Data Fusion using an MLP

D M Booth*, N A Thacker†, J E W Mayhew †and M K Pidcock ‡

The Research Initiative in Pattern Recognition,
DRA, RSRE, St. Andrews Road, Malvern, Worcs. WR14 3PS.

Abstract

A binary classification problem is solved by acting on the combined evidence of several early vision modules. Each module gives an opinion as to the identity of an individual image element, and a consensus is reached by a trained Multi-Layer Perceptron (MLP).

1 Introduction

Images of three dimensional scenes can be interpreted using various cues, such as, texture, stereopsis, shading etc. However, combining their results can be difficult. Firstly, opinions generated by a particular module must be weighted in accordance both with its reliability, and with the amount of independent information upon which the opinions were based. In addition, some fusion techniques require an opinion to be expressed as the posterior probability of a particular event, however, a more general form of expression would be desirable e.g. as an unnormalised probability, or as a measure of belief arising from an arbitrary distribution. Normally, a probabilistic output representation is favoured. Labeling is trivial, while coherence of image features can be enforced by probabilistic relaxation. The MLP will be shown to satisfy all of the above constraints.

2 Multi-Layer Perceptron

The nodes in a standard MLP are layered - an input layer, a number of hidden layers and an output layer. There is full connectivity between the nodes of adjacent layers and no connectivity between non-adjacent layers. Each processing unit (i.e. excluding input nodes) computes the weighted sum of its inputs, one of which, the bias, can be considered as a weight from a dummy unit whose output is always one. A transfer function F (normally a sigmoid) is applied to the summation. Formally,

$$F(x_j) = \frac{1}{1 + e^{-x_j}} \quad \text{and} \quad x_j = \sum_i y_i W_{ij}$$

*Seconded by DRA Electronics Division, RSRE, Malvern.
†AI Vision Research Unit, University of Sheffield, Sheffield, Yorks.
‡Dept. of Computing and Maths, Oxford Polytechnic, Headington, Oxford.

BMVC 1991 doi:10.5244/C.5.45

where x_j, y_i and W_{ij} denote the input to neuron j, the output from neuron i, and the weight between nodes i and j respectively. Training is by repeated presentation of a series of input patterns together with their corresponding target classes. The mapping from input to target space is learned by minimising

$$E = \frac{1}{2}\sum_{p=1}^{n}(\underline{y}_p - \underline{d}_p)^2$$

where $\underline{y}$ and $\underline{d}$ represent the observed and target vectors respectively, and n is the number training vectors. The learning rule is given by

$$\Delta W_{ij}(t) = -\epsilon\frac{\delta E}{\delta W_{ij}(t)} + \alpha\Delta W_{ij}(t-1)$$

where t denotes the iteration, ϵ is the learning rate, and α is the momentum.

3 Data Fusion

For a binary classification network, Dodd (1990) proved that squared error minimisation can cause outputs to approximate probabilities, provided that the output is an unbiased estimate of the probability (say a one from c coding) and that the training data is a representative sample of the pattern space. The quality of this approximation may be degraded by inadequate architecture or convergence to local minima. However, the chances of this happening can be reduced by searching for suitable network architectures and repeated training from random weight starts.

A three channel sensor system has been simulated. The image from each channel was represented by a different pair of textures bounded by a diagonal line running from the top left to the bottom right hand corner. The images were processed in turn by a texture discrimination program resulting in three grey level encoded posterior probability maps. These represent the opinions of three independent vision modules.

The MLP was trained on 240 off-diagonal image blocks, with each training vector consisting of three input probabilities, one from each channel, together with a target class label (zero or one). The trained MLP was tested on the probability images shown in Figures 1a,b and c. The types of texture used in their construction correspond with those used in the training set, though the samples are different. Note that expert "a" is least reliable, and that the labeling convention adopted by expert "b" is not in agreement with its counterparts. The MLP output is shown in Figure 1d. A 0.5 probability threshold reveals 9 classification errors in the fused segmentation, while the input channels contain 65, 35, and 31 errors respectively.

The more general problem of combining correlated measures of belief was tested by training the MLP with weighted summations of the original probability maps, whose distributions were subsequently scaled and offset by differing amounts. The test data is shown in Figures 2a, b and c. A comparison between Figures 1d and 2d shows that performance has been unaffected. The mapping has been learned and the outputs are still probabilities.

