Comparative Performance of Artificial Neural Networks for UV Spectral Classification

Soma Mukherjee, Ujjwal Bhattacharya and S.K. Parui

Indian Statistical Institute, 203, Barrackpore Trunk Road, Calcutta 700 035, India

and

Ranjan Gupta and Ravi K. Gulati

IUCAA, Post Bag 4, Ganeshkhind, Pune 411007, India

ABSTRACT

In this paper we present an application of an artificial neural network model based on a multi-layered backpropagation algorithm for spectral classification of UV data from the International Ultraviolet Explorer (IUE) low dispersion spectra reference atlas. The model used is similar to that of von Hippel et al. (1994), and is found to reduce the classification error as compared to the recently reported results on the same data set (Gulati et al. 1994b). The improved version of the network is much simpler in structure and the training time is reduced by a factor of almost 20. Such networks will prove very useful in efficient classification of large databases.

Subject headings: neural networks, stellar spectra, classification

1. Introduction

The importance of stellar spectral classification lies not only in identifying the individual stars, but also in the study of stellar atmospheric composition and stellar population studies. After the availability of large number of spectral databases, automated classification techniques have been employed with the distinct advantage of uniform classification in a relatively short time. Previous works in the field include employing cross correlation and height of correlation functions (Kurtz, 1984, Adorf, 1986), principal component and cluster analysis (Heck et al., 1986). Artificial neural network (ANN) techniques (Gulati et al. 1995, hereafter referred as G95; Vieira and Ponz; 1995, Gulati et al. 1994a, Gulati et al. 1994b, hereafter referred as G94a and G94b; von Hippel et al. 1994, hereafter referred as H94) are also being used for the purpose. However, the networks considered so far had a rather large size and complicated structure. This paper presents an ANN scheme with a much simplified network which not only reduces training time, but also gives a more accurate classification with reduced relative error and lesser misclassification. The neural network model that is used here is multi-layered backpropagation algorithm. The organization and the training algorithm of such a network model and the results obtained are described in the subsequent sections.

2. Multilayer Backpropagation Algorithm

A multilayer perceptron with the inclusion of hidden units helps to approximate general mappings from one finite dimensional input space to output space (Minsky, 1969). The number and the size of hidden layer(s) depends upon the complexity of the function to be approximated. The number of nodes in the output layer is equal to the dimension of the range space. The function used to obtain the activation state of a higher level node (i.e., hidden or output layer node) is called an activation function and normally the sigmoidal function is used. The most commonly used algorithm to evolve the weights of a multilayer perceptron to produce the mapping from input space to the output space, is the backpropagation algorithm of Rumelhart et al. (1986a,1986b).

In the first phase the input vectors from the training set are presented to the input layer of the network sequentially, and after every such presentation, each higher layer node calculates a weighted sum of the inputs to it (i.e., $\Sigma w_{ij} x_i$) and then subtracts a threshold θ_j (Y. H. Pao, 1989) which is considered as a connection weight between it and a node in the lower layer having node value -1. The sigmoidal function value of the thresholded weighted sum is stored into the node as its output and this in turn is used as an input to each node in the next higher layer. In other words, $o_j = 1/[1 + exp(-\Sigma w_{ij} x_i + \theta_j)]$ is the output of the j-th node of the higher layer where x_i's are the input. Thus the input pattern to the network is propagated forward. The output vector obtained in the output layer is compared with the desired output vector and an error is calculated given by $E = \Sigma(t_j - o_j)^2$, where t_j and o_j are the target and the output values respectively.

In the second phase of learning, this error is propagated back through lower layers so that the connection weights can be modified. The incremental weight change at any step is obtained by multiplying a negative constant of proportionality with the derivative of E with respect to the corresponding weight, i.e., $\Delta w_{ij} = -\eta \, (\delta E / \delta w_{ij})$, where η $(0 < \eta < 1)$ is called the learning rate. This modification rule for weights actually implements steepest descent on the error surface in the weight space. At the beginning of the learning process all the connection weights and thresholds are initialized with random values lying between -0.5 and 0.5.

The role of the learning rate η is very important in the learning process. A large value of η may result in oscillation during learning while too small a value of η may stagnate the learning process at an undesirable point in the weight space. In both cases, the proper values of the weights w_{ij} may not be available even after a reasonably long learning session. To reduce this problem, we start with a large value of η and slowly reduce it during learning to achieve satisfactory values for the weights within a reasonable time (*Quasi-Adaptive Method*).

