NEURAL NETWORKS FOR PHOTOMETRIC REDSHIFTS EVALUATION

R. Tagliaferri [1],[2], G. Longo [3], S. Andreon [4],
S. Capozziello [5], C. Donalek [3],[2],[6] and G. Giordano [3]
[1] DMI - University of Salerno, 84081, Baronissi (SA), Italy
E-mail: robtag@unisa.it
[2] INFM, Unità di Salerno, 84081 Baronissi, Italy
[3] Department of Physical Sciences
University Federico II of Naples, I-80126, Italy
[4] INAF - Osservatorio Astronomico di Brera, Milano
[5] Dipartimento di Fisica, Università di Salerno, Baronissi, Italy
[6] Dipartimento di Matematica Applicata, University Federico II Naples, I-80126, Italy

Abstract. We present a neural network based approach to the determination of photometric redshift, which is a very important parameter to find the depth of astronomical objects in the sky. The method was tested on the Sloan Digital Sky Survey Early Data Release reaching an accuracy comparable and, in some cases, better than Spectral Energy Distribution template fitting techniques. We used Multi-Layer Perceptrons operating in a Bayesian framework to compute the parameter estimation, and a Self Organizing Map to estimate the accuracy of the results, evaluating the contamination between the classes of objects with a good prediction rate and with a poor one. In the best experiment, the implemented network reached an accuracy of 0.020 (robust error) in the range $0 < z_{phot} < 0.3$**, and of 0.022 in the range** $0 < z_{phot} < 0.5$**.**

INTRODUCTION

Redshifts number among the most crucial cosmological parameters. They, in fact, are a conventional term to denote the recession velocity of galaxies and trough the Hubble law which establishes a linear relationship between distance and recession velocity, redshifts become the most effective way to evaluate galaxy distances. The accurate knowledge of the redshifts for large samples of galaxies is therefore a pre-condition for most extragalactic and cosmological studies. Unfortunately, the measurement of accurate redshifts requires low/medium resolution spectroscopy with large telescopes, a technique which is very demanding in terms of (expensive) telescope time. An alternative (even though less accurate) approach is the evaluation of the so called

"photometric redshifts", id est the derivation of redshift estimates starting from photometric data obtained in several broad or intermediate photometric bands. This technique exploites the fact that in wide field astronomical images tens of thousands of objects are recorded at the same time and only a few exposure are required to provide the needed input data. Many different approaches have been proposed to the evaluation of photometric redshifts (see for instance [1],[2], [3], [4]). An approach, which is in the same line of the one discussed here, can be applied only to what we shall call 'mixed surveys', *id est* datasets where accurate and multiband photometric data for a large number of objects are supplemented by spectroscopic redshifts for a small but statistically significant subsample of the same objects. In this case, the spectroscopic data can be used to constrain the fit of a polynomial function mapping the photometric data [5], [6], [7].

Interpolative methods offer the great advantage that they are trained on the real Universe and do not require strong assumptions on the physics of the formation and evolution of stellar populations. Neural Networks (hereafter NNs) are known to be excellent tools for interpolating data and for extracting patterns and trends (cf. the standard textbook by Bishop [8]) and in this paper, we shall discuss the application of a set of neural tools to the determination of photometric redshifts in large "mixed surveys". The Multi Layer Perceptron (MLP) in the framework of the Bayesian learning was used to interpolate the photometric redshfit with a very good predictive result on objects until a given depth, while Self Organising Maps (SOM) were used to identify the confidence of the objects to belong to good prediction classes and to evaluate the degree of contamination of the final redshift catalogues.

NEURAL NETWORKS

NNs, over the years, have proven to be a very powerful tool capable to extract reliable information and patterns from large amounts of data even in the absence of models describing the data [8] and are finding a wide range of applications also in the astronomical community: catalogue extraction [9], star/galaxy classification [10], [9], galaxy morphology [11], [12], classification of stellar spectra [13], [14], [15], data quality control and data mining [16].

The AstroMining software [17] is a package written in the Matlab environment to perform a large number of data mining and knowledge discovery tasks, both supervised and unsupervised, in large multiparametric astronomical datasets. The package relies also on the Matlab "Neural Network", the "SOM" [18] and the "Netlab" [19] toolboxes.

Using AstroMining, via interactive interfaces, it is possible to perform a large number of operations: i) manipulation of the input data sets; ii) selection of relevant parameters; iii) selection of the type of neural architecture; iv) selection of the training validation and test sets; v) etc. The package is completed by a large set of visualization and statistical tools which allow to estimate the reliability of the results and the performances of the network.

