

Neural Network Training Using Multi-channel Data with Aggregate Labelling

N McGrogan*, C M Bishop†, L Tarassenko*

*Department of Engineering Science, University of Oxford

†Microsoft Research, Cambridge

Abstract

The solution of classification problems using statistical techniques requires appropriately labelled training data. In the case of multi-channel data, however, the labels may only be available in aggregate form rather than as separate labels for each individual channel. Standard techniques, using a trained model to predict each channel separately, are therefore precluded. In this paper we present a new method of training neural network classifiers from aggregate labels. This technique allows the network to learn what significant events on individual channels result in the given labels. We apply this training method to two synthetic (but, in the second case, realistic) problems and compare the results with those from a classifier trained on the accurate channel labels, which would usually not be available. On previously unseen test data for the two problems 97.75% and 99.1% of feature vectors were classified correctly. These represent reductions of only 0.5% and 0.1% from classifiers trained on accurate labels for all channels.

Introduction

The use of neural networks for classification is well documented and the requirements for training are similarly well known. One prerequisite of any training method is correctly labelled training data [5]. When a neural network is used to analyse time-varying data it is usual for the data to be temporally segmented and a label assigned to each segment [4].

In a multi-channel environment the same

segmentation process can be used on the data and classification networks applied to each channel independently. However, the available labelling may only be aggregate, i.e., for each time segment only a single label is given; the channels are not labelled individually. The label indicates the occurrence of a particular event on at least one of the recorded channels, but it cannot be taken as correct when the channels are inspected independently.

An example of this problem occurs in the detection of spikes in the human electroencephalogram (EEG) during the diagnosis of epilepsy. Typically a number of channels of data (commonly 20) are recorded and segmented temporally. A single aggregate label is assigned to each time segment indicating the presence of spikes in at least one of the channels but there is no indication of the channels in which the spikes occurred. As a result, the channels in which there is no spike are wrongly labelled. The task of relabelling each channel independently would require a significant amount of time on the part of a trained EEG technician and this is not a practical option.

In this paper we present a method which allows a neural network to be trained on the available aggregate labelling to identify what characteristic of individual channels gives rise to the observations. The trained network can subsequently be used to classify each channel individually. Our approach builds on that adopted by Keeler et al. [3] to learn the spatial segmentation of handwritten numerals.

Figure 1 shows a simple example of the labelling problem which we have described. A time sequence of features (A, B, C or D) is shown over five channels. Each time slice

C	A	C	A	A	C	B	A	D	A	D
B	C	A	A	C	C	D	A	C	B	A
D	A	A	D	A	C	A	C	D	B	B
D	A	A	B	B	A	B	C	A	D	A
A	A	C	B	D	A	D	A	A	A	B
1	0	0	1	1	0	1	0	0	1	1

Figure 1: A simple example of the aggregate labelling problem.

has been given an aggregate label according to the presence (label 1) or absence (label 0) of a particular feature in at least one of the channels. By examining the data we can identify the critical feature (in this case, B) which results in an event being signalled. Once this has been established, subsequent data could be classified on each channel independently.

We start by presenting the theoretical background to the training method and then demonstrate its use on two synthetic data sets. In each case, results are compared against a neural network classifier trained on the full labelling of each channel. After a discussion of these results we conclude with some possible areas for future developments of this method.

Theoretical Background

To learn a solution to aggregate labelled problems we use the following approach. Suppose that the available training data consists of N time slices and C channels. In this case we have a set of feature vectors \mathbf{x}_{cn} for $1 \leq c \leq C$ and $1 \leq n \leq N$. We also have an aggregate label provided by an expert for each time slice given by t_n , where $t_n \in \{0, 1\}$. This label indicates the presence of a particular event in at least one of the C channels at time slice n .

In order to be able to classify the channels independently we need to train one model per channel, $m_c(\mathbf{x}_{cn}, \mathbf{w}_c)$, where \mathbf{w}_c is a vector of adaptive parameters. The output of model m_c provides an estimate of the probability of our event being observed in channel c at a given time slice n .

If we assume that the distribution of feature vectors is independent of channel, so

we could use *the same model* for each of the channels, in which case $m(\mathbf{x}_{cn}, \mathbf{w})$ now represents the probability of our event being observed in channel c at time slice n . It is possible to use a feed-forward neural network, such as a multi-layer perceptron (MLP), as the non-linear model m , so that

$$y_{cn} = m(\mathbf{x}_{cn}, \mathbf{w}) \quad (1)$$

$$= \frac{1}{1 + \exp(-a_{cn})}, \quad (2)$$

where a_{cn} is a linear combination of ‘hidden unit’ activations.