During a successful learning session the error E gradually decreases with the number of presentations of the complete set of input vectors and in the long run it converges to a stable set of weights which results in a reasonably small value of the error. In order to avoid possible oscillation during learning, Rumelhart et al. (1986a) suggested to include a momentum term which is a proportionality constant multiple of the weight change in the previous step. This proportionality constant lying between 0 and 1 is called the momentum factor and is denoted by α .

3. The Input Data and The Network Architecture

Our input data is drawn from International Ultraviolet Explorer (IUE) low dispersion spectra reference atlas (Heck et al., 1984). For a more detailed information on the selection from the database and other details on the spectro-luminosity class coding scheme etc. the reader is referred to the recent publication G95. We have restricted ourselves from O to early F type spectra because the original catalog contains biased samples of different spectral classes with insufficient data. The luminosity classes have been restricted to super-giants, giants and dwarfs. Following this criterion, 211 spectra are selected from the catalog, of which 128 spectra spanning 75 spectro-luminosity classes (O-F) form the training set, and the remaining 83 form the test set. The selection of the training set is made by avoiding as far as possible, the cases where interstellar extinction is not high so that the interstellar reddening does not affect the continuum (G95). For a proper training of the neural network, some spectro-luminosity types which had just one example are duplicated during training session. UV spectral classification is given in terms of O, B, A, F (main classes), 0.0-9.5 sub-classes and luminosity (super-giants, giants and dwarfs). The spectra cover a range of 1213 - 3201 Å with a spectral resolution of $\sim$ 6 Å . All the spectra are normalized to unity at their respective peaks.

The spectro-luminosity classes are named in an alpha-numeric fashion as follows :
Code Number $= 1000 \times A1 + 100 \times A2 + A3$
where A1 refers to the spectral type (O - F, coded as 1 - 4) , A2 refers to the sub-spectral type (coded as 0.0-9.5) and A3 referes to the luminosity class of the star coded as 2, 5 or 8.

Each spectro-luminosity class is characterised by a set of absorption features appearing at a few known wavelengths which are useful for UV spectral classification (Heck et al., 1986, Table 1). We have used 35 such features within the spectral range of interest. This reduces the training time without losing information on the system.

The multilayer backpropagation network used here gives a mapping from 35-dimensional space (corresponding to the 35 feature input) to the spectro-luminosity class code. It has an input layer with 35 nodes, one hidden layer with 50 nodes and an output layer with just one node which renders a continuous value appropriate to the spectral class code. Thus, the network has $(36 \times 50 + 1) = 1801$ connection weights to be learnt during training. A schematic representation of the network used in the present study is shown in the Figure 1. Classification is based on the comparison of this single output code from the network to the closest spectral class codes of 75 stored catalog classes.

4. Results and Discussion

The training took approximately 9000 iterations and the trained network was then used to classify 83 stars which were not used during training. During training, both the learning rate (η) and the momentum factor (α) were gradually decreased from a high value to a low value. η was decreased from 0.9 to 0.3 and α from 0.7 to 0.1. Figure 2 shows the learning curve of the training session. Of the 83 unknown spectra, 28 of them were correctly classified in terms of spectral, sub-spectral and luminosity types and 4 were correctly classified in terms of only spectral and sub-spectral types. Further, 23 out of the rest were classified correctly in terms of spectral type, but was lying within ± 1.0 subclass. Figure 3 shows the scatter plot for network classification against the catalog classes. In case of ideal classification, all the points should lie along the diagonal line in the ANN vs. catalog classification scatter plot. Statistical analysis of the data shows a classification error of about one sub-spectral type (156 units) with respect to the diagonal line in the scatter plot without rejecting any point whereas the same is not true for the work by G94b. since they have obtained one-subclass error only after rejection of points lying outside the 3-σ. Figure 4 shows the scatter plot for the earlier work (G94b) i.e. ANN1 versus this work ANN2. The five stars which were lying outside the 3-σ in the G94b algorithm are clearly identified in this plot, thereby making sure that the earlier algorithm had misclassified these stars. However, on comparing the overall results of ANN1 and ANN2, it was found that 55% of the total test set matched in terms of classification of the spectral type and about 60% of the total set matched in terms of classification of luminosity. Further, on performing a standard students t-test, we found that there were no significant differences ($p = 0.16$) between ANN1 and ANN2 classes. This shows that the outliers of G94b are due to the ANN1 method and not due to classification itself.

