Research Signpost
37/661 (2), Fort P.O., Trivandrum-695 023, Kerala, India

Recent Res. Devel. Astrophys., 2(2005): ISBN: 81-7736-295-X

Applications of neural networks in astronomy and astroparticle physics

**Angelo Ciaramella[1], Ciro Donalek[2,3,4], Antonino Staiano[1]
Michelangelo Ambrosio[3], Carla Aramo[2,3], Piero Benvenuti[5]
Giuseppe Longo[2,3], Leopoldo Milano[2,3], Giancarlo Raiconi[1,6]
Roberto Tagliaferri[1,6] and Alfredo Volpicelli[7]**
[1]Department of Mathematics and Informatics - DMI, University of Salerno Fisciano, Italy; [2]Department of Physical Sciences, University Federico II via Cinthia, Napoli, Italy; [3]INFN- Sezione di Napoli, via Cinthia, Napoli, Italy
[4]Department of Mathematics and Applications, University Federico II via Cinthia , Napoli, Italy; [5]INAF - Sede Centrale via del Parco Mellini, Roma Italy; [6]INFM - Section of Salerno, via S. Allende, Baronissi, Italy
[7]INAF - Osservatorio Astronomico di Capodimonte, via Moiariello 16 80131 Napoli, Italy

Abstract

Neural Networks try to emulate the behavior of human brain in performing simple classification and pattern recognition tasks. The huge explosion in both quality and quantity of data experienced in the last decade by both astronomy and high energy physics, has stimulated the search for artificial intelligence tools capable to help the scientist in both the routine work (such as prereduction and analysis) and in the

Correspondence/Reprint request: Dr. Giuseppe Longo, Department of Physical Sciences, University Federico II via Cinthia, Napoli, Italy. E-mail: Giuseppe.Longo@na.infn.it

visualization and interpretation of the data. In this review we outline some recent developments in the field of neural networks, together with their main applications to a variety of problems.

1. Introduction

Artificial Neural Networks (NNs) were originally introduced as simplified models of the brain (processing nodes instead of neurons, multiple connections instead of dendrites and axons). Such a brain analogy, however, may mislead unexperienced users who tend to overlook the fine details requested by the proper application of NNs, thus obtaining discouraging results. In fact, as stressed by Bailer et al. [1], even though it is likely that the logic of the processing of the signals in both the brain and NNs is very similar, the scale is enormously different: the human brain consists of 10^{11} neurons each connected to many others with a complex topology for a minimum of 10^{14} synaptic connections, while even the most complex NN consists of a few hundred neurons and achieves a level of complexity at least 10 orders of magnitude smaller. As a matter of fact, however, in most individual processes, the brain uses only a very small subset of its resources and therefore, in spite of the lack of complexity, some specific tasks may be emulated by NNs. In fact, they offer the advantage of being objective, relatively fast and, what is more relevant, not necessarily biased by some human limitations such as, for instance, the lack of capability in dealing with high dimensionality spaces. This last feature makes NNs appealing to all these fields where there is the need to organize and extract the relevant information from multiparametric spaces of high dimensionality. This is particularly true for astronomy, where the recent technological advances have produced a true explosion in both the quality and the amount of data available to any user [2]. Conservative predictions lead in fact to expect that in a few years, more than 10 TB of data will be acquired worldwide every night and, due to the ongoing efforts for the implementation of the International Virtual Observatory (IVO), most of these data will become available to the community via the network (cf. [3], [4] and the references therein). These huge and heterogeneous data sets will open possibilities which so far are just unthinkable, but it is already clear that their scientific exploitation will require the implementation of automatic tools capable to perform a large fraction of the routine data reduction, data mining and data analysis work.

In its most general definition, a NN is a software which learns about a problem through relationship which are intrinsic to the data rather than through a set of predetermined rules. 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. The values of the functions associated to the connections are called “weights” and the whole game of NN's is in the fact that, in order for the network to yield appropriate outputs for given inputs, the weights must be set to a suitable combination of values. The way this is obtained leads to the first important difference among modes of operations, namely between “supervised” and “unsupervised” methods. In supervised methods, in order to teach to the NN how to provide the correct output, the user needs to know the correct output value for a fair subsample of the input data. This set is further divided in other three subsets named, respectively, training, validation and test sets. The first one is used to fine tune the weights, the second one to check whether the network has achieved an acceptable generalization capability and, finally, the third subset is used to evaluate the performances.

In unsupervised methods, instead, the input data are clustered on the basis of their statistical properties only. Whether the obtained clusters are or are not significant to a specific problem and which meaning has to be attributed to a given cluster, is not obvious and requires an additional phase, the so called “labeling”. The labeling requires that the user knows the characteristics of a small sample of input vectors (labeled set). It needs to be stressed that the labeled set needs to be much less conspicuous than the training, validation and test sets necessary to supervised methods.

A further distinction among different NN's can be based on the way the information propagates across the network: either feedforward (i.e. the information propagates only from the layer K to the layer $K + 1$), or recurrent (i.e. the information may propagate in loops).

As we shall see in what follows, the optimal choice of the architecture of the network and of its operating modes depends strongly on the intrinsic nature of the specific problem to be solved and, since no well defined recipe exists, the user has often to rely on a trial and error procedure. It has therefore to be stressed that, in order to be effective, all neural techniques require a lengthy procedure to be selected and optimized, and an extensive testing to evaluate their robustness against noise and inaccuracy of the input data. This also restricts their application to those data intensive or computational intensive problems where the work required by the implementation of a network may prove to be advantageous with respect to more traditional methods.

1.1 A short summary of astronomical applications

Some recent reviews of specific aspects of the applications of NN's to astronomical problems can be found in [5], [6] and [7]. Historically speaking, the first attempts to use NNs on astronomical data were meant to separate stars from galaxies or to classify galaxy morphologies or spectral features [8], [9].

While the problem of star/galaxy classification seems to have been satisfactorily answered [10], the other two problems, due to their intrinsic complexity, are still matter of experiments (see for instance, [11], [12], [13], [14], and [15], [16], [17], for recent applications to galaxy morphology and spectral classification, respectively). NN's have also been applied to planetary studies [18], [19]; to the study and prediction of solar activity and phenomena (cf. [20], [21], [22], [23], [24]), to the study of the interplanetary magnetic field [25], and to stellar astrophysics [26], [27], [28].

Other fields of application include: time series analysis (cf. [29], [30]), and the identification and characterization of peculiar objects such as QSO's, ultraluminous IR galaxies, and Gamma Ray Bursters [31]; the determination of photometric redshifts (cf. [32], [33]), the noise removal in pixel lensing data [34], the decomposition of multifrequency data simulated for the Planck mission [35], the search for galaxy clusters [36].

Still in its infancy, is the use of NN's for the analysis of the data collected by the new generation of instruments for astroparticle physics such as, for instance, the Solar energetic proton events [37] the cosmic ray telescopes AUGER [38], [39] and ARGO [40], [41]; the gamma ray Cherenkhov telescope [42], [43], the VIRGO gravitational waves interferometer [44], [45] and even for the search of the Higgs boson [46].

Due to the extreme richness of the field, in what follows we shall concentrate only on some of the applications which have been addressed by our group in the framework of the AstroNeural collaboration which was started in 1999 as a joint project of the Department of Mathematics and Informatics at the University of Salerno and of the Department of Physics at the University Federico II of Napoli. The tools produced by the AstroNeural collaboration are integrated within the AstroMining package: a set of modules written in MatLab and C++.

This paper is structured as follows. In Sect. 2 we shortly outline the mathematical background of the used NN models: the Multi Layer Perceptron (par. 2.1), the Self Organizing Maps (par. 2.2) and the Neural Networks for non-linears Principal Component Analysis (par. 2.3). In Sect. 3 we shall describe the application of the above quoted neural models to: photometric redshifts evaluation (par. 3.1), clustering of photometric data (par. 3.2), search for periodicities in light curves (par. 3.3), and analysis of simulated cosmic ray air showers (par. 3.4). Finally, in Section 4 we shortly summarize some future developments.

2. Some neural models used in astronomy

2.1 The Multi Layer Perceptron

The simple neural model outlined in the introduction is usually said to be a Multi Layer Perceptron (MLP; [47]) where the activation functions are sigmoidal or linear. (Fig. 1).

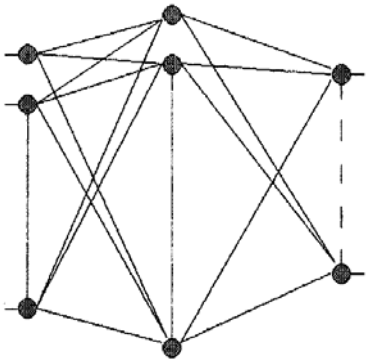

Figure 1. Layout of the Multi Layer Perceptron.

The MLP is suited to a wide range of applications, such as pattern recognition, prediction, system identification, etc.. An MLP network comprises a number of identical units organized in layers, with those on one layer connected to those on the next layer (except for the last layer or output layer).

The output of the j–th hidden unit is obtained first by forming a weighted linear combination of the d input values, and then by adding a bias to give:

$$z_j = f\left(\sum_{i=0}^{d} w_{j,i}^{(1)} x_i\right) \tag{1}$$

where d is the number of the input, $w_{j,i}^{(1)}$ denotes a weight in the first layer (from input i to hidden unit j). Note that $w_{j,0}^{(1)}$ denotes the bias for the hidden unit j, and f is an activation function such as the continuous sigmoidal function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{2}$$

The outputs of the network are obtained by transforming the activation of the hidden units using a second layer of processing elements:

$$y_k = g\left(\sum_{j=0}^{M} w_{k,j}^{(2)} z_i\right) \tag{3}$$

Where M is the number of hidden units, $w_{k,j}^{(2)}$ denotes a weight in the first layer (from hidden unit j to output unit k). Note that $w_{k,0}^{(2)}$ denotes the bias for the output unit k, and g is an activation function of the output units which does not need to be the same function as for the hidden units. MLP are tipically trained using a supervised training algorithm known as 'back propagation' which works as follows: we give to the input neurons the first pattern and then the net produces an output. The difference (error) between the desired output (target value) and the produced output is computed and the weights are changed in order to minimize it. These operations are repeated for each input pattern until the mean square error of the system is minimized. Given the p-th pattern in input, a classical error function E_p (called sum-of-squares) is:

$$E_p = \frac{1}{2}\sum_j (t_j^p - y_j^p)^2 \tag{4}$$

where t_j^p is the p-th desired output value and y_j^p is the output of the corresponding neuron. Due to its interpolation capabilities (universal approximation), the MLP is one of the most widely used neural architectures. The MLP can be trained also using probabilistic techniques such as the Bayesian learning framework that offers several advantages over classical ones [47]: i) it cannot overfit the data; ii) it is automatically regularized; iii) the uncertainty in the prediction can be estimated.

2.2 The Self Organizing Maps

The SOM algorithm [48] 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 [49]. 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 the approximation of the probability density function of the data in the training set (id est prototype vectors best describing the data), and a highly visualized approach to the understanding of the statistical characteristics of the data. In a crude approximation, a SOM is composed by neurons located on a regular, usually 1 or 2-dimensional, grid.
Each neuron i of the SOM may be represented as an n-dimensional weight:

$$\mathbf{m}_i = [m_{i_1}, \ldots, m_{i_n}]^T \tag{5}$$

where *n* is the dimension of the input vectors. 3–D or higher dimensional grids are not commonly used since in this case the visualization of the outputs becomes problematic.

In most implementations, SOM's neurons are connected to the adjacent ones by a neighborhood relation which dictates the structure of the map. In the 2-dimensional case, the neurons of the map can be arranged either on a rectangular or a hexagonal lattice and the total number of neurons determines the granularity of the resulting mapping, thus affecting the accuracy and the generalization capability of the SOM.

The use of SOM as data mining or data visualization tools requires several logical steps: the construction and the normalization of the data set, the initialization and the training of the map, the visualization and the analysis of the results. In the SOM's, the topological relations and the number of neurons are fixed from the beginning via a trial and error procedure, with the neighborhood size controlling the smoothness and generalization of the mapping. The initialization consists in providing the initial weights to the neurons and, even though the SOM are robust with respect to the initial choice, a proper initialization usually allows faster convergence.

For instance, the AstroMining package allows three different types of initialization procedures: *random initialization*, where the weight vectors are initialized with small random values; *sample initialization*, where the weight vectors are initialized with random samples drawn from the input data set; *linear initialization*, where the weight vectors are initialized in an orderly fashion along the linear subspace spanned by the two principal eigenvectors of the input data set. The corresponding eigenvectors are then calculated using the Gram-Schmidt procedure detailed in [49]. The initialization is followed by the training phase.

In each training step, 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 whose weight vector has the greatest similarity with the input sample $\mathbf{x}$. This similarity is usually defined via a distance (usually Euclidean) and, formally speaking, the BMU can be defined as the neuron for which:

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_i \|\mathbf{x} - \mathbf{m}_i\| \tag{6}$$

where $\|\cdot\|$ is the adopted distance measure. After finding the BMU, the weight vectors of the SOM are updated and the weight vectors of the BMU and of its topological neighbors are moved in the direction of the input vector, in the input space. The SOM updating rule for the weight vector of the unit *i* can be written as:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{7}$$

where t denotes the time, $\mathbf{x}(t)$, the input vector and $h_{ci}(t)$ the neighborhood kernel around the winner unit, defined as a non-increasing function of the time and of the distance of the unit i from the winner unit c which defines the region of influence that the input sample has on the SOM. This kernel is composed by two parts: the neighborhood function $h(d, t)$ and the learning rate function $\alpha(t)$:

$$h_{ci}(t) = h(\|\mathbf{r}_c - \mathbf{r}_i\|, t)\alpha(t) \tag{8}$$

where r_i is the location of unit i on the map grid. The AstroMining package allows the use of several neighborhood functions, among which the most commonly used is the so called Gaussian neighborhood function:

$$\exp(-\|\mathbf{r}_c - \mathbf{r}_i\|^2/2\sigma^2(t)) . \tag{9}$$

The learning rate $\alpha(t)$ is a decreasing function of time which, always in the AstroMining package, is:

$$\alpha(t) = (A/t + B) \tag{10}$$

where A and B are some suitably selected positive constants. Since also the neighbors radius is decreasing in time, then the training of the SOM can be seen as performed in two phases. In the first one, relatively large initial α value and neighborhood radius are used, and decrease in time. In the second phase both α value and neighborhood radius are small constants right from the beginning.

2.3 Neural networks for Principal Component Analysis

Principal Component Analysis (PCA) is a widely used technique in data analysis. Mathematically, it is defined as follows: let $\mathbf{C} = E(\mathbf{x}\mathbf{x}^T)$ be the covariance matrix of L-dimensional zero mean input data vectors $\mathbf{x}$. The i-th principal component of $\mathbf{x}$ is defined as $\mathbf{x}^T\ \mathbf{c}(i)$, where $\mathbf{c}(i)$ is the normalized eigenvector of $\mathbf{C}$ corresponding to the i-th largest eigenvalue $\lambda(i)$. The subspace spanned by the principal eigenvectors $\mathbf{c}(1),\ldots,\mathbf{c}(M)$, $(M < L)$ is called the PCA subspace (of dimensionality M) [50].

PCA's can be neurally realized in various ways [51]; [50];[54]. The PCA neural network used by us is a one layer feedforward neural network which is able to extract the principal components of the stream of input vectors. Typically, Hebbian type learning rules are used, based on the one unit learning algorithm originally proposed by [50].

The structure of the PCA neural network can be summarised as follows [52, 53, 50]: there is one input layer, and one forward layer of neurons totally

connected to the inputs; during the learning phase there are feedback links among neurons that classify the network structure as either hierarchical or symmetric. After the learning phase the network becomes purely feedforward. The hierarchical case leads to the well known GHA algorithm [53, 54]; in the symmetric case we have the Oja's subspace network.

PCA neural algorithms can be derived from optimisation problems, such as variance maximization and representation error minimisation [52, 53] so obtaining nonlinear algorithms (and relative neural networks). These neural networks have the same architecture of the linear ones: either hierarchical or symmetric. These learning algorithms can be further classified in: robust PCA algorithms and nonlinear PCA algorithms. We define robust a PCA algorithm when the objective function grows less than quadratically. The nonlinear learning function appears at selected places only. In nonlinear PCA algorithms all the outputs of the neurons are nonlinear function of the responses.

In the robust generalization of variance maximisation, the objective function $f(t)$ is assumed to be a valid cost function such as $\ln(\cosh(t))$. This leads to the algorithm:

$$\begin{aligned} \mathbf{w}_{k+1}(i) &= \mathbf{w}_k(i) + \mu_k g(y_k(i))\mathbf{e}_k(i), \\ \mathbf{e}_k(i) &= \mathbf{x}_k - \sum_{j=1}^{I(i)} y_k(j)\mathbf{w}_k(j) \end{aligned} \tag{11}$$

In the hierarchical case we have $I(i) = i$. In the symmetric case $I(i) = M$, the error vector $\mathbf{e}_k(i)$ becomes the same $\mathbf{e}_k$ for all the neurons, and equation (11) can be compactly written as:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu \mathbf{e}_k g(\mathbf{y}_k^T) \tag{12}$$

Where $\mathbf{y}_k = \mathbf{W}_k^T \mathbf{x}$ is the instantaneous vector of neuron responses. The learning function g, derivative of f, is applied separately to each component of the argument vector.

The robust generalisation of the representation error problem [52, 53], with $f(t) \leq t^2$, leads to the stochastic gradient algorithm :

$$\begin{aligned} \mathbf{w}_{k+1}(i) &= \mathbf{w}_k(i) + \mu(\mathbf{w}_k(i)^T g(\mathbf{e}_k(i))\mathbf{x}_k + \\ &+ \mathbf{x}_k^T \mathbf{w}_k(i) g(\mathbf{e}_k(i))) \cdot \end{aligned} \tag{13}$$

This algorithm can be again considered in both hierarchical and symmetric cases. In the symmetric case $I(i) = M$, the error vector is the same ($\mathbf{e}_k$) for all

the weights $\mathbf{w}_k$. In the hierarchical case $I(i) = i$, equation (13) gives the robust counterparts of principal eigenvectors $\mathbf{c}(i)$.

Let us consider now the nonlinear extensions of PCA algorithms. We can obtain them in a heuristic way by requiring all neuron outputs to be always nonlinear in the equation (11). This leads to:

$$\begin{aligned} \mathbf{w}_{k+1}(i) &= \mathbf{w}_k(i) + \mu g(y_k(i))\mathbf{b}_k(i), \\ \mathbf{b}_k(i) &= \mathbf{x}_k - \sum_{j=1}^{I(i)} g(y_k(j))\mathbf{w}_k(j) \quad \forall i = 1, \ldots, p \end{aligned} \tag{14}$$

3. Some applications

In the following paragraphs we shall describe some recent results obtained with the above described NNs in widely different fields of research: evaluation of photometric redshifts, analysis of stellar light curves, characterization of photometric data in the GOODS catalogues and, finally, a preliminary analysis of simulated cosmic ray air showers detected with Cherenkov telescopes.

3.1 Photometric redshift

The accurate knowledge of redshifts for large samples of galaxies is a precondition to most extragalactic and cosmological studies. Unfortunately, the measurement of accurate redshifts requires low/medium resolution spectroscopy with large telescopes. An alternative (even though less accurate) approach is the evaluation of "photometric redshifts", *id est* estimates derived from photometric data obtained in several properly selected broad or intermediate photometric bands. The many approaches proposed to derive photometric redshifts may be divided into two groups:

- Analytical methods based on the fitting of the observed photometric data with those predicted from spectral synthesis models (see for instance [56], [57], [58], [59]);
- Interpolative methods relying on spectroscopic data for a sample of template objects which can be used to constrain the fit of a polynomial function mapping the photometric data [60], [61], [62]. While more demanding in terms of input data, these 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.

As stressed above, NNs are excellent tools for interpolating data and for extracting patterns and trends and are therefore particularly suited to an interpolative approach. The implemented package makes use of a Bayesian

MLP to interpolate the photometric redshift with a very good predictive result on objects, and of a SOM to identify the confidence level of the derived redshifts and to evaluate the degree of contamination of the final redshift catalogues. In the experiments described below, the package was applied to the Sloan Digital Sky Survey Early Data Release (EDR) [63] which provides photometric, astrometric and morphological data for an estimated 16 millions of objects in two fields: an Equatorial 2° 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 fairly large redshift range (93% of the objects have $z < 0.7$). In order to build the training, validation and test sets, we first extracted from the SDSS-EDR a set of parameters (both total and petrosian) magnitudes in the *u*, *g*, *r*, *i*, *z* bands, petrosian radii, 50% and 90% petrosian flux levels, surface brightness and extinction coefficients, [63] for all galaxies in the photometric sample.

It needs to be stressed that an 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 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.

All used NNs had only one hidden layer and the experiments were performed varying the number of input parameters and of hidden units. Extensive experiments lead to conclude that the Bayesian framework provides better generalization capabilities with a lower risk of overfitting, and that the best compromise between speed and accuracy is achieved with a maximum of 22 hidden neurons and 10 Bayesian cycles.

Table 1 summarizes some of the results obtained from the experiments while Fig. 2 shows the comparison between spectroscopic and photometric redshifts for the test set objects in the best experiment. As it can be seen, the r.m.s scatter around the average line is $z \simeq 0.021$.

One of the problems encountered in the use of photometric redshifts 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 even more relevant in the case of NNs based methods, since supervised NNs are necessarily trained only in a limited range of redshifts and, when applied to real data, they produce misleading results for most (if not all) objects which "in the real word" have redshifts falling outside the training range. A behavior which, once more, is due

to the fact that NN are good interpolators but have very little, if any, extrapolation capabilities.

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

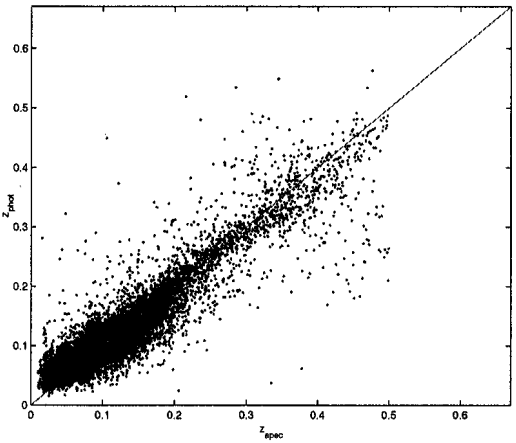

Figure 2. Photometric versus spectroscopic redshifts obtained with a Bayesian MLP with 2 optimization cycles, 50 learning epochs of quasi-Newton algorithm and 5 inner epochs for hyperparameter optimization. Hyperparameters were initialized at $\alpha = 0.001$ and $\beta = 50$.

In the SDSS-EDR spectroscopic sample, over a total of 47,000 (divided roughly in three equal sets for test, validation and training) objects having $z > 0$, only 88%, 91% and 93% have redshift z lower than 0.3, 0.5 and 0.7, respectively. To train the network only on objects falling in the above ranges implies, respectively, a minimum fraction of 12%, 9% and 7% of objects in the photometric data set which will have wrong estimates of the photometric redshift. A more robust estimate of the contamination may however be obtained using unsupervised SOM clustering techniques over the training set.

In Figure 3 we show 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 outer and right parts of the map, intermediate redshift ones ($0.5 < z < 0.7$) activate mainly 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 (summarized 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. 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. Obviously, 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.

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%

The above derived topology of the network is 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. 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 (contamination) matrix helps in better quantifying the quality of the results (see 3.1): data were divided into three classes accordingly to their spectroscopic redshifts, namely

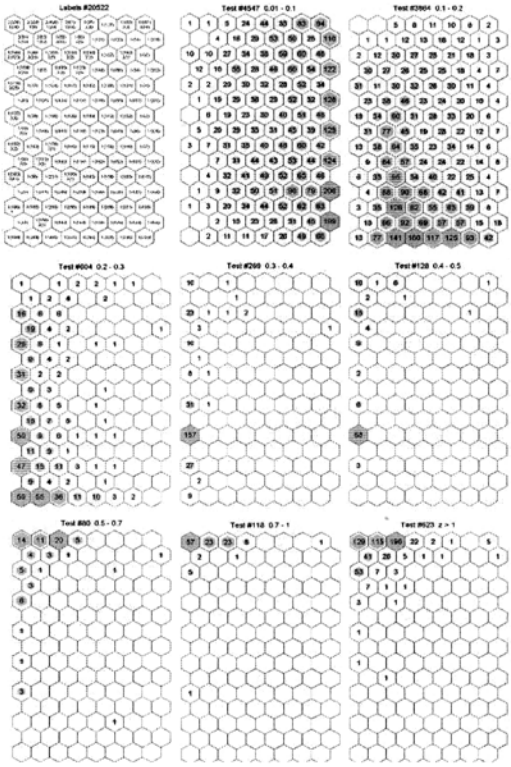

Figure 3. Maps of the neurons activated by the input data set. Exagons represent the NN nodes. In the upper left corner map, for a given node, the figures n(m) can be read as follows: n is the class (n=1 meaning $z < 0.5$ and n=2 meaning $z > 0.5$) and m is the number of input vector of the correspondent 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.

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 were correctly identified and only 4.6% was attributed a wrong redshift estimate. In total, 94.2% of the objects 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%.

3.2 SOM clustering of the GOODS catalogues

The recent Great Observatories Origin Deep Survey (GOODS, [64]) provides an excellent test bed to explore the clustering and visualization capabilities of SOMs. The GOODS survey covers at several wavelengths the Chandra Deep Field South (CDF-S) [65]. The available catalogues provide photometric multiwavelength data reduced to a common system: optical broad band UBVRI photometry obtained using the Wide Field Imager (WFI) at the ESO/MPG 2.2 m telescope, near infrared (JHK) photometry obtained with the SOFI imager at the ESO/NTT. Additional information on the X and radio fluxes are also available but since they will not be used here, we shall neglect them. Object catalogues were extracted using the package S-Extractor [66] from each co-added image were combined in two multi-color lists:

- the UBVRI catalogue contains 75000 sources (WFI) in a sky area of approximately 0.5 square degrees;
- the UBVRIJK catalogue contains more than 28000 sources (WFI + SOFI) in an area of approximately 0.25 square degrees.

The problem addressed here is to explore whether the clustering and visualization properties of the SOM can be used to single out clusters of objects with homogeneous physical characteristics (e.g., QSOs and active nuclei vs. normal galaxies, stars of different spectral types, etc. [55]).

One of the main problems is posed by the so called "drop-outs", namely objects which are below the detection treshold in at least one of the available bands. This problem is especially relevant in the GOODS catalogues due to the very different sensitivities of the instruments (54752 drop-outs in the UBVRI, and 24872 in the UBVRIJK catalogues, respectively).

Missing data may be of two types: just empty fields or upper limits. In the GOODS catalogues, the latter are derived from the limiting magnitude of the image and the error is arbitrarily set to −1. In what follows we describe the results of two different groups of experiments which were performed on the two catalogues, in the first group drop-outs were treated as missing data (NaN, *id est*, Number), while in the second one we made use of the associated upper limits. The parameters extracted with S-Extractor [66] and used in the experiments were the following:

- the 5 (or 7) MAG-AUTO magnitudes (corrected for the galactic extinction), together with the errors;

- the semi-axes as measured on each individual band (2 × 5 or 7 bands), with their errors;
- the background level associated to each source ;
- the CLASS-STAR galaxy/star classification parameter.

Among all the experiments with SOM, only those with grid size set to 65 × 31 will be described. This value was empirically determined and is the result of a compromise between the accuracy of the modelling and the computational resources available. The training set was linearly scaled so that for each feature the variance and the mean are 1 or 0, respectively.

In Fig. 4 we show the U-matrix and the similarity coloring map in the case of NaN missing objects. Two main clusters are evident: a very large and homogeneous area plus two smaller areas located at the opposite corners. Using as label the stellarity index provided by S- Extractor, and the unit activation diagrams for the *class* – *star* = 1 and *class* – *star* = 0 objects (Fig. 5) it is clear that: the vast majority of the *class* – *star* = 1 objects is localized in the top right corner areas of the map and a smaller number activates neurons located at the top side of the map. *Class–star* = 0 are instead distributed over the entire map even though the majority of them activates neurons in the bottom part of the map. In other words, the SOM clearly recognizes the existence of two large groups plus several regions of overlap containing ca. 200 objects.

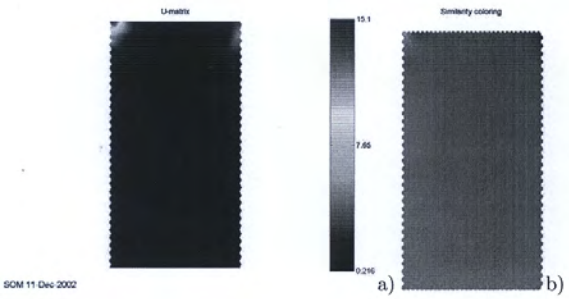

Figure 4. UBVRI catalogue with NaN: a) U-matrix; b) similarity coloring.

If instead of NaN upper limits are used, the resulting visualizations appear different (Figg. 6 and 7). The U-matrix and similarity coloring show several well defined clusters in which the *class* – *star* = 1 and *class* – *star* = 0 objects are localized mainly in the top and bottom sides of the map, respectively. Keeping in mind that objects belonging to the same cluster share (in a statistical sense) the same properties, it becomes apparent that the clusters are due to the fact that upper limits derived from the limiting magnitude of

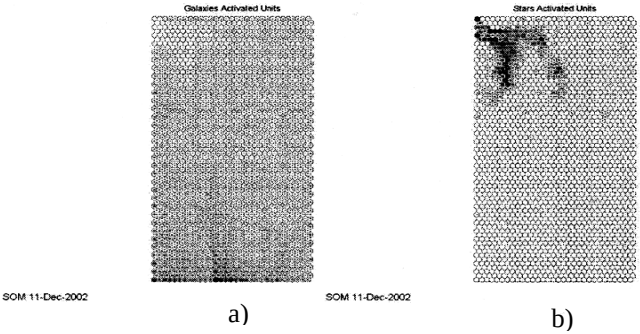

Figure 5. UBVRI catalogue with NaN: a) BMU's of *class* – *star* = 0 objects; b) BMU's of *class* – *star* = 1 objects.

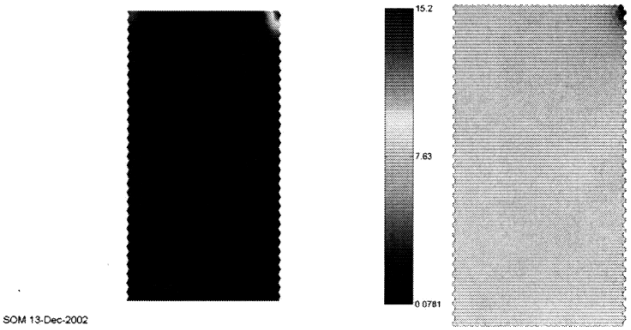

Figure 6. UBVRI catalogue with Upper Limits: a) U-matrix; b) similarity coloring.

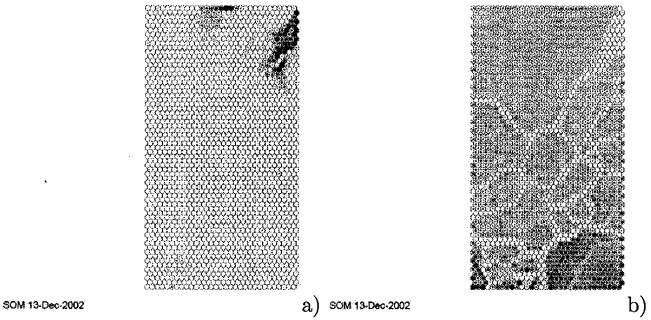

Figure 7. UBVRI catalogue with Upper Limits: a) BMU's of *class* – *star* = 1 objects; b) BMU's of *class* – *star* = 0 objects.

different images are the same for many objects and introduce a spurious similarity among objects. If clustered objects are compared on a one-to-one basis, however, the NaN and upper limits maps are consistent.

Surprisingly, the introduction of the J and K bands does not change significatively the performance of the trained SOM's. A possible reason could be the presence of a considerable number of drop-outs in this second catalogue. If we compare the results with those obtained with the UBVRI catalogue (in both the NaN and Upper limits cases) we observe a rather regular trend (cf. Figg. 8 through 11). We wish however to stress that the large number of drop-out, models the shape of the resulting SOM's when using upper limits.

It has to be stressed that photometric errors were not discussed here since using them as independent variable would not provide additional information to better characterize the objects in the catalogues. However, errors may be included by weighting the magnitudes on a relative error values basis. In this way, for example, the values in a given pass-band are considered more or less reliable on the basis of its weight.

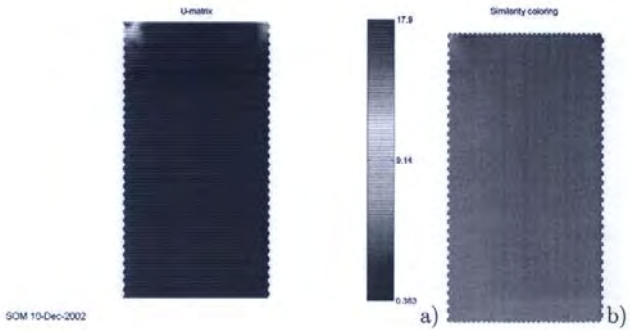

Figure 8. UBVRIJK catalogue with NaN: a) U-matrix; b) similarity coloring.

3.3 Time series analysis

The analysis of light curves of variable objects is among the oldest types of research in astrophysics, nevertheless, however, many methodological problems connected with the search for periodicities have still to be satisfactorily solved. Astronomical light curves, in fact, are often very unevenly sampled and sometimes cover only a fraction of the variation period. When dealing with a light curve, two type of questions must usually be answered: is there a fundamental period? Are there multiple periodicities?

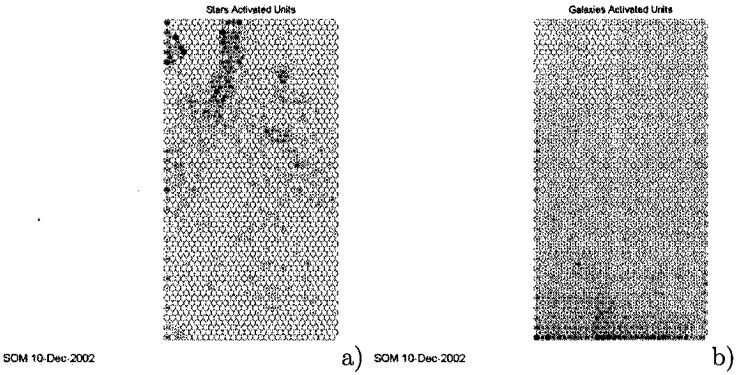

Figure 9. UBVRIJK catalogue with NaN: a) BMU's of *class* – *star* = 1 objects; b) BMU's of *class* – *star* = 0 objects.

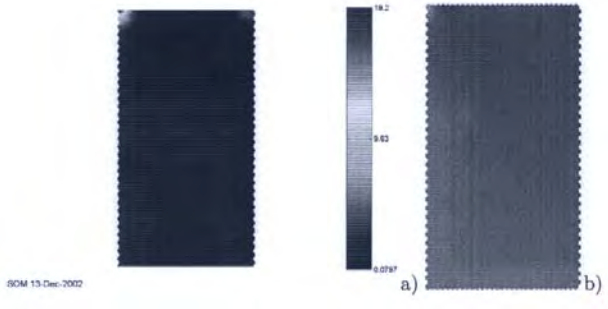

Figure 10. UBVRIJK catalogue with Upper Limits: a) U-matrix; b) similarity coloring.

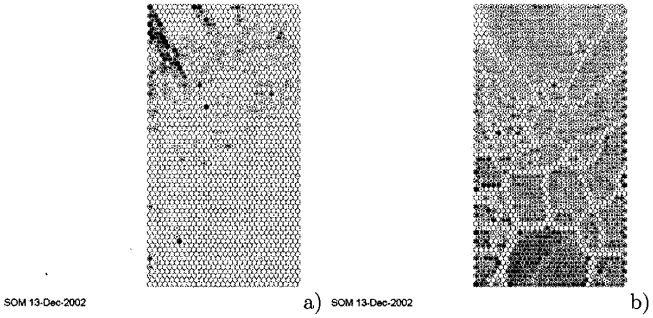

Figure 11. UBVRIJK catalogue with Upper Limits: a) BMU's of *class* – *star* = 1 objects; b) BMU's of *class* – *star* = 0 objects.

The general problem is how to estimate the frequencies of periodic signals which may be contained in a physical variable x measured at discrete times t_i, where we assume the physical variable x to be the sum of a variable signal and a random noise i.e. ($x_i = x(t_i) = x_s(t_i) + r_i$) [29], [30], [67].

Many different tools based on different types of Fourier analysis may be effectively used in the case of even sampling [68] [69] [70], but they lead to ambiguous results when are applied to evenly sampled data (see, for instance, [73], [30]).

The most commonly used tool is the so called Periodogram, which is an estimator of the signal energy in the frequency domain (cf. [71], [68], [72], [73]) and has been extensively applied to the analysis of unevenly sampled stellar light curves.

Its use, however, is undermined by the difficulties encountered in dealing with the aliasing effects. A variation of the classical Periodogram (P) was introduced by Lomb [73] and is distributed in the Numerical Recipes in C [74]. The normalized Lomb's Periodogram (LP) is the power spectra as function of an angular frequency $\omega \equiv 2\pi f > 0$ estimated with respect to the measured times t_i [73].

More modern frequency estimators, based on the signal autocorrelation matrix eigenvectors, have been introduced to overcome the biases of more traditional Fourier methods.

In the following we describe a recently introduced algorithm (Stima) that is based on the frequency estimator MUSIC and a nonlinear PCA technique based on NNs to extract the principal components of the autocorrelation matrix (cf. [29], [30]).

Let us assume to have a signal with *p* sinusoidal components (narrow band). The *p* sinusoids are modelled as a stationary ergodic signal, and this is possible only if the phases are assumed to be indipendent random variables uniformly distributed in [0, 2π) [72]. In order to estimate the frequencies we exploit the properties of the signal autocorrelation matrix (a.m.) [72]. The a.m. is the sum of the signal and the noise matrices; the *p* principal eigenvectors of the signal matrix allow the estimate of frequencies; the *p* principal eigenvectors of the signal matrix are the same of the total matrix.

In this way, to extract the principal components we use a robust nonlinear PCA NN ([75], [76], [77], [78]) and then we apply a modified version of the MUlty SIgnal Classifier (MUSIC) estimator [71], which can be applied directly on unevenly sampled data, to obtain the periodicities ([29], [30])

In this case, the fundamental learning parameters of the NN are: i) the initial weight matrix; ii) the nonlinear learning algorithm; iii) the number of neurons (*p*), id est the number of principal eigenvectors which need to be

estimated or twice the number of periodicities expected in the signal; iv) and two learning parameters [30].

The weight matrix **W** is initialized either with small random values or, only in the case of regular signals and in order to speed up the convergence of the neural estimator, with the first pattern of the signal as columns of the matrix. In order to establish when the NN has converged and to accomplish the periodicities estimation on unevenly sampled data (without interpolation) we use the following modified MUSIC estimator:

$$P_{MUSIC} = \frac{1}{M - \sum_{i=1}^{M} \left| \mathbf{e}_f^H \mathbf{w}(i) \right|^2} \tag{15}$$

where $\mathbf{w}(i)$ is the i-th weight vector of the neural net after learning, and

$$\mathbf{e}_f^H = [1, \mathbf{e}_f^{j2\pi f t_0}, \ldots, \mathbf{e}_f^{j2\pi f t_{(L-1)}}]^H. \tag{16}$$

where $\{t_1, \ldots, t_{(L-1)}\}$ are the first L components of the temporal coordinates of the uneven signal. We note that for interpolated data

$$\mathbf{e}_f^H = [1, \mathbf{e}_f^{j2\pi f}, \ldots, \mathbf{e}_f^{j2\pi f(L-1)}]^H \tag{17}$$

After the learning process, the weight matrix columns are feed to the frequency estimator MUSIC and the estimated signal frequencies are obtained as the peak locations of the functions. We note that if f is the frequency of the $i - th$ sinusoidal component $f = f_i$, we shall have $\mathbf{e} = \mathbf{e}_i$, and $P_{MUSIC} \rightarrow \infty$. In practice, this means that there will be a peak in correspondence of the component frequency P_{MUSIC}.

The above methods have been applied, for instance, to the light curve of the W UMa-system U Pegasi [79, 80, 81, 82, 83, 84]. The UMa-stars have been the subject of more controversy than any other group of close binaries. Many factors account for their great importance in the framework of stellar evolution, for example, their extraordinary abundance in space, and the instability of their light curves [83]. Numerous models have been introduced to explain these light curves, which are affected by the geometry of the star system and the intensity distribution over the surface of the components. The models on the eclipses produced by tidally distorted stars [83]. For our analysis we consider two different data sets. In the first case (I-DS in the following) the sequence is composed by 446 *V* observations and by 441 *B* observations obtained from October to November 1958 [79] (Fig. 12a). In the second case

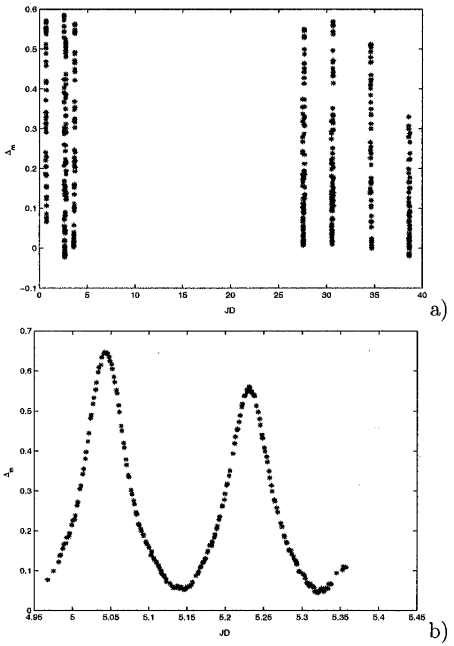

Figure 12. Light curve (B): a) light curve in [79]; b) light curve in [82].

(II-DS in the following) the observations are obtained with 60 cm reflector at the Beijing Observatory [82]. In [82] is obtained a complete coverage of the *B* and *V* light curves of U Pegasi on October 3/4 in 1978 and a total of 217 pairs of *B* and *V* observations were obtained (fig. 12b).

Many photometric periods have been derived for the system U Pegasi; e.g. 0^d. 37478192 [79], 0^d.37478133 [84], and 0.374781802 [82]. From our analysis (in the following we consider only the *B* sampling) we obtain a period of 0^d.17713636 using the LP approach (fig. 13a) and a period of 0^d. 18739024 using the Stima approach (fig. 13b) on the first data set, and a period of 0^d. 18125581 using the LP approach (fig. 14a) and a period of 0^d.18739024 using the Stima approach (fig. 14b) on the second data set. In figure 15 we plot the fitting of the light curve with the sinusoidal curves having the periods obtained by us and the period of 0^d. 37478192 [79]. We note that for this kind of light curve (see fig. 15) the secondary minimum is similar to the principal minimum [79, 83, 82]. For this reason the known periodicity is twice the period by us estimated. If we consider the double of the period estimated by Stima, we obtain 0^d. 37478192 that is the period estimated in [79].

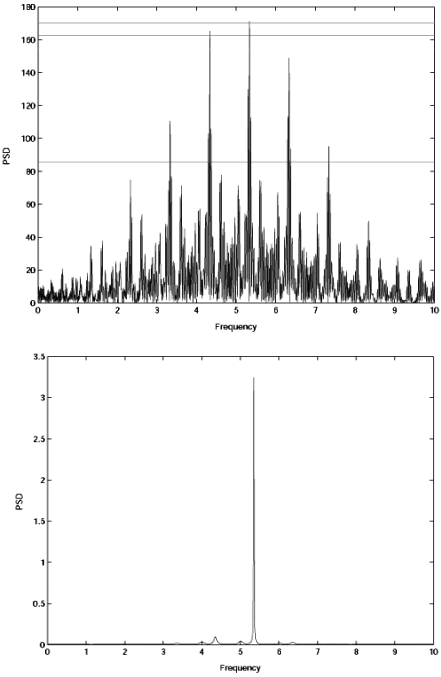

Figure 13. Periodicities estimation of the U Pegasi (I-DS): a) PSD of the Lomb approach; b) PSD of the Stima approach.

3.4 Analysis of simulated AUGER data

The AUGER cosmic ray observatory is under construction near Malargue Argentina [85]. In its final configuration it will consist of an array of 1600 water cherenkhov detectors (10 m^3 each) distributed over an area of ca. 3000 km^2. The ground based array operates in conjunction with 24 air fluorescence detectors, each covering a solid angle of 30°× 30° square degrees. At the selected energies, each primary particle produces extensive showers of several billions of secondary particles exiting Nitrogen molecules which thus emit fluorescence light. The experiment is therefore conceived to study both the vertical and horizontal structures of showers produced in the upper layers of the atmosphere by very high energy progenitors.

The COsmic Ray Simulations for KAscade (CORSIKA) software [86] was implemented to produce simulated showers taking into account the nature, inclination with respect to the horizon and the energy of the primary particle. As it can be seen from Fig. 16 the showers expected for different primary

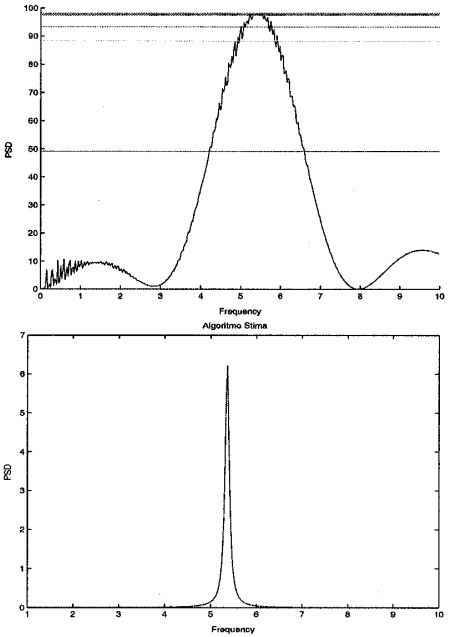

Figure 14. Periodicities estimation of the U Pegasi (II-DS): a) PSD of the Lomb approach; b) PSD of the Stima approach.

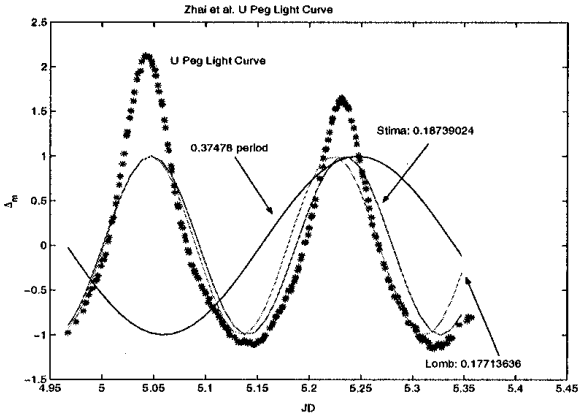

Figure 15. Periodicities estimation of the U Pegasi Light curve ((II-DS) *B* observation).

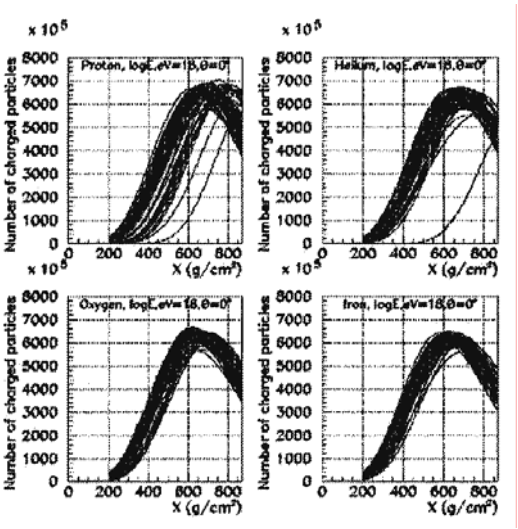

Figure 16. Simulated AUGER events obtained with CORSIKA for four types of primary particles (*p*, *He*, *O*, *Fe*) having energy 10^{18} GeV and vertical incidence.

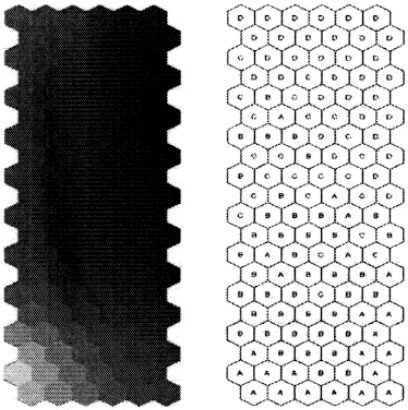

Figure 17. Left: SOM similarity coloring map: each hexagon represents a neuron and different colors denote different clusters. Right: neurons are labeled using simulated data (A=proton; B=Helium; C=Oxygen; D=Iron).

nuclei are discouragingly similar in shape. In order to assess whether NNs could be used to disentangle the nature of the primary particles several

experiment were performed using both supervised and unsupervised methods. In all experiments we used only the information concerning the shower longitudinal development in the atmosphere and we made use of 4000 simulated CORSICA Montecarlo showers referring to vertical primary particles of 10^{18} GeV and four primary particles: *p*, *He*, *O* and *Fe* (1000 simulations each).

The unsupervised experiments were performed using a SOM with 120 neurons and as labeling set we made use of the 4000 simulated curves mentioned above. Each neuron was then attributed to a specific class accordingly to the type of event which had activated that neuron more times. Results may be summarised as follows: *P* = 34% success rate, *He* = 30%, *O* = 28%, *Fe* = 41%.

The supervised experiments were instead performed using a Bayesian MLP and auxiliary sets extracted from the above quoted simulated curves (600 input for the training set, 300 and 100 for the validation and test sets respectively). In order to obtain for each input vector the probability that it belongs to a given class, we adopted the SoftMax activation function and the Entropy function as error estimator.

As a first attempt we trained the network on the 6 parameters resulting from the model driven fit of the Gaiser–Hillas curves and therefore the adopted MLP consisted of 6 input neurons, one hidden layer with a variable number of neurons and 4 output neurons. Results are shown in the Fig.18.

From the above preliminary results it is apparent that even in absence of a fine tuning of the networks, supervised methods outperform unsupervised ones. More extensive testing is therefore needed in order to select the most suitable neural model (in terms of architecture, number of input neurons and hidden layers, activation and error function etc.). Experiments need also to be performed on more realistic simulated data taking into account the noise, the uneven sampling and the expected incompleteness of the sampling.

4. Conclusions

In this paper we summarised the most relevant properties of the main types of NNs used for astronomical applications focusing our attention to those models which are most commonly encountered or which were developed within the framework of the *Astroneural* collaboration. Neural Networks turn out to be very promising and effective tools for astronomical data mining in high dimensionality parameter spaces for a wide variety of tasks: exploratory analysis of the data, classification, pattern recognition, data compression. The main problems encountered so far, which still prevent a more widespread use may be summarised as follows:

- NNs tools require to be tailored and optimized on the specific tasks and, at least so far, no general, user friendly tool has been yet implemented. This obviously restricts their domain of application either to tasks which are very demanding in terms of quantity of data to be processed or to tasks were the results cannot be properly handled with other, more traditional techniques.
- NN's do not provide a straightforward set of rules and therefore may cause some uneasiness to non experienced users who prefer results which may be less accurate but are more transparent.
- Even though in most applications, supervised methods provide better results than unsupervised ones, their use is often limited by the problems encountered in building significant training/validation/test sets. This problem, however, should be greatly reduced in a near future when the implementation of the International Virtual Observatory will allow to gather large amounts of homogeneous data.

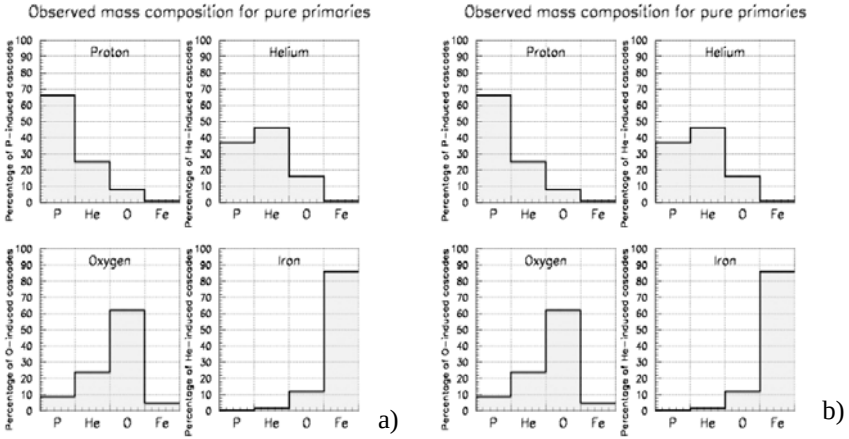

Figure 18. Results for two MLP (22 hidden neurons, SoftMax activation function). Panel a: coniugate gradient optimization algorithm. Panel b: discendent gradient optimization function. The diagrams give the distribution among the four types of primary particles as obtained after labeling.

Acknowledgements

This work was partly sponsored by the Italian Ministry for Public Education and Research (MIUR) through a COFIN action, by the European Action COST-TIST 283 and by the INFN.

References

1. Bailer–Jones, C.A.L., Gupta, R., Singh, H.P. (2001). Automated Data Analysis in Astronomy, Gupta et al. eds., astro-ph/0102224
2. Brunner J.R., Djorgovski S.G., Prince T., Szalay A.S.(2002). Massive data sets in Astronomy, Handbook of Massive Datasets, Ch. 27, J. Abello, P. Pardalos, and M. Resende Eds., Kluwer:Dordrecht
3. Brunner J.R., Djorgovski S.G., Szalay A.S. editors (2000), Proceed. of the Int. Workshop: Virtual observatories of the future, Astron. Soc. of the Pacific Conf. Series, n.225
4. Banday A.J., Zaroubi S., Bartelmann M. editors (2001), Proceed. of the Int. Workshop: Mining the sky, Springer:Heidelberg
5. Tagliaferri R., Longo G., D'Argenio B., Incoronato B. Eds., (2003), 2003 Special Issue Neural Network Analysis of Complex Scientific Data: Astronomy and Geosciences, vol. 16.
6. 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 in Astronomy, Neural Networks, 16, 297
7. Gulati, R.K., Altamirano, L. (2001). Artificial Neural Networks in Stellar Astronomy, in Focal Points in Latin American Astronomy, A. Aguilar ed., Revista Mexicana de Astronomia y Astrofisica Serie de Conferencias, **85**
8. Odewhan, S.C., Stockwell, E.B., Penninton, R.L., Humpreys, R.M., Zumach, W.A. (1992). Automated Star/Galaxy Discrimination with Neural Networks, Astronomical Journal, **103**, 318–331
9. Klusch, M., Napiwotzki, R. (1993). HNS - A Hybrid Neural System and its Use for the Classification of Stars, Astronomy & Astrophysics, **276**, 309–319
10. Andreon, S., Gargiulo, G., Longo, G., Tagliaferri, R., Capuano, N. (2000). Wide Field Imaging. I. Applications of Neural Networks to Object Detection and Star/Galaxy Classification, Monthly Notices Royal Astronomical Society, **319**, 700–716
11. Odewhan, S.C., Nielsen, M.L. (1994). Star-Galaxy Separation Using Neural Networks, Vistas in Astronomy, **38**, 281–301
12. Goderya, Shaukat, N., Lolling, S.M. (2002). Morphological Classification of Galaxies Using Computer Vision and Artificial Neural Networks: A Computational Scheme, Astrophysics and Space Science, **279**, 377–387
13. Bazell, D., Aha, D.W. (2001). Ensembles of Classifiers for Morphological Galaxy Classification, Astrophysical Journal, **548**, 219–223
14. Ball, N.M. (2001). Morphological Classification of Galaxies Using Artificial Neural Networks, Master Thesis, astro-ph/0110492
15. Coryn, A.L., Bailer-Jones, C. (2000). Stellar Parameters from Very Low Resolution Spectra and Medium Band Filters: Teff, log g and [M/H] Using Neural Networks, astro-ph/0003071
16. Snider, S., Allende–Prieto, C., von Hippel, T., Beers, T.C., Sneden, C., Qu, Y., Rossi, S. (2001). Three-dimensional Spectral Classification of Low-Metallicity Stars Using Artificial Neural Networks, Astrophysical Journal, **562**, 528–548
17. Weaver, B. (2002). Artificial Neural Networks and Spectral Classification, The Observatory, **122**, 166

18. Lepper, K., Whitley, V.H. (2002). Distinguishing Sedimentary Depositional Environments on Mars Using In-Situ Luminescence Measurements and Neural Network Analysis, in 33-rd Annual Lunar and Planetary Science Conference, Abs. 1400
19. Birge, B., Walberg, G. (2002). A Computational Intelligence (CI) Approach to the Precision Mars Lander Problem, Technical Report, North Carolina State Univ.
20. Borda, F., Roberto, A., Mininni, P.D., Mandrini, C.H., Gómez, D.O., Bauer, O.H., Rovira, M.G. (2002). Automatic Solar Flare Detection Using Neural Network Techniques, Solar Physics, **206**, 347–357
21. Ganguli, S.B., Gavrichtchaka, V.V., Von Steiger, R. (2002). Adaptive Systems for Detection and Forecasting of Coronal Mass Ejections From Solar Mean Magnetic Field, in 2002 American Geophysical Union – Spring Meeting, SH52A-01
22. Lopez Ariste, A., Rees, D.E., Socas-Navarro, H., Lites, B.W. (2001). Pattern Recognition Techniques and the Measurement of Solar Magnetic Fields, in Proc. of Astronomical Data Analysis, J.L. Starck and F.D. Murtagh eds., SPIE Vol. 4477, pp. 96–106
23. Steinegger, M., Veronig, A., Hanslmeier, A., Messerotti, M., Otruba, W. (2002). A Neural Network Approach to Solar Flare Alerting, in 11-th Cambridge Workshop on Cool Stars, Stellar Systems and the Sun, ASP Conference Series, Vol. 223, R.J. Garcia Lopez R. Rebolo and M.R. Zapaterio Osorio eds., pp. 1165–1169
24. Rosa, R.R., Sawantí, H.S., Cecatto, J.R., Neto, C., Rodrigues Lopes, V.C.A., Subramanian, K.R., Fernandes, F.C.R., Saito, J.H., Moron, C.E., Mucheroni, M.L., Furuya, N., Mascarenhas, N. (2000). Phenomenological Dynamics of Coronal Loops Using a Neural Network Approach, Advances in Space Research, **25**, 1917–1921
25. Veselovskii, I.S., Dmitriev, A.V., Orlov, Yu.V., Persiantsev, I.G., Suvorova, A.V. (2000). Simulation of Statistical Distributions in the Space of Parameters of the Solar Wind and Interplanetary Magnetic Field Using Artificial Neural Networks, Astronomicheskii Vestnik, **34**, 116–125
26. Fuentes, O. (2001). Automatic Determination of Stellar Atmospheric Parameters Using Neural Networks and Instance-Based Machine Learning, Experimental Astronomy, **12**, 21–31
27. Torres, S., Garca-Berro, E., Isern, J. (2001). Classification of the White Dwarf Populations Using Neural Networks, in Mining the Sky, A.J. Banday S. Zaroubi and M. Bartelmann eds., Heidelberg: Springer-Verlag, pp. 391–393
28. Weaver, Wm.B. (2000). Spectral Classification of Unresolved Binary Stars with Artificial Neural Networks, Astrophysical Journal, **541**, 298–305
29. Tagliaferri, R., Ciaramella, A., Barone, F., Milano, L. (2001). Neural Net-works for Spectral Analysis of Unevenly Sampled Data, in Mining the Sky, A. J. Banday Zaroubi S. and M. Bartelmann eds., Heidelberg: Springer-Verlag, pp. 386–390
30. Tagliaferri, R., Ciaramella, A., Milano, L., Barone, F., Longo, G. (1999). Spectral Analysis of Stellar Light Curves by Means of Neural Networks, Astronomy & Astrophysics Supplement Series, **137**, 391–405
31. Balastegui, A., Ruiz-Lapuente, P., Canal, R. (2001). Reclassification of Gamma-Ray Bursts, astro-ph/0108272

32. Tagliaferri, R., Longo, G., Andreon, S., Capozziello, S., Donalek, C.,Giordano, G. (2002). Neural Networks and Photometric Redshifts, astro-ph/0203445
33. Vanzella, E, Cristiani, S., Arnouts, S., Dennefeld, M., et al. (2002). A VLT/FORS2 spectroscopic survey in the HDF-S, astro-ph/0210146
34. Funaro, M., Oja, E., Valpola, H. (2003). Independent Component Analysis for Artefact Separation in Astrophysical Images, in this volume, Neural Networks
35. Baccigalupi, C., Bedini, L., Burigana, C., De Zotti, G., Farusi, A., Maino, D., Maris, M., Perrotta, F., Salerno, E., Toffolatti, L., Tonazzini, A. (2000). Neural Networks and the Separation of Cosmic Microwave Background and Astrophysical Signals in Sky Maps, Monthly Notices Royal Astronomical Society, **318**, 769–780
36. Murtagh, F., Donalek, C., Longo, G., Tagliaferri, R. (2002). Bayesian Model Selection for Spatial Clustering in 3D Surveys, Proc. SPIE, Vol. 4847, in press
37. Patrick, G., Gabriel, S., Rodgers, D., Clucas, S. (2002). Neural Network Prediction of Solar Energetic Proton Events with Long Lead Times, in Second Solar Cycle and Space Weather Euroconference, Sawaya-Lacoste H. eds., ESA SP-477, pp.517–520
38. Medina Tanco, G.A., Sciutto, S.J. (2001). A Multivariate Study of Mass Composition for Simulated Showers at the Auger South Observatory, astro-ph/0109140
39. Aramo C., Donalek C., et al. (2003). Analysis of simulated AUGER data, in "Thinking Observing and Mining the Universe", Longo and Miele eds., World Scientific Press.
40. Amenomori, M., Ayabe, S., Caidong, et al. (Tibet ASGamma Collaboration) (2000). Primary Proton Spectrum Between 200 TeV and 1000 TeVObserved with the Tibet Burst Detector and Air Shower Array, astro-ph/0010292
41. Bussino, S. (1999). Gamma-Hadron Discrimination with a Neural Network in the ARGO-YBJ Experiment, astro-ph/9906201
42. Schaefer, B.M., Hofmann, W., Lampeitl, H., Hemberger, M. (2001). Particle Identification by Multifractal Parameters in Gamma-Astronomy with the HEGRA-Cherenkov-Telescopes, astro-ph/0101318
43. Razdan, A., Haungs, A., Rebel, H., Bhat, C. L. (2002). Image and Nonimage Parameters of Atmospheric Cherenkov Events: a Comparative Study of their gamma ray/hadron Classification Potential in Ultrahigh Energy Regime, Astroparticle Physics, **17**, 497–508
44. Barone, F., De Rosa, R., Eleuteri, A., Garufi, F., et al. (1999). A Neural Network-based ARX Model of Virgo Noise, in 11-th Italian Workshop on Neural Networks – WIRN, M. Marinaro and R. Tagliaferri eds., Springer-Verlag, pp. 171–183
45. Acernese, F., Barone, F. De Rosa, M., De Rosa, R., Eleuteri, A., Milano, L., Tagliaferri, R. (2002). A Neural Network-based Approach to Noise Identification of Interferometric GW Antenna's: the Case of 40m Caltech Laser Interferometer, Classical and Quantum Gravity, **19**, 3293–3307
46. Smirnov, D. (2002). A Neural Net Technique for the Higgs Search at the Tevatron, in Proc. of the Meeting of the American Physical Society
47. Bishop, C.M. (1995). Neural Networks for Pattern Recognition, Clarendon Press: Oxford

48. Kohonen, T. (1995). Self-Organizing Maps, Springer: Berlin
49. Vesanto, J. (1997). Data Mining Techniques Based on the Self-Organizing Map, Ph.D. Thesis, Helsinki University of Technology
50. Oja E., Ogawa H., Wangviwattana J., 1991, Learning in nonlinear constrained Hebbian network. In T. Kohonen et al. (Eds.), Artificial neural networks, North-Holland, Amsterdam, p. 385
51. Oja E., 1982, Journal of Mathematical Biology, 15, 267
52. Karhunen J., Joutsensalo J., 1994, Neural Networks 7, 113
53. Karhunen J., Joutsensalo J., 1995, Neural Networks, 8, 549
54. Sanger T. D., 1989, 2, 459
55. Staiano A. (2003). Ph. D. Thesis. "Unsupervised neural networks for the extraction of scientific information from astronomical data" University of Salerno
56. Koo D.C., 1999, astro-ph/9907273
57. Fernandez-Soto A., Lanzetta K.A., Chen H.W., Pascarelle S.M., Yakate N., 2001, ApJSS, 135, 41
58. Massarotti M., Iovino A., Buzzoni A, 2001a, AA, 368, 74
59. Massarotti M., Iovino A., Buzzoni A., Valls-Gabaud D., 2001b, AA, 380, 425
60. Connolly A.J., Csabai I., Szalay A.S., Koo D.C., Kron R.G., Munn J.A., 1995, AJ, 110, 2655
61. Wang Y., Bachall N., Turner E.L., 1998, AJ, 116, 2081
62. Brunner R.J., Szalay A.S., Connolly A.J., 2000, ApJ, 541, 527
63. Stoughton C., Lupton R.H., Bernardi M., Blanton M. R., et al., 2001, AJ, 123, 485
64. http://www.stsci.edu/science/goods/
65. P.Rosati, P.Tozzi, R.Giacconi, R.Gilli, G.Hasinger, L.Kewley, V.Mainieri, M.Nonino, C.Norman, G.Szokoly, J.X.Wang, A.Zirm, J.Bergeron, S.Borgani, R.Gilmozzi, N.Grogin, A.Koekemoer, E. Schreier, W.Zheng (2002). Astrophys. J. Suppl., 139, 369-410
66. Bertin, E., Arnouts, S. (1996). SExtractor: Software for Source Extraction. Astronomy and Astrophysics Supplement, **117**, 393–404
67. Tagliaferri, R., Pelosi, N., Ciaramella, A., Longo, G., Milano, M., Barone, F., Soft Computing Methodologies for Spectral Analysis in Cyclostratigraphy, Computer and Geosciences, volume 27, issue 5, pag. 535-548, June 2001
68. Deeming, T.J. (1975). Fourier Analysis with Unequally-Spaced Data, Astrophysics and Space Science, **36**, 137–158
69. Horne, J. H., Baliunas, S. L. (1986). A Prescription for Period Analysis of Unevenly Sampled Time Series, Astrophysical Journal, **302**, 757–769
70. Scargle, J.D. (1982). Studies in Astronomical Time Series Analysis: Statistical Aspects of Spectral Analysis of Unevenly Spaced Data, Astrophysical Journal, **263**, 835-853
71. Oppenheim, A.V., Schafer, R.W. (1989). Discrete-Time Signal Processing, Prentice-Hall: Englewood Cliffs–NJ
72. Kay, S.M. (1988). Modern Spectral Estimation: Theory and Application, Prentice-Hall: Englewood Cliffs–NJ
73. Lomb, N.R. (1976). Least-squares Frequency Analysis of Unequally Spaced Data, Astrophysics and Space Science, **39**, 447-462

74. Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press: Cambridge, p. 525 (1992)
75. Karhunen, J., Joutsensalo, J. (1994). Representation and Separation of Signals Using Nonlinear PCA Type Learning, Neural Networks **7**, 113-127
76. Karhunen, J., Joutsensalo, J. (1995). Generalization of Principal Component Analysis. Optimization Problems and Neural Networks. Neural Networks, **8**, 549-562
77. Oja, E., Karhunen, J., Wang, L., Vigario, R. (1996)., in VII Italian Workshop on Neural Networks, Marinaro M. Tagliaferri R. eds, IAU Symp. N. 161, World Scientific: Singapore 16–20
78. Sanger, T.D. (1989). Optimal Unsupervised Learning in a Single Layer Linear Feedforward Network, Neural Network, **2**, 459-473
79. Binnendijk, L., The Light Variation and Orbital Elements of U Pegasi, The Astronomical Journal, vol. 65, n.2, March 1960
80. Binnendijk, L., Synthetic Parameters of Five W-Type Contact Binaries, Publications of the Astronomical Society of the Pacific, vol. 96, pp. 646-656, August 1984
81. Wenxian, L., A New Spectropic Orbit of the WURSAE Majoris System U Pegasi, Publications of the Astronomical Society of the Pacific, vol. 97, pp. 1086-1091, November 1985
82. Zhai, D., Leung, K.-C., Zhang, R., A New Photoeletric Investigation of the W UMa System U Pegasi, Astronomy & Astrophysiscs Supplement Series, vol. 57, pp. 487-494, 1984
83. Lafta, S. J., Grainger, J. F., A Photoeletric Study of the W UMa-System U Pegasi, Astrophisycs ans Space Science, vol. 121, pp. 61-81, 1986
84. Rigterink, P.V., Astrophysical Journal, vol. 77, p. 319, 1972
85. Privitera P. for the AUGER collaboration, The Pierre Auger Observatory - Design Report (2002). Nuclear Physics Proc. Suppl. 110, 487
86. Heck D., Knapp J., Capdevielle J.N., Schatz G., Thouw T., COR-SIKA (1998). FZKA 6019, Forshungszentrum Karlsruhe (http://www-ik3.fzk.de/ ~ heck/corsika