If we also assume that the channels are independent of one another then the probability that, at a time slice n , at least one of the channels contains our event is given by p_n , where

$$p_n = 1 - \prod_{c=1}^C (1 - y_{cn}). \quad (3)$$

We can now train the network by minimising the negative log-likelihood, E [1]:

$$E = - \sum_{n=1}^N \{t_n \ln p_n + (1 - t_n) \ln(1 - p_n)\}. \quad (4)$$

The derivative of E with respect to the adaptive parameters \mathbf{w} is then given by

$$\frac{\partial E}{\partial w_l} = - \sum_{n=1}^N \frac{(t_n - p_n)}{p_n} \left(\sum_{c=1}^C y_{cn} \frac{\partial a_{cn}}{\partial w_l} \right), \quad (5)$$

and these derivatives can be used to train the MLP with a standard non-linear optimisation method.

Having developed the theory behind the training method we shall now turn to some practical examples using synthetic data sets.

Sampled Gaussians

Our first synthetic problem consists of data from four two-dimensional, radially symmetric Gaussian distributions ($\sigma = 1$) with the sampled x and y values being the features used. Figure 2 gives a plot of the distributions used.

Independent training, validation and test data sets were constructed and consisted of

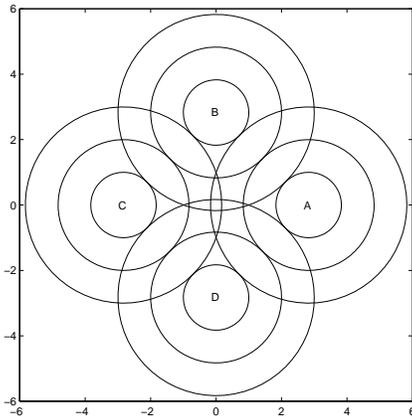


Figure 2: The four Gaussians sampled for the first synthetic data set. Circles show the σ , 2σ and 3σ circles for each distribution.

100 sampled points on each of 4 “channels”. For each channel, at time slice n , a point is sampled randomly from one of the four Gaussians (with equal priors). The labels for each channel are artificially assigned according to the following rule: the label is 1 if the point is taken from Gaussian A, 0 otherwise. The per-channel and aggregate labelling of the training data set are shown in Figure 3.

MLP classifiers with structures of the form $2-i-1$, for $2 \leq i \leq 20$, were trained using the scaled conjugate gradient optimisation method with the error function given in Equation 4. The t_n values are the aggregate labels shown in Figure 3(b). After training, the 2-11-1 network was identified as having the lowest error on the validation data set and is the model used for further testing.

Setting the decision boundary to 0.5 and applying the trained network independently to each of the channels of the test data set resulted in 9 feature vectors (2.25%) being misclassified. Figure 4 shows the results graphically.

These results can be compared with the classification accuracy of an MLP network trained on the fully labelled data (i.e., the same training data and the same training procedure, except that we now use per-channel labels t_{cn} rather than aggregate labels t_n). A 2-2-1 network gave the best generalisation performance and left 7 feature

vectors (1.75%) misclassified from the test data set.

Inter-ictal Spikes

Study of the human electroencephalogram (EEG) recorded during the investigation of epilepsy has shown that a large majority of subjects suffering from epilepsy exhibit spikes in their EEG between seizures (inter-ictal spikes) [6]. In most cases when epilepsy is confirmed by analysis of the EEG, it is on the basis of inter-ictal activity [2]. The detection of these inter-ictal spikes is therefore an important step in the diagnosis of epilepsy.

Recordings of the EEG are generally made over multiple (approximately 20) channels and the expert labelling of this data for spikes is a prime example of aggregate labelling — spikes are identified as occurring within a particular time period, but the channels in which the spikes occur is not recorded. The labelling of individual channels would be too time-consuming and so the ability to train a neural network spike detector from just the aggregate labels would be an important step forward. For this reason we have assembled another synthetic, but realistic, data set, designed to mimic the detection of EEG spikes. A five coefficient auto-regressive (AR) model of human EEG sampled at 256 Hz during wakefulness has been used to generate four channels of synthetic background EEG. Spikes of variable height and duration (between 50 and 100 ms) have been inserted into this data randomly (with a probability of 0.1 that a spike will occur in a one second time period). Figure 5 shows a short section of one channel of the signal.

Four-channel training and test data sets were constructed, each 250 seconds long. Since this is artificial data, as with the sampled Gaussian data in the first problem, the actual per-channel labels are known for both data sets.

The data is segmented into one-second time slices and the features used as input to the neural network are the mean slope and mean sharpness of the signal over each time slice. For three consecutive EEG sample values, x_{t-1} , x_t and x_{t+1} , slope and sharpness



(a) Per-channel labels.



(b) Aggregate labels.

Figure 3: Training data set labels for the sampled Gaussian problem. Four channels are shown with 100 samples per channel. Black boxes represent a labelled event (i.e., a point sampled from Gaussian A). Note how the aggregate label indicates an event when the corresponding time slice in the per-channel labelling contains at least one marked channel.



(a) Actual per-channel labels.



(b) Network output labels (decision boundary 0.5).



(c) Differences between Figures 4(a) and 4(b). White squares represent false negatives, black is false positive.

Figure 4: Test results shown in graphical form for the sampled Gaussian data set.