Figure 5 shows a 3-D plot for classification error in luminosity (x-axis) and spectral type (y-axis) vs. percentage of total number of stars. The ideal classification would imply a delta function at the centre of the xy plane with a peak of 100%. The present study shows that 35 % of stars were classified exactly.

The present study focusses on the following points:

a. Faster Training :

In the earlier works by G95 and G94b, the network proposed for the same problem had 71 hidden nodes, 75 output nodes and $71 \times 35 + 71 \times 75 = 7810$ connection weights. In Vieira et al., 1995, the best results were obtained for a network configuration of $744 \times 120 \times 120 \times 51$. Thus our proposed network with 1801 connection weights, is much simpler and its training time is found to be faster by a factor of about 20. However, in contrast to the previous work, the current network model cannot give probability on the output classes since the mapping of input is done only on one output spectro-luminosity class.

b. Classification Accuracy:

The present network enables exact classification of 35% of the stars present in the test set of 83 unknown spectra, which is similar to the value obtained by G95 and G94b. However, the standard deviation with respect to the diagonal line as found in the study by G94b was 220 units as against 156 units in the present study. While they found that 5 stars were misclassified beyond 3-σ, *not a single star* was misclassified to the same extent in the present study. This is also evident from table 1 which shows the ANN classification of the these stars as obtained in the present study as against those obtained in the study by G94b. Clearly, the classification error is much less in the present case. This indicates the superiority of the present analysis over the previous schemes adopted in terms of both more accurate classification and reduction in relative error in classification. However, at this level of classification error, the contributions to it could be due to the quality of data since one is monitoring the line depths which are affected by the signal to noise in the data. Further, the results indicate that the classification is independent of the of the interstellar extinction effects.

It has been suggested by H94 that the number of nodes in a layer should be minimized in order to allow for more free parameters. We have repeated the training process with one layer having three nodes for 11,000 iterations. However we found that only 7% of the stars were correctly classified while 66% of the stars were misclassified beyond one sub-class. Thus, using lesser number of nodes does not improve the results in any way. The network becomes simpler at the cost of accuracy.

As can be seen from the coding scheme that we have adopted, the luminosity class falls in the least significant digit. Luminosity classification is significantly more difficult as it relies on fewer and weaker features. However, the luminosity classification achieved in the present work is *not* due to any correlation that the network might have found between the spectral types and luminosity classes. The correlation coefficients between the spectral types and the luminosity classes are found to be -0.0284 for the training set and 0.0015 for the test set. This clearly rules out the possibility that the luminosity classification could have been achieved due to classes being correlated.

In the backpropagation algorithm, large values of η and α imply that the step size with which the weights are being updated in the weight space is large. Thus large values of η and α will make the algorithm oscillate and it becomes difficult to get the optimal values of the weights. On the other hand, their small values give better estimates of the weights but the algorithm in that case may take an extremely long time to converge. To avoid these two problems, we decided to change the values of η and α interactively during training in order to achieve fine tuning of the weights in the process. Initially it is desirable that this step size be large enough to get an approximate idea of the position of the optimal weights in the weight space in a reasonably short time. Then the step size is reduced (by reducing η and α) slowly to get more and more accurate weight vectors. Finally, the results were found to be reliable and reproducible.

The present study shows the application of simple networks for higher efficiency in training as well as in error reduction, both of which is essential for sufficiently accurate spectral classifications. This proves particularly useful in case of large data bases comprising unknown spectra which may have to be classified efficiently in an unbiased manner.

Star number	Catalog	ANN1	ANN2
HD167838	2502	1405	2502
HD148605	2208	1308	2052
BD497	1708	3702	1958
HD77581	2052	2902	1505
HD40893	2005	2502	2052

Table 1: Table showing the catalog classification of the 5 misclassified stars and their classification by earlier work (ANN1) and the present work (ANN2).

