextual information about the *central region* to be presented in a local manner. The remaining 9 units coded the *central region's* unary features, also described below.

layer 2: 16 Hidden units (8 and 4 Hidden units where also tried for this implementation). Although the network converged with the smaller configurations the weight set obtained was not as good as that found with the 16 hidden unit configuration. With a much larger number of hidden units (eg. 30) the network did not converge.

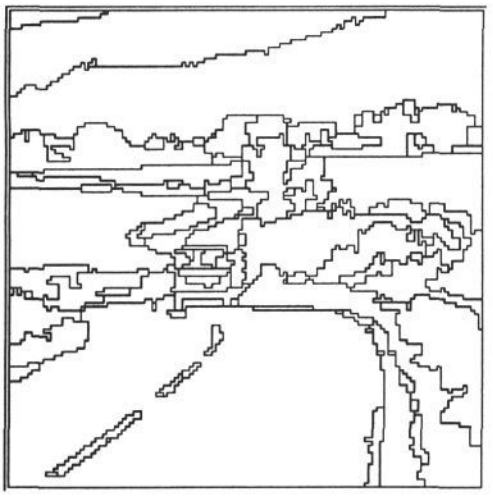

Figure 4: Segmentation of the test image

Angle	Code
338-22	1 0 0 0
23-67	1 1 0 0
68-112	0 1 0 0
113-157	0 1 1 0
158-202	0 0 1 0
203-247	0 0 1 1
248-292	0 0 0 1
293-337	1 0 0 1

Table 1: Gray coding for angles between region centroids

layer 3: 2 Output units. This allowed the coding of *road* [1,0] and *not-road* [0,1] for the *central region* label.

The position of each adjacent region relative to the *central region* was found by taking the angle measured clockwise from the positive Y-axis to the line joining the centroids of both regions. Each angle was given to the network as one of eight Gray codes, taking up four elements of the input for each region, see table 1. Gray codes were used because the code is cyclic and therefore has no discontinuity at 360 degrees. The Gray code angle [0,0,0,0] for the *central region* was also given for completeness. The remaining six input elements for each adjacent region coded the features:

1. mean grey level: this value was normalised by dividing the value by the maximum grey level value of 255,
2. standard deviation of the grey levels inside the region: this value was normalised in the same way as for 1,
3. homogeneity of the grey levels (see [22]),
4. relative area of the region: this is the ratio of the region area with respect to the area of the whole image, in pixels,
5. compactness of the region, $\frac{4\pi \times area}{(perimeter\ of\ region)^2}$,
6. adjacent-perimeter length: this is the ratio of the perimeter common to the adjacent region and the *central region* with respect to the perimeter of the *central region*. This value was not given for the central region.

As is pointed out some of the above values were normalised. This ensured that all the values presented to the network were bounded on the unit interval and, thus, no artificial bias was introduced into the initial stages of training as this would slow the network's convergence to its final configuration.

All the above features and angles were presented to the network as an 89 element long string. The first 9 elements of the string held five *central region* statistics, in the order given above, together with its angle code. The next 80 units, divided into 8×10 blocks, contained the statistics and angle codes for the 8 adjacent regions. When more than 8 adjacent regions were present the 8 with the largest adjacent-perimeter were taken. For regions with less than 8 neighbours null codings (zeros) were given for the unoccupied inputs. The ordering of the adjacent-region statistics in the input was determined by the adjacent-perimeter length. The features for the region with the largest adjacent-perimeter were placed closest to the central region's features and so on.

The choice of only using up to 8 regions is, to a certain extent, an arbitrary one. A region can have anything from between 1 to $\sim$ 100 adjacent regions. However, one finds that the regions polarise between those that are highly connected (on average $\sim$ 40 adjacent regions), and those that have a low connectivity (on average $\sim$ 4). If the input were extended to allow for a large number of adjacent regions the training of the network would become too computationally expensive. 8 adjacent regions were taken as a compromise since it allows all the adjacent regions to be taken into account for the regions with a low connectivity, and allows the network to realise that the region possibly has a high connectivity (ie. when more than 7 adjacent regions are present).

The way that the adjacent region information has been selected and presented to the network obviously encodes a certain degree of prior information about the type of region that the network is trying to label. This is also true about the selection of the other region features used in this particular example. This type of information may be termed *extrinsic*; that is the information given to the network by the operators choice of feature rather than the information present in that data which may be termed *intrinsic*. Extrinsic information maybe likened to what is termed "background information" in the field of artificial intelligence. The amount of extrinsic information given to the network is important since too large an amount can over constrain the network. Under these circumstances the operator restricts the problem for which the network was designed, reducing the ability of the network to generalise. It is also found, in these situations, that the more simple statistical clustering algorithms, such as K nearest neighbour, can be used on the data

True Label	Assigned Label		Freq
	Road	Not-Road	
Road	**66.70**	33.30	48
Not-Road	17.30	**88.70**	52

Table 2: Best guess estimates (%) for road detection network on the test data

True Label	Assigned Label		Freq
	Road	Not-Road	
Road	**82.60**	17.40	23
Not-Road	33.00	**67.00**	97

Table 3: Best guess estimates (%) for road detection network for the test image

instead of the type of adaptive network discussed here [23]. This problem in relation to this network is discussed further in section .

The network was trained on the features described above using error-back-propagation [15]. The training data was obtained from the segmentations of 36 images of differing outdoor scenes obtained from the Alvey data base produced for the MMI 007 project. To allow the training to take place the *true* identities of the regions contained in these segmented images were obtained by hand labelling the segmentations prior to training. This gave $\sim$ 250 examples of *road* regions. In addition $\sim$ 250 *not-road* regions, evenly sampled from the most frequent types of region present in the segmented images, were also taken as counter examples. The order that the regions were presented to the network was randomised to prevent any bias being introduced into the training. The weights on the network's links were updated after each presentation of a region and its associated neighbours' features. This continued until the total error over the training set, generated by the network, converged, ie became stable. Typically for this network, to produce a small error, this took 2,000 iterations (one iteration being the application and weight update of all the vectors in the training data).

Results

Once an adequate convergence on the training data was obtained, the network's weight configuration was tested on:

Test data: Region data which had been used to form the *road/not-road* data but had been removed and therefore not used to train the network.

A test image: A complete image which had *not* been used to form the *road/not-road* data (see figure 3). This image, therefore, was not used to train the network.

The performance of the network was determined by comparing the output of the network against the true output or label. These results are shown in table 2 for the test data, and table 3 for the test image. The assigned label was found by taking the label with the closest Euclidean distance to the networks output.

The action of the network on the test image reflects its performance on the test data (see tables 2 and 3). Close inspection of these results suggests that this implementation is able to use contextual information; figure 5

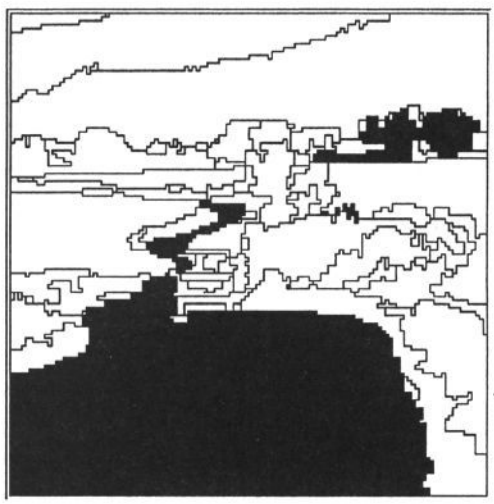

Figure 5: Segmentation of the test image with the regions labelled by the neural network as *road* displayed in black.

demonstrates this. The white lines in the road, and the gravel sides of the road, as desired, are labelled as *road* just as is the main piece of tarmac. Although some of the regions are labelled incorrectly there is some suggestion that the network is stable against uncertain data as it was able to cope with the poor segmentation between the vehicle and the road.

The weights generated by the learning mechanism can, in some cases, be interpreted as rules which describe how the network differentiated between *road* and *not-road* regions. The number of rules that have been determined is very limited at the moment but they mainly mirror those rules that experienced workers in the area regard as important. However, in some cases, the network appears to have highlighted new relationships which were not originally thought to be significant. For example, *compactness* is shown to be an important signature when looking for roads, whereas the central region's mean grey level does not appear to be significant. Further examination of the weights suggests that the network only considers the first 5 adjacent neighbours to be important. All the weights corresponding to the inputs for the 3 adjacent neighbours with the smaller adjacent perimeter were almost zero. Since 8 neighbours were considered this suggests that the restriction of only taking a maximum of 8 neighbours has not over constrained the training of the network. This is also supported by the

Figure 6: K nearest neighbour labeling (K= 6).

fact that the performance of this network is markedly better than the best performance of a K nearest neighbour clustering scheme on the same data, the results of which are given in figure 6.