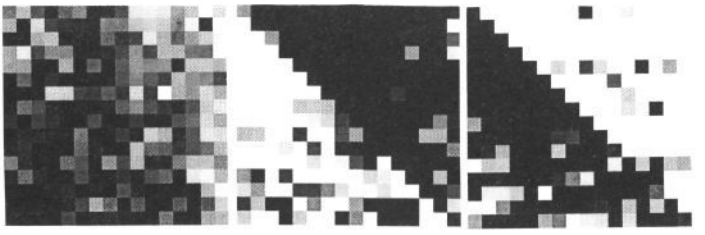

Figure 1. Independent input channels a, b and c. **Fig.1d** MLP output

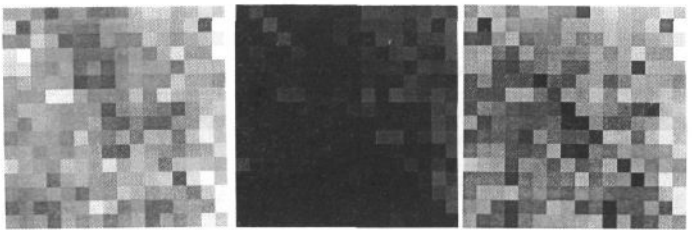

Figure 2. Correlated measures of belief a, b and c. **Fig.2d** MLP output

4 MLP Architectures

The classification error incurred when applying a trained MLP to its original training set will decrease as the number of nodes increases i.e. as the decision boundary becomes more complex. When applied to the test set, performance will increase until some optimum configuration is found, beyond which the decision boundary has modeled noise and consequently lost its ability to generalise. The experimental results from several architectures are tabulated below.

Table 1. MLP architecture performance (ten random weight starts)

image data	network config.	transfer function	training	testing	
			lowest $\sum$ error^2	$\sum$ error^2	labeling error
independent	3I-1O	linear	4.91	13.29	7.08%
	3I-1O	logistic	0.96	9.39	3.75%
	3I-2H-1O	logistic	<0.01	10.21	4.58%
correlated	3I-1O	linear	4.90	13.41	7.92%
	3I-1O	logistic	0.90	8.89	3.33%
	3I-2H-1O	logistic	0.80	10.48	5.00%

From these preliminary results it appears that the best architecture has three input nodes, no hidden nodes (though one has been masked) and one logistical output node. This happens to be the simplest network that produces outputs guaranteed to look like probabilities (i.e. in the range zero to one). Performance might be improved by adjusting the learning rate (0.1), momentum (0.9) or transfer function, however there is no guarantee that any fine tuning will be applicable to other problems.

5 Comparison with Standard Classifiers

To determine whether or not the MLP is learning a mapping that is readily determinable by conventional methods, its performance is compared with those of the nearest class mean (NCM) (with Mahalanobis distance metric) and the k nearest neighbour (KNN) classifiers. In the latter, k is odd and $\simeq \sqrt{n}$, where n is the total number of training samples. The Euclidean distance metric is used. However, to combat the effects of variability between features, and a nonorthogonal feature space, experiments have also been conducted on both standardised and orthogonalised data.

Table 2. MLP, NCM and KNN performances

image data	classification and sum squared error rates				
	NCM	KNN			MLP
	(M'obis)	raw	stand.	orthog.	(3I-1O,logistic)
independent	7.5% 18.0	5.0% 10.8	5.0% 9.9	5.4% 10.8	3.8% 9.4
correlated	7.5% 18.0	11.3% 19.9	10.0% 19.7	5.8% 11.2	3.3% 8.9

6 Conclusions

The advantage of the NCM classifier is its immunity to the effects of correlated features, however, it has the disadvantage of being parametric and of not producing classifications in the form of posterior probabilities. The KNN classifier is nonparametric and theoretically, when supplied with an infinite amount of training data, its performance is optimal. In practice, however, the KNN classifier relies on having an effective distance metric, and consequently performance drops when features exhibit widely differing variances or are highly correlated. Attempts to alleviate these problems by standardisation or principal component analysis can be effective in certain situations, but in general they cannot be relied upon. In particular, standardisation is only appropriate if the spread of feature values is due to normal random variation and not the presence of subclasses. MLPs are subject neither to the restrictions of a parametric model, nor to the problems arising from the computation of distances in feature space. Although there are situations when the KNN classifier will out perform an MLP, it is the MLPs general purpose nature that makes it appealing.

To summarise, the multi-layer perceptron has shown itself to be robust to differences in expert reliability and sensitive to correlations between experts. In addition, the outputs of a binary classification network can be interpreted, under certain conditions, as probabilities. This is not dependent on the inputs being probabilistic.

References

[1] Dodd, N. "Intensive care ward monitoring using 'default' training," *Int. Conf. Neural Networks*, Paris, 1990.