REFERENCES

Adorf, H. M.1986, "Classification of low-resolution stellar spectra via template matching – a simulation study" in *Data Analysis in Astronomy II, V,* Eds. V. Di Gesú, L. Scarsi, P. Crane, J. H. Friedman, S. Levialdi (Plenum Press, Ny, USA), 61-69

Gulati,R.K., Gupta, R, Gothoskar, P. and Khobragade, S. 1994, Astrophysical Journal, 426, 340-344 (G94a)

Gulati,R.K., Gupta, R, Gothoskar, P. and Khobragade, S. 1994, Vistas in Astronomy, 38(3), 293 (G94b)

Gulati,R.K., Gupta, R, Gothoskar, P. and Khobragade, S. 1995, Bulletin of the Astronomical Society of India, In press (G95)

Heck,A., Egret, D., Jaschek,M., Jaschek,C. 1984, "IUE low-resolution spectra: a reference Atlas-Part I, normal stars", ESA SP-1052

Heck,A., Egret,D., Nobelis, P.H., Turlot,J.C. 1986, Astrophysics Space Science, 129, 223-237

von Hippel, T., Storrie-Lombardi, L.J., Storrie-Lombardi,M.C. and Irwin, M.C. 1994, MNRAS, 269,97 (H94)

Hornik, K 1989, Neural Networks, 2, 359-366

Kurtz, M.J. 1984,"Progress in automation techniques for MK classification" in *The MK Process and Stellar Classification,* Ed. R. F. Garrison (David Dunlap Observatory, Toronto, Canada), 136-152

Lippmann, R.P. 1987, IEEE ASSP Magazine, 4, 4-22

Minsky, M. and Papert,S. 1969, *Perceptrons.* (Cambridge: MIT Press)

Rumelhart,D. E., Hinton G. E. and Williams,R. J. 1986," Learning internal representations by error propagation", *Parallel distributed processing : Explorations in the microstructures of cognition, D. E. Rumelhart and J. L. McClelland Eds.* (MIT Press, Cambridge, MA), Vol. 1, 318-362

Rumelhart, D. E., Hinton G.E. and Williams, R.J. 1986, Nature, 323, 533

Pao, Yoh-Han 1989, *Adaptive Pattern Recognition and Neural Networks* (Addison-Wesley, Reading, Massachusetts)