Further inspection of the weights also suggests that the network has formed contextual relationships between the data; rules such as:

> A *central region* that has a small homogeneous region with a small adjacent perimeter to the left of it, and has a region to the right which is not homogeneous, is road like.

are evident in the weights. This rule may be interpreted as finding the edge of the road. The ability of the network to generate contextual rules is encouraging and implies that such networks could be used to generate rules for more conventional systems. However, these results are tentative and still require further work.

Lastly, these results suggest that a more compact network implementation may be possible with an input reduced to only those features that the weights suggest have a significant influence, and a reduced number of hidden units. Analysis of the weights suggests that 3 of the hidden units are superfluous; all the weight values to and from these units are practically zero.

Conclusion

The results presented in this article are limited and by no means conclusive. Nevertheless, they do indicate that neural networks may provide a useful tool for analysis in computer vision. Furthermore if these results can be shown to be more general then they suggest that these systems may provide an alternative and useful region classification system. However, much work still remains to be done before a definite answer can be given as to the extent to which neural network implementations may be said to "solve" the image labelling problem.

Acknowledgements

The author would like to thank: A. Page, for his invaluable help in carrying out the image segmentations that were essential for this work; A. Murton for the routines used to display the results; the members of the Research Initiative in Pattern Recognition for their helpful comments on this work, and finally P. Greenway, for reading the manuscript.

References

1. M. Fischler and O. Firschein, eds., *Readings in Computer Vision.* Morgan Kaufmann, 1987.

2. D. Ballard and C. Brown, *Computer Vision.* Prentice–Hall, 1982.

3. K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press: New York, 1972.

4. K. Fu, *Syntactic Methods in Pattern Recognition.* Academic Press: New York, 1974.

5. R. Haralick and L. Shapiro, "The consistent labelling problem: part 1.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 173–184, 1979.

6. J. Kittler, J. Illingworth, and V. Mallesh, "A study of optimisation approaches to probabilistic relaxation labelling on a 3 node 2 label problem," in *Proceedings of the Third Alvey Vision Conference*, pp. 311–318, September 1987.

7. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

8. J. Kittler 1989. private communication.

9. R. Michalski, J. Carbonell, and T. Mitchell, *Machine Learning an Artificial Intelligence Approach.* Tioga, 1983.

10. J. Quinlan, "Learning from noisy data," in *Proceedings of the International Machine Learning Workshop*, (University of Illinois), pp. 58–64, 1983.

11. D. Hutber and P. Sims, "Use of machine learning to generate rules," in *Proceedings of the Third Alvey Vision Conference*, p. 27, September 1987.

12. T. Parsons, "Disordered databases and ordered explanations," in *Proceedings of the Fourth Alvey Vision Conference*, September 1988.

13. J. Hopfield and D. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 2, pp. 141–152, 1985.

14. T. Jamison and R. Schalkoff, "Image labeling: a neural network approach," *Image and Vision Computing*, pp. 203–214, 1988.

15. D. Rummelhart, G. Hinton, and R. Williams, "Learning representations by back propagation of errors," *Nature*, vol. 323, pp. 533–536, October 1986.

16. L. Valiant, "A theory of the learnable," *Communications of the Association of Computing Mathematics*, vol. 27, 1984.

17. T. Sejnowski and C. Rosenberg, "Parallel network that learns to pronounce english text," *Complex Systems*, vol. 1, pp. 145–168, 1987.

18. W. Wright and D. Bounds, "Contextual image analysis with a neural network," Tech. Rep. RIPREP/1000/48/89, Research Initiative in Pattern Recognition, 1989.

19. D. Rumelhart and J. McClelland, *Parallel distributed processing: Explorations in the Microstructure of Cognition.* Vol. 1, Bradford Books MIT Press, 1986.

20. D. Pomerleau, "An autonomous land vehicle in a neural network," in *IEEE Conference on Neural Information Processing Systems: Natural and Synthetic*, (Denver), December 1988.

21. A. Page, "Segmentation algorithms," Tech. Rep. AOI/TR/BASR/880201, Sowerby Research Centre, British Aerospace, Bristol, 1988.

22. R. Haralick, K. Shanmugam, and I. Distein, "Textual features for image classification," *IEEE Transactions Systems Man and Cybernetics*, vol. 3, p. 620, 1973.

23. D. Bound, P. Lloyd, and B. Matthew, "A comparison of neural network and other pattern recognition approaches to the diagnosis of low back disorders," Tech. Rep., Research Initiative in Pattern Recognition, 1989. RIPRREP/1000/54/89.