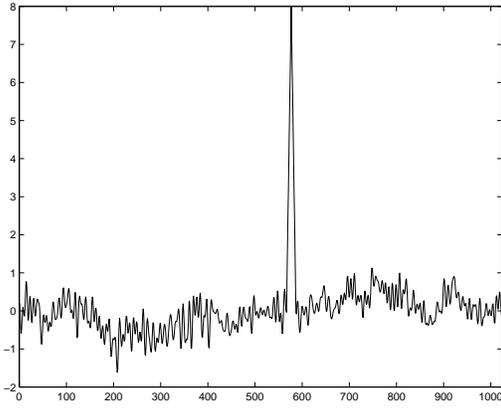


Figure 5: Four seconds of artificially generated EEG (single channel) containing a synthetic spike in the third second. The x coordinate shows the signal sample number (four seconds at 256 Hz gives a total of 1024 samples).

are defined as [6]:

$$\delta_t^0 = x_t - x_{t-1}, \quad (6)$$

$$\delta_t^1 = x_{t+1} - x_t, \quad (7)$$

$$slope_t = \frac{1}{2}(|\delta_t^0| + |\delta_t^1|), \quad (8)$$

$$sharpness_t = |\delta_t^1 - \delta_t^0|. \quad (9)$$

Figure 6 shows the distribution of slope and sharpness values for 250 seconds of training data.

The two classes in this problem are almost linearly separable. A 2-4-1 network structure was used for classification with both the aggregate and the fully labelled data. Figure 7 shows the classifications given by the network trained on aggregate labels using a 0.5 decision boundary: 9 feature vectors (0.9%) were misclassified.

For comparison a 2-4-1 network trained on the fully labelled training set (i.e., t_{cn} labels rather than t_n labels) left 8 feature vectors (0.8%) misclassified when applied to the test data.

Conclusion

Results from the application of this training method to two training sets have shown that it is possible to train a neural network classifier from aggregate labels with only a

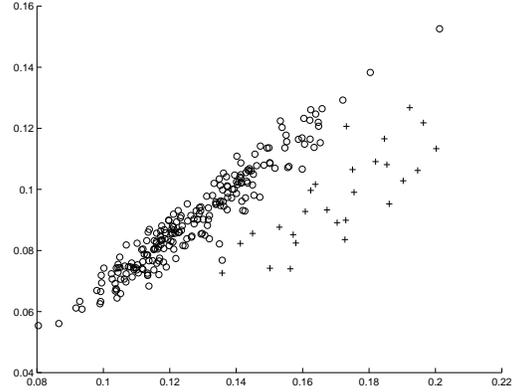


Figure 6: The distribution of slope (x axis) and sharpness (y axis) values for 250 seconds of artificial EEG. Time slices containing spikes are indicated by +.

very slight reduction in performance. This degradation is insignificant with respect to the cost (either financial or in terms of manpower) of extensive expert relabelling.

In the studies presented in this paper we have used synthetic data to show that the same model $m(\mathbf{x}_{cn}, \mathbf{w})$ can be fitted to the data \mathbf{x}_{cn} from individual channels as a result of training with an aggregate label t_n . We used synthetic (but realistic, in the case of the EEG) data in order to have the correct individual labels t_{cn} also available, so that per-channel training could be compared with the method presented in this paper. Testing using real EEG data is currently in progress and we hope to use this method to detect automatically the onset of epileptic seizures in long-term recordings for which the amount of time required for a technician to relabel the available data on a per-channel basis is considered prohibitive.

Further development of the training method is required to support different models for each channel, to allow for spatial correlation between neighbouring channels, and to move beyond two class problems by allowing multiple outputs from the classifier.

Acknowledgements

Nick McGrogan is supported by an EPSRC studentship. We gratefully acknowledge the help of our clinical collaborators at the National Hospital for Neurology and Neuro-



(a) Actual per-channel labels.



(b) Network output labels (decision boundary 0.5).



(c) Differences between Figures 7(a) and 7(b). White squares represent false negatives, black is false positive.

Figure 7: Test results for the inter-ictal spike data set shown in graphical form.

surgery, Mr Philip Allen and Dr Sheilagh Smith, with the data collection and analysis.

References

- [1] Christopher M Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [2] John R Hughes. *EEG in Clinical Practice*. Butterworth-Heinemann, second edition, 1994.
- [3] James D Keeler, David E Rumelhart, and Wee-Kheng Leow. Integrated segmentation and recognition of hand-printed numerals. In *Advances in Neural Information Processing Systems*, volume 3, pages 557–563, 1991.
- [4] J Pardey, S Roberts, and L Tarassenko. A review of parametric modelling techniques for EEG analysis. *Medical Engineering and Physics*, 18(1):2–11, January 1996.
- [5] L Tarassenko. *A Guide to Neural Computing Applications*. Arnold, 1998.
- [6] L Tarassenko, Y U Khan, and M R G Holt. Identification of inter-ictal spikes in the EEG using neural network analysis. *IEE Proc.-Sci. Meas. Technol.*, 145(6):270–278, November 1998.