Vieira, E. F. and Ponz, J. D. 1995, A & A Supp. Ser., 111, 393-398

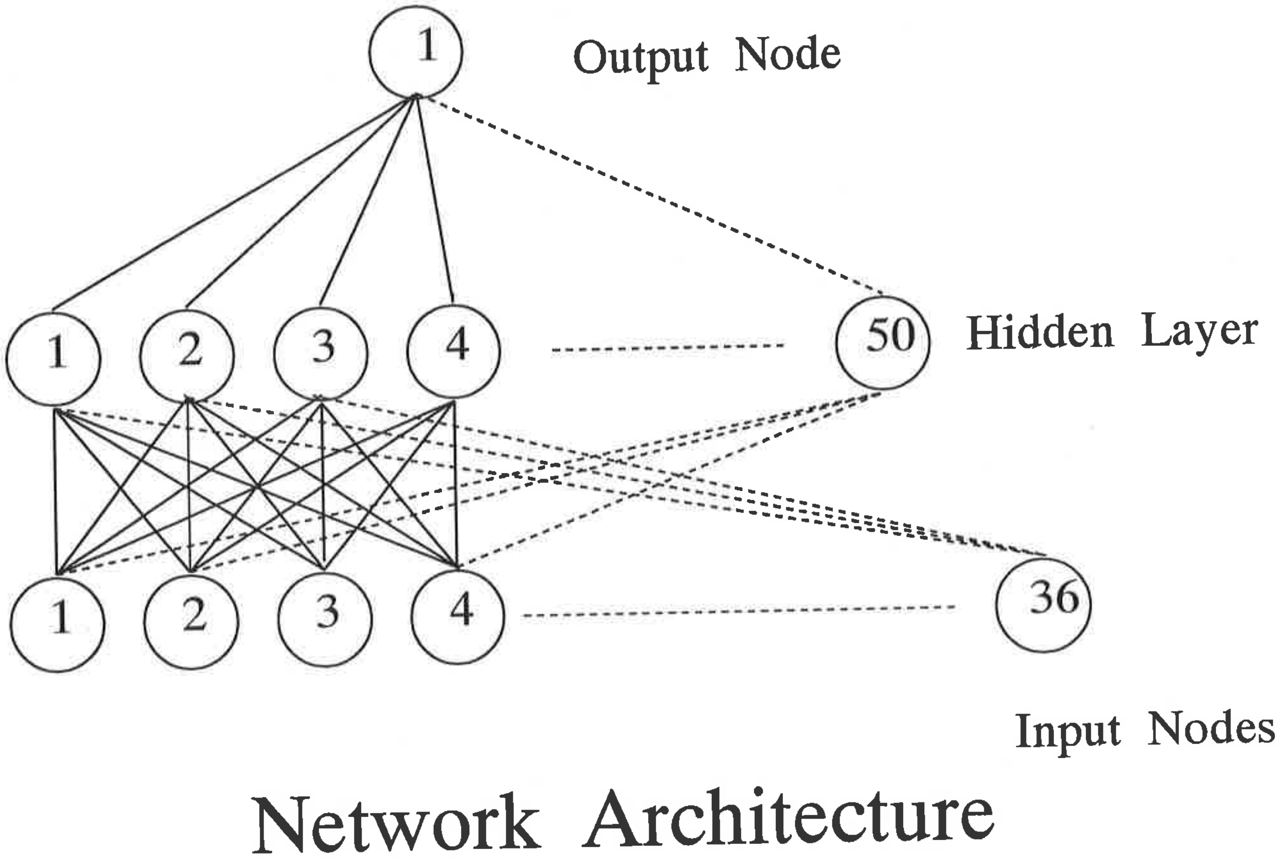

Fig. 1.— A schematic of the Multilayer Backpropagation Model used for the classification of the UV data.

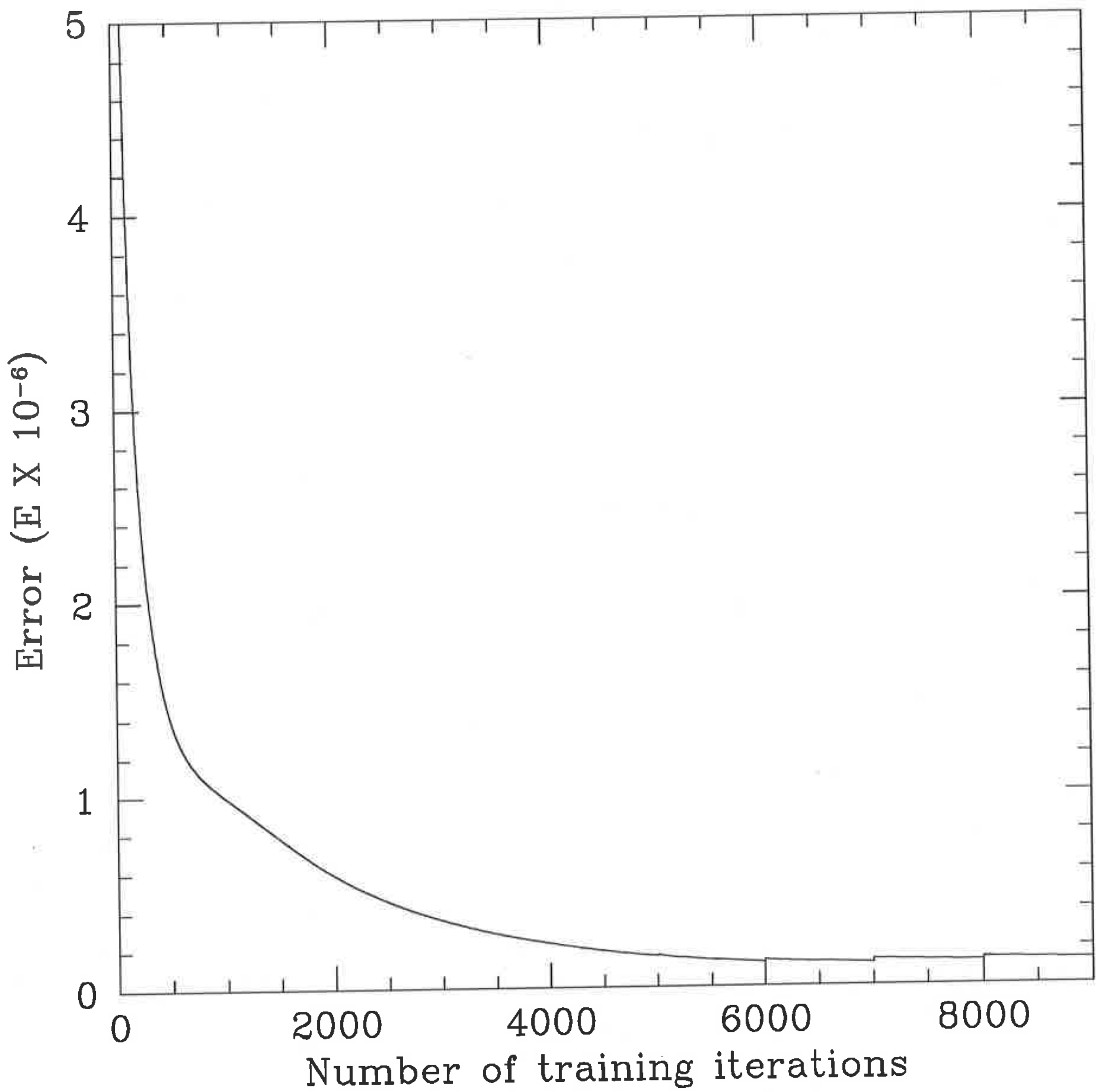

Fig. 2.— Learning curve showing the error (E) versus the number of iterations in the training session.

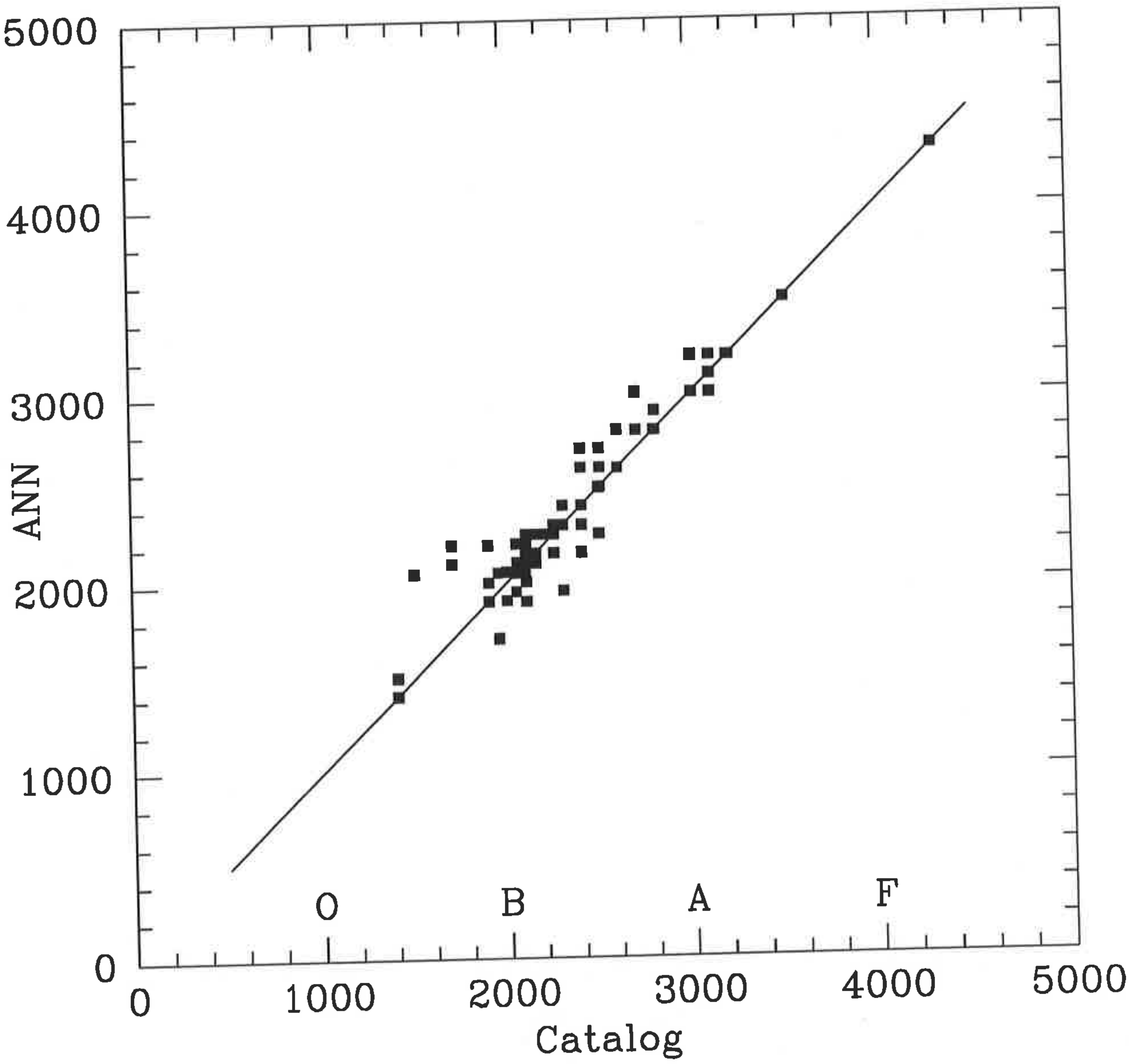

Fig. 3.— Scatter plot of classification for ANN2 (this work) vs. catalog.

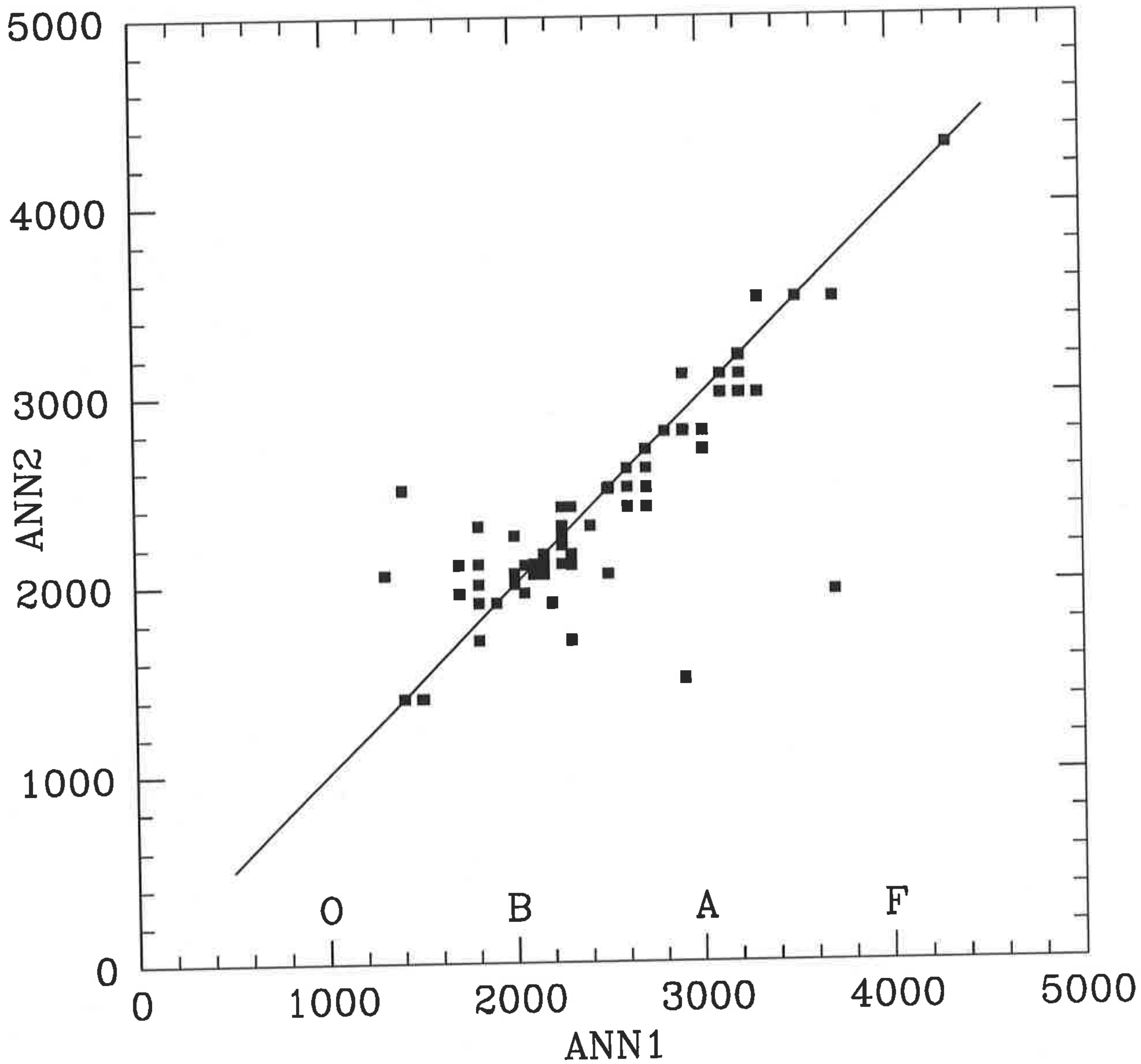

Fig. 4.— Scatter plot of classification for earlier work ANN1 and the present work ANN2.

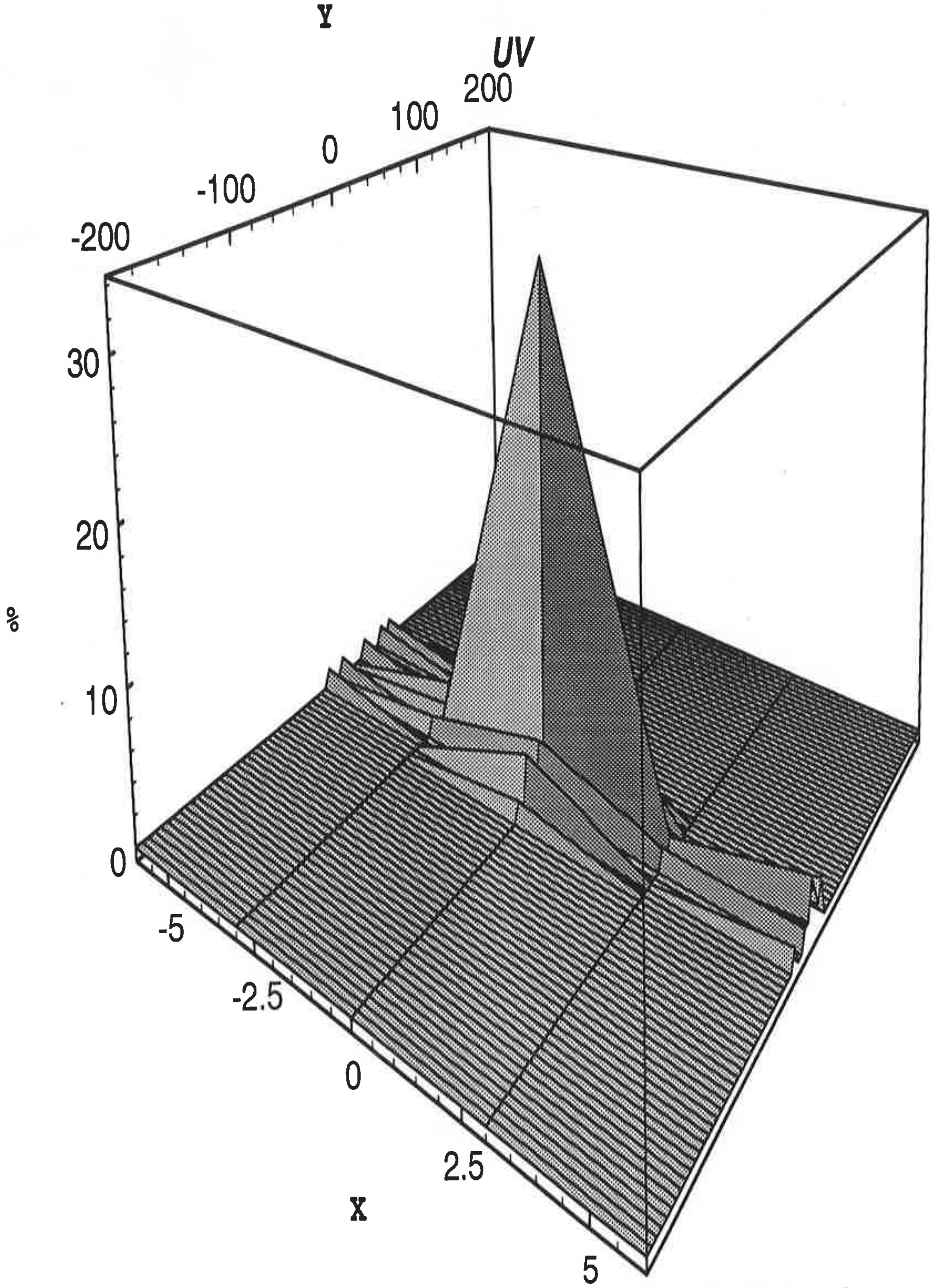

Fig. 5.— A 3D plot for classification errors in luminosity (x-axis) and spectral type (y-axis) vs. the % of total number of stars (z-axis), for this work.