The user friendly interface and the generality of the package allow both a wide range of applications and the easy execution of experiments (more details on other aspects of the AstroMining tool which are not relevant to the present work may be found in [16].

The Multi Layer Perceptron - MLP

A NN is usually structured into an input layer of neurons, one or more hidden layers and one output layer. Neurons belonging to adjacent layers are usually fully connected and the various types and architectures are identified both by the different topologies adopted for the connections and by the choice of the activation function. Such networks are generally called Multi Layer Perceptron (MLP; [8]) when the activation functions are sigmoidal or linear. Due to its interpolation capabilities, the MLP is one of the most widely used neural architectures. We implemented an MLP with one hidden layer and n input neurons, where n is the number of parameters selected by the user as input in each experiment.

It is possible to train NN's also in the Bayesian framework, which allows to find the more efficient among a population of NN's differing in the hyperparameters controlling the learning of the network [8], in the number of hidden nodes, etc.

The Bayesian method allows the values of the regularization coefficients to be selected using only the training set, without the need for a validation set.

The implementation of a Bayesian framework requires several steps: initialization of weights and hyperparameters; training the network via a non linear optimization algorithm in order to minimize the total error function. Every few cycles of the algorithm, the hyperparameters are re-estimated and eventually the cycles are reiterated.

The Self Organizing Maps

The SOM algorithm [20] combines a competitive learning principle with a topological structuring of nodes such that adjacent nodes tend to have similar weight vectors. The training is unsupervised and it is entirely data-driven and the neurons of the map compete with each other [18]. These networks are Self Organizing in that, after training, nodes tend to attain weight vectors that capture the characteristics of the input vector space. SOM allows an approximation of the probability density function of the training data, the derivation of prototype vectors best describing the data, and a highly visualized and user friendly approach to the investigation of the data. This property turns SOM into an ideal tool for KDD and expecially for its exploratory phase: data mining [18].

During the training phase, one sample vector $\mathbf{x}$ from the input data set is randomly chosen and a similarity measure is calculated between it and all the weight vectors of the map. The Best-Matching Unit (BMU), denoted

as c, is the unit with weight vector having the greatest similarity with the input sample $\mathbf{x}$. The similarity is usually defined by means of a distance measure, typically an Euclidean distance. After finding the BMU, the weight vectors of the SOM are updated. The training is usually performed into two phases. In the first phase, relatively large initial α value and neighborhood radius are used. In the second phase both the α value and the neighborhood are small from the beginning. This procedure corresponds to first tuning the SOM approximately to the same space as the input data and then fine-tuning the map. The SOM toolbox [18] includes the tools for the visualization and analysis of SOM. Another advantage of SOM is that it is relatively easy to label individual data, *id est* to identify which neuron is activated by a given input vector. The utility of these properties of the SOM will become clear in the next paragraphs.

APPLICATION TO THE SDSS-EDR DATA

A preliminary data release (Early Data Release or EDR) of the SDSS was made available to the public in 2001 [21]. This data sets provide photometric, astrometric and morphological data for an estimated 16 millions of objects in two fields: an Equatorial 2^{circ} wide strip of constant declination centered around δ=0 and a rectangular patch overlapping with the SIRTF First Look Survey.

The EDR provides also spectroscopic redshifts for a little more than 50.000 galaxies distributed over a large redshift range and is therefore representative of the type of data which will be produced by the next generation of large scale surveys. In order to build the training, validation and test sets, we first extracted from the SDSS-EDR a set of parameters (u, g, r, i, z, both total and petrosian magnitudes, petrosian radii, 50% and 90% petrosian flux levels, surface brightness and extinction coefficients, [21] for all galaxies in the spectroscopic sample.

In this data set, redshifts are distributed in a very dishomogeneous way over the range $0 - 7.0$ (93% of the objects have $z < 0.7$).

It needs to be stressed that the highly dishomogeneous distribution of the objects in the redshift space implies that the density of the training points dramatically decreases for increasing redshifts, and that: i) unless special care is paid to the construction of the training set, all networks will tend to perform much better in the range where the density of the training points is higher; ii) the application to the photometric data set will be strongly contaminated by the spurious determinations.

The photometric redshift evaluation

The experiments were performed using the NNs in the Matlab and Netlab Toolboxes, with and without the Bayesian framework. All NNs had only one hidden layer and the experiments were performed varying the number of the

Table 1: Column 1: higher accepted spectroscopic redshift for objects in the training set; column 2: input parameters used in the experiment; column 3: number of neurons in the hidden layer; column 4: robust errors evaluated on the test set; column 5: number of objects used in each of the training, validation and test set.

Range	**parameters**	**h.n.**	**err.**	**obj.s**
$z < 0.3$	r, u-g, g-r, r-i, i-z	18	0.029	12000
$z < 0.5$	r, u-g, g-r, r-i, i-z	18	0.031	12430
$z < 0.7$	r, u-g, g-r, r-i, i-z	18	0.033	12687
$z < 0.3$	r, u-g, g-r, r-i, i-z, radius	18	0.025	12022
$z < 0.5$	r, u-g, g-r, r-i, i-z, radius	18	0.026	12581
$z < 0.7$	r, u-g, g-r, r-i, i-z, radius	18	0.031	12689
$z < 0.3$	r, u-g, g-r, r-i, i-z, radius, p. fluxes, s. brightness	22	0.020	12015
$z < 0.5$	r, u-g, g-r, r-i, i-z, radius, p. fluxes, s. brightness	22	0.022	12536
$z < 0.7$	r, u-g, g-r, r-i, i-z, radius, p. fluxes, s. brightness	22	0.025	12680

input parameters and of the hidden units. Extensive experiments lead us to conclude that the Bayesian framework provides better generalization capabilities with a lower risk of overfitting, and that an optimal compromise between speed and accuracy is achieved with a maximum of 22 hidden neurons and 10 Bayesian cycles.

In Table 1, we summarize some of the results obtained from the experiments and, in Figure 1, we compare the spectroscopic redshifts versus the photometric redshifts derived for the test set objects in the best experiment.

Contamination of the catalogues

In practical applications, one of the most important problems to solve is the evaluation of the contamination of the final photometric redshift catalogues or, in other words, the evaluation of the number of objects which are erroneously attributed a z_{phot} significantly (accordingly to some arbitrarily defined threshold) different from the unknown z_{spec}. This problem is usually approached by means of extensive simulations. The problem of contamination is even more relevant in the case of NNs based methods, since NNs are necessarily trained only in a limited range of redshifts and, when applied to the real data, they will produce misleading results for most (if not all) objects which "in the real word" have redshifts falling outside the training range. This behaviour of the NNs is once more due to the fact that while being good interpolation tools, they have very little, if any, extrapolation capabilities.

Moreover, in the SDSS-EDR spectroscopic sample, over a total of 54,008 objects having $z > 0$, only 88%, 91% and 93% have redshift z lower than, respectively than 0.3, 0.5 and 0.7. To train the network on objects falling in the above ranges implies, respectively, a minimum fraction of 12%, 9% and 7% of objects in the photometric data set having wrong estimates of the photometric redshift.

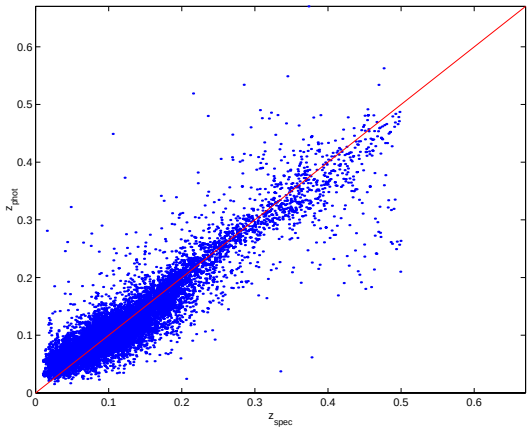

Figure 1: Photometric versus spectroscopic redshifts obtained with a Bayesian MLP with 2 optimization cycles, 50 learning epochs of quasi-Newton algorithm and 5 inner epochs for hyperparameter optimization. Hyperparameters were initialized at α=0.001 and β=50

An accurate estimate of the contamination may be obtained using unsupervised SOM clustering techniques over the training set.

In Figure 2 we show the position of the BMU as a function of the redshift bin. Each exagon represents a neuron and the figures inside it give the number of input vectors (in a given range) which have that neuron as BMU. It is clearly visible that low redshift objects ($z < 0.5$) tend to activate neurons in the lower right part of the map, intermediate redshift ones ($0.5 < z < 0.7$) neurons in the lower left part and, finally, objects with redshift higher than 0.7 activate only the neurons in the upper left corner. The labeling of the neurons (shown in the upper left map) was done using the training and validation data sets in order to avoid overfitting, while the confidence regions were evaluated on the test set. In Figure xx we split the objects into two groups: in the first one we included all objects with $z < 0.5$ and in the second one all those with $z >= .5$. Each cell is labeled as class 1 or class 2 accordingly to the relative distribution of input vectors belonging to a given group which activate that cell. Therefore, test set may be used to map the neurons in the equivalent of confidence regions and to evaluate the degree of contamination to be expected in any given redshift bin. Conversely, when the network is applied to real data, the same confidence regions may be used to evaluate whether a photometric redshift correspondent to a given input vector may be trusted upon or not.

The above derived topology of the network is also crucial since it allows to derive the amount of contamination. In order to understand how this may be achieved, let us take the above mentioned NN, and consider the case of objects which are attributed a redshifts $z_{phot} < 0.5$. This prediction has a high degree of reliability only if the input vector activates a node in the central or right portions of the map. Vector producing a redshift $z_{phot} < 0.5$ but activating a node falling in the upper left corner of the map are likely to be misclassified.

Table 2: Confusion matrix for the three classes described in the text.

	objects	Class I	Class II	Class III
Class I	9017	95.4%	2.96%	1.6%
Class II	419	6.4%	76.6%	16.9%
Class III	823	3.8%	2.1%	94.2%

In our experiment, out of 9270 objects with $z_{phot} < 0.5$, only 39 (*id est*, 0.4% of the sample) have discordant spectroscopic redshift. A confusion matrix helps in better quantifying the quality of the results. In Table 3.2, we give the confusion (or, in this case, 'contamination') matrix obtained dividing the data into three classes accordingly to their spectroscopic redshifts, namely class I: $0 < z < 0.3$, class II: $0.3 < z < 0.5$, class III: $z > 0.5$. The elements on the diagonal are the correct classification rates, while the other elements give the fraction of objects belonging to a given class which have been erroneously classified into another class. Furthermore, in the redshift range $(0, 0.3)$, 95.4% of the objects are correctly identified and only 4.6% is attributed a wrong redshift estimate. In total, 94.2% are correctly classified. By taking into account only the redshift range $0 < z < 0.5$, this percentage becomes 97.3%. From the confusion matrix, we can therefore derive a completeness of 97.8% and a contamination of about 0.5%.

SUMMARY AND CONCLUSIONS

The application of NNs to mixed data, *id est* spectroscopic and photometric surveys, allows to derive photometric redshifts over a wide range of redshifts with an accuracy equal if not better to that of more traditional techniques.

The method makes use of two different neural tools: i) an MLP in Bayesian framework used to estimate the photometric redshifts; ii) an unsupervised SOM used to derive the completeness and the contamination of the final catalogues. On the SDSS-EDR, the best result (robust error = 0.020) was obtained by a MLP with 1 hidden layer of 22 neurons, after 10 Bayesian cycles.

The method fully exploits the wealth of data provided by the new digital surveys since it allows to take into account not only the fluxes, but also the morphological and photometric parameters.

The proposed method will be particularly effective in mixed surveys, *id est*, in surveys were a large amount of multiband photometric data is complemented by a small subset of objects having also spectroscopic redshifts.

REFERENCES

[1] Koo D.C., 1999, astro-ph/9907273

[2] Fernandez-Soto A., Lanzetta K.A., Chen H.W., Pascarelle S.M., Yakate N., 2001, ApJSS, 135, 41

[3] Massarotti M., Iovino A., Buzzoni A, 2001a, AA, 368, 74

[4] Massarotti M., Iovino A., Buzzoni A., Valls-Gabaud D., 2001b, AA, 380, 425
[5] Connolly A.J., Csabai I., Szalay A.S., Koo D.C., Kron R.G., Munn J.A., 1995, AJ, 110, 2655
[6] Wang Y., Bachall N., Turner E.L., 1998, AJ, 116, 2081
[7] Brunner R.J., Szalay A.S., Connolly A.J., 2000, ApJ, 541, 527
[8] Bishop C.M., 1995, Neural Networks for Pattern Recognition, Oxford University Press
[9] Andreon S., Gargiulo G., Longo G., Tagliaferri R., Capuano N., 2000, MNRAS, 319, 700
[10] Bertin E., Arnout S., 1996, AAS, 117, 393
[11] Storrie-Lombardi M.C., Lahav O., Sodré L. jr, Storrie-Lombardi L.J., 1992, MNRAS, 259, 8
[12] Lahav O., Naim A., Sodré L. jr., Storrie-Lombardi M.C., 1996, MNRAS, 283, 207
[13] Bailer-Jones C.A.L., Irwin M., von Hippel T., 1998, MNRAS, 298, 361
[14] Allende Prieto C., Rebolo R., Lopez R.J.G., Serra-Ricart M., Beers T.C., Rossi S., Bonifacio P., Molaro P., 2000, AJ, 120, 1516
[15] Weaver W.B., 2000, ApJ, 541, 298
[16] Tagliaferri R., Longo G., Milano L., Acernese F., Barone F., Ciaramella A., De Rosa R., Donalek C., Eleuteri A., Raiconi G., Sessa S., Staiano A., Volpicelli A., 2003, Neural Networks. Special Issue on Applications of Neural Networks to Astrophysics and Geosciences, R. Tagliaferri, G. Longo, D'Argenio B. eds.
[17] Longo G., Tagliaferri R., Sessa S., Ortiz P, Capaccioli M., Ciaramella A., Donalek C., Raiconi G., Staiano A., Volpicelli A., in Astronomical Data Analysis, J.L. Stark and F. Murtagh eds., SPIE n. 4447, p.61
[18] Vesanto J., 1997, Ph.D. Thesis, Helsinky University of Technology
[19] Nabney I.T., Bishop C.M., 1998, Netlab: Neural Network Matlab Toolbox, Aston University
[20] Kohonen T., 1995, Self-Organizing Maps, Springer:Berlin-Heidelberg
[21] Stoughton C., Lupton R.H., Bernardi M., Blanton M. R., et al., 2001, AJ, 123, 485

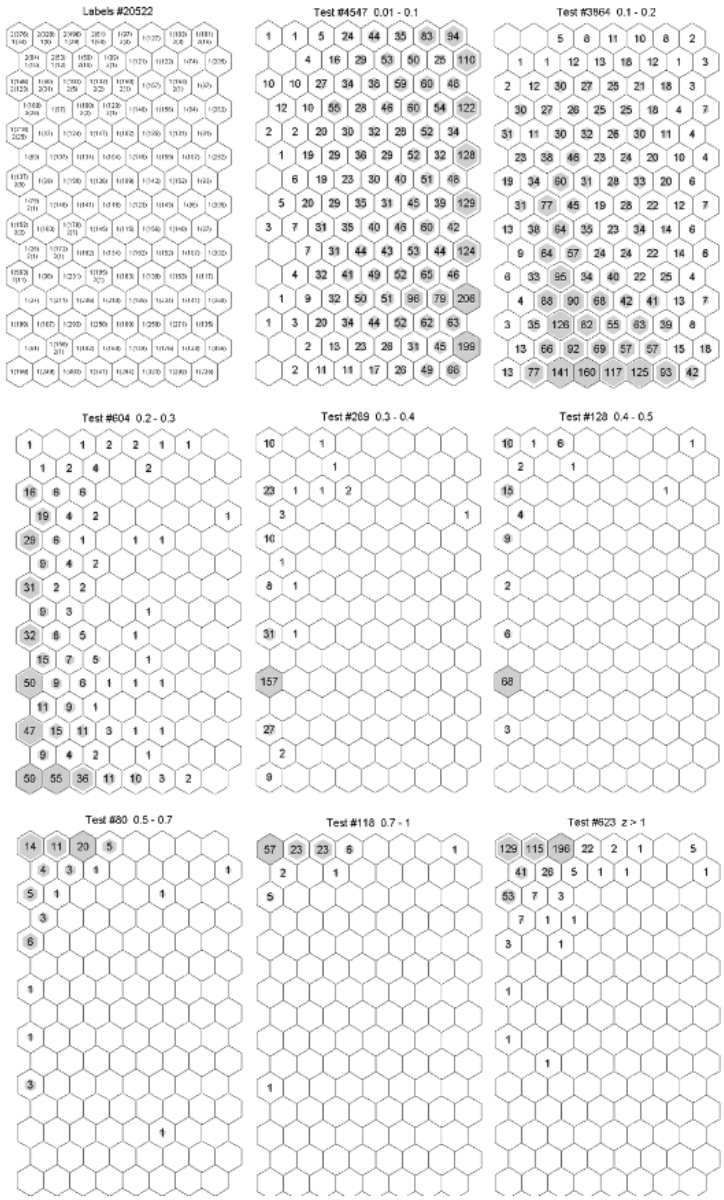

Figure 2: Maps of the neuron activated by the input data set. Exagons represent the NN nodes. In the map in the upper left corner, for a given node, the figures n(m) can be read as follows: n is the class (n=1 meaning $z < 0.5$ and n=2 meaning $z > 0.5$) and m is the number of input vector of the correspondent class which have activated that node. This map was produced using the training and validation data sets. The other maps, produced each in a different redshift bin, indicate how many input vector from the test data set activated a given node.