Neural Network Analysis of leptonic τ decays

E. Sánchez Alvaro

October 11, 1994

Abstract

In this note measurements of the τ lepton leptonic branching ratios, universality checks and a measurement of α_s are presented. The events have been selected from the 1992 sample collected by the L3 detector at LEP, and analysed using neural network method.

1 Introduction

1.1 Classification problems in particle physics

A classification problem can be defined, in general, as follows:
"Given an object o, one wants to associate it to a class among the set $C_1, C_2, ..., C_r$*, according to some properties characterizing the objects and the class itself".*
Ussually, the object must be described through a finite number n of parameters which are useful to distinghish among classes. These parameters are the results of some measurements. The n quantities define a vector x in a n-dimensional space, the pattern space P. In the most general case, the classes may share patterns, but the main interest is in disjoint classes. Then, the event classification task is reduced to do a partition of pattern space P in disjoint regions, one region for each class.
This problem appears repeatedily in particle physics, from low level trigger decisions to the extraction of theoretically relevant parameters from collected data. The variables that describe an event should be selected from the information given by the detectors. In this way, particle physics becomes a natural application of the ideas developed in the study of classification problems.

1.2 Procedure for pattern classification

The standard procedure for pattern classification in particle physics is the introduction of relevant cuts in the variables describing the events. These methods have been applied successfully up to date, but during the last years, the increasing of computing power has allowed the develop of new non-linear statistical and classification methods, like parametric methods, non-parametric methods, multivariate statistics, decision tree induction, neural networks, ... (see ref. [2]).
In many of these methods, the classification task is not done in the pattern space P, but in a more convenient space F, called the feature space. Each event x in P is mapped to a vector z in F. F is a m-dimensional space, and the transformation between P and F can be linear or non-linear. The goals of this transformation are [3]:
- To reduce the dimensionality of the working vector space ($m < n$), without loss of significant information.
- To obtain vector components which are more suitable than the initial ones for pattern classification.
The transformation is done by a features extractor E, whose goal is the reduction of the complexity in the pattern classification task. After the features extraction, a classifier C maps F to the decision space D, and it does the classification.
The whole process can be mathematically described by a set of mappings:

$$\begin{array}{ccccccc} \mathrm{O} & \xrightarrow{T} & P & \xrightarrow{E} & F & \xrightarrow{C} & D \\ \mathrm{o} & \mapsto & x & \mapsto & z & \mapsto & y \end{array}$$

Obviously, this description of the problem is arbitrary and the whole process can be seen

as the global map $f : O \rightarrow D$ [2], [4].
There are some general conditions for the feature extraction and classification. The features should be invariant (or at least insensitive) to small traslations, rotations, changes of scale, etc. of the initial variables, while they must improve the differences that are important in order to do the discrimination among events of different classes. Assume that a sufficiently large amount of information x_i has been collected for each event. All the information is contained in the quantities x_i, but some of them can be useless or less important than others. The feature selection methods look for a smaller number of variables either keeping a subset, or applying linear or more complex combinations of the original variables [3], [2], [5].
After the features set has been defined, the classifier must be designed. The perfect classifier is such that all the patterns are associated to the correct class. Unfortunately, this is only possible in very simple cases. Besides, a good classifier for a set of patterns can be bad for another set of patterns. Thus, the classification problem becomes to statistical decision theory. The theory tells us that the best classifier is the Bayesian one. This classifier identifies classes through their probability density functions on the input features. It uses the Bayes rule to decide the class asociated to the given input. If we want to use the Bayes classifier, we must to know the probabilities and the conditional densities, however this information is not available on most classification problems. On the other hand, those probabilities can be estimated from sample patterns of each class.

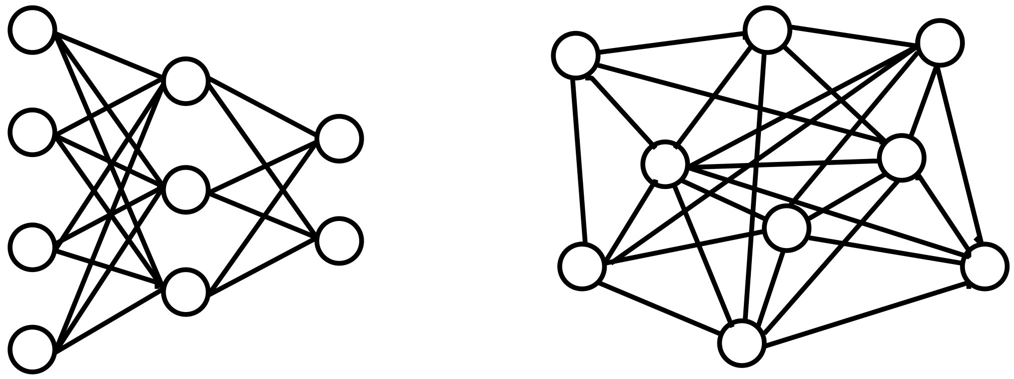

Figure 1: Feed-forward neural network (left) and self-organizing neural network (right).

2 Neural Networks

2.1 What is a Neural Network?

The most general definition that one can give is as follows:
A neural network is a set of processing units that are interconnected. The connections carry numerical information (as opposite to symbolic) while the units work with their local data and with the inputs they receive through the connections.
The difference between neural networks and other mathematical methods is the design. In principle, neural networks can approximate any computable function. So, they are very suitable for the task of mapping between vector spaces. Besides, it has been shown that neural networks have the capability to generalize data samples, even if the measurements are affected by experimental unbiased random errors.
The use of neural networks is appropiate when dealing with classification problems. The procedure is a functional fit to the data that provides a map f between the pattern space P and the decision space D, $y_i = f(x_k)$, fitting f to a set of M known training patterns $(x_k^{(p)}, t_i^{(p)}, p = 1, ..., M)$. When all the parameters of f are fixed, this parametrization can be used to interpolate and find the features of the test pattern sets (different from the training set). The neural network procedure is a choice of the function f. It is an expansion on units with a network structure.

2.2 Types of Neural Networks

According to their architecture, there are two basic types of neural networks: feed-forward networks and Self-organizing networks [6].
Feed-forward networks are a set of ordered layers. Each layer can be numbered from an input layer to an output layer. Self-organizing networks do not have this structure, and their topology is completely free. Each type of network could be represented as shown in figure 1.

Feed-forward networks will be described in more detail, because these will be the networks used in this work.

2.2.1 Feed-forward Neural Networks

As said before, the topology of these networks is a set of ordered layers, with unidirectional connections. Usually, the units are not connected to those on their own layer, but each one is connected to all units in the next layer. Networks with only one hidden layer will be introduced, but the number of hidden layers is not limited at all, and the generalization is easy.

FUNCTIONAL FORM: Each neuron in the network sums its inputs and adds a constant (the bias). After this, it applies an activation function ϕ. This is done in each layer. The links have weights ω_{ij} that multiply the signals which are travelling along the link. Input units have $\phi \equiv 1$. Then, the network can be written as:

$$y_k = \phi(\alpha_k + \sum_{j \to k} \omega_{jk} \phi_j(\alpha_j + \sum_{i \to j} \omega_{ij} x_i)) \tag{1}$$

Usually, the ϕ functions are the same in all the units in a layer. In general, the activation functions in the hidden layer are assumed to be logistic:

$$\phi(x) = \frac{1}{1 + e^{-\frac{x}{\theta_0}}} \tag{2}$$

while the activation functions in the output layer are logistic, linear or threshold (Heaviside function). The θ_0 parameter, which is called the temperature of the network, can be reabsorbed by a redefinition of weights and biases, so, generally it is taken to be one.
This structure implies that the parameters used to parametrize the function to approximate are the weights ω_{ij} and the biases α_i. As will be seen later, there are some results about the power of these structures for aproximation purposes. However, the proofs are not constructive. So, it is know that there are optimal weights and biases, but their values are unknown, and to find these values a learning algorithm must be applied.

LEARNING: A quantitative formulation of this problem is to find the appropiate network configuration $\{\omega_{ij}, \alpha_i\}$, that implements the desired mapping. This problem requires a measurement of the quality of the approximation, that is, how big are the diferences between the network and the mapping to be approximated. The disagreement can be measured as follows [2],[7]. Given a subset of inputs $x^{(\mu)}, 1 \leq \mu \leq m$, for which the desired output $t^{(\mu)}, 1 \leq \mu \leq m$ is known, and given the outputs produced by the network $y^{(\mu)} = f(x^{(\mu)}$, the distance[1]

$$d^{(\mu)} = d(y^{(\mu)}, t^{(\mu)}) \tag{3}$$

[1]The term 'distance' is not used here in its rigorous mathematical sense because simmetry and triangular inequality are not required

between the target $t^{(\mu)}$ and the output $y^{(\mu)}$ measures the error made by the network processing the input $x^{(\mu)}$. Then, the error function

$$E \equiv \sum_{\mu=1}^{m} d^{(\mu)} = \sum_{\mu=1}^{m} d(y^{(\mu)}, t^{(\mu)}) \tag{4}$$

measures the difference between both mappings in the domain $\{x^{(\mu)}, 1 \leq \mu \leq m\}$. For a fixed set of pairs $\{x^{(\mu)}, t^{(\mu)}, 1 \leq \mu \leq m\}$, E is an implicit function of the weights $\{\omega_{ij}\}$ through the outputs $\{y^{(\mu)}\}$.
There are many choices for $d(y^{(\mu)}, t^{(\mu)})$. Only one property is required:
$d(y^{(\mu)}, t^{(\mu)}) \geq 0$ and $d(y^{(\mu)}, t^{(\mu)}) = 0$ iff $y^{(\mu)} = t^{(\mu)}$, which implies $E \geq 0$ and $E = 0$ iff $y^{(\mu)} = t^{(\mu)}\ \forall \mu,\ 1 \leq \mu \leq m$.
The learning problem becomes the search of global minima of $E(\omega_{ij})$. The configuration space is defined by the points that satisfy $E(\omega_{ij}) = 0$. The learning algorithm finds the desired information from the training set $\{x^{(\mu)}, t^{(\mu)}, 1 \leq \mu \leq m\}$.
There are many training methods [2], [7], [8], [9], but all of them can be classified in two fundamental types:
In supervised learning, there is a teacher who shows the net if it is performing well, or what should be the good behaviour.
In unsupervised learning, the network is independent and learns how to recognize some properties from the training set. These properties, that the network finds, are dependent on the model and training method.
This classification of training methods is independent of the net topology, but many of these methods are closely connected to a given architecture. For feed-forward networks, some learning algorithms are:
unsupervised learning: learning matrix, linear associative memory, fuzzy associative memory, counterpropagation,...
supervised learning: backpropagation, adaptive logic network, cauchy machine,...
The backpropagation algorithm that will be used in this job is described in the next paragraph.

BACKPROPAGATION: The most usual choices for the error function are:

$$E = \sum_{\mu} \|t^{\mu} - y^{\mu}\|^2 \tag{5}$$

which could be seen as a least squares fit, and

$$E = \sum_{\mu} \sum_{k} [t_k^{\mu} \log \frac{t_k^{\mu}}{y_k^{\mu}} + (1 - t_k^{\mu}) \log \frac{1 - t_k^{\mu}}{1 - y_k^{\mu}}] \tag{6}$$

or Kullback-Leibler distance (entropy) [10], that could be associated to a 'maximum likelihood'. These functions have the advantage that their derivatives respect to the weights can be computed recursively from output to input using the chain rule, hence, the name of backpropagation.

The algorithm to optimize the network parametersis based in a steepest-descent method [11], [10]. The weights are updated by

$$\Delta\omega_{ij}(t+1) = -\eta\frac{\partial E}{\partial \omega_{ij}} \tag{7}$$

thus,

$$\omega_{ij}(t+1) \leftarrow \omega_{ij}(t) - \eta\frac{\partial E}{\partial \omega_{ij}} \tag{8}$$

where η is a parameter (learning rate) that must be fixed empirically and t is a discrete index that counts the number of iterations.
The derivative of the error fuction can be written as:

$$\frac{\partial E}{\partial \omega_{ij}} = \sum_p y_i^p \delta_j^p \tag{9}$$

(the second name of the method, generalized delta rule, comes from this equation [1], [11]) and δ is computed from output to input trough the network.
Suppose that each unit has input x_j and output $y_j = f(x_j)$. Then:

$$\frac{\partial E}{\partial \omega_{ij}} = \frac{\partial E}{\partial x_j}\frac{\partial x_j}{\partial \omega_{ij}} = y_i\frac{\partial E}{\partial x_j} = y_i f_j^{'}(x_j)\frac{\partial E}{\partial y_j} = y_j\delta_j \tag{10}$$

For output units, $\frac{\partial E}{\partial y_j}$ can be computed easily, while for units in preceding layers one has

$$\delta_j = f_j^{'}(x_j)\frac{\partial E}{\partial y_j} = f_j^{'}(x_j)\sum_{j\to k}\omega_{jk}\frac{\partial E}{\partial x_k} = f_j^{'}(x_j)\sum_{j\to k}\omega_{jk}\delta_k \tag{11}$$

δ_j only depends on the parameters of the following layers, thus it is possible to compute it from output to input.
If one wants to finish the computation, one needs to write $f_j^{'}(x_j)$ in terms of y_j. For logistic units $f_j^{'}(x) = y(1-y)$. Thus, for an output unit o results:
$\delta_o = 2y_o(1-y_o)|y_o - t_o|$ logistic output unit, least squares
$\delta_o = |y_o - t_o|$ logistic output unit, entropy fit
$\delta_o = 2|y_o - t_o|$ linear output unit, least squares
In the update of weights a 'momentum term' is often used in order to smooth the derivatives. This is,

$$\Delta\omega_{ij}(t+1) = -\eta\sum_{patterns}\delta_j y_o + \alpha\Delta\omega_{ij}(t) \tag{12}$$

It is very important to show the training patterns randomly. The α parameter is called 'momentum' (or inertia) and $0 \le \alpha \le 1$ and its value is computed empirically. A usual choice is $\alpha = 0.9$.
The values of biases α_i should be computed too. This can be done easily, considering that α_j is like the weight from an extra input unit, whose output is constant and equal to one, to a unit j.

A difficulty of this method is that it finds a critical point of E, but this point can be a local minimum and not the global one. The initial values of weights and biases must be chosen near the best ones, and then the best local minimum of E should be searched. In practice, one takes random initial values for weights and biases, while η and α are chosen empirically.
Very little is known about the convergence behaviour of the backpropagation algorithm. Especially how it depends on the size and difficulty of the problem. The algorithm with a constant value for η does not minimize E, like the in stochastic approximation, one must do $\eta \to 0$ with the number of iterations. The computation is stopped when a fixed rate of the training patterns have been correctly classified.
When the network has been trained, its classification behaviour is studied in a test set different from the training one. The generalization problem is that one can not know if the relevant features are being generalized, and this point must ensure. There are some theoretical limits to the size of the training set in order to have a succes rate in the set ϵ times worse than in the training set. For feed-forward neural networks with threshold units, the training set size must be $\approx W/\epsilon$ where $W =$ number of weights [12]. For logistic units, no results are known.
The design problem (How many layers?, how many hidden units?) and the learning problem are very hard. It is true that the external knowledge about the problem can be used in order to choose a network topology, but this is very limited. Judd [14] showed that the problem of proving the existence of a network and a set of weights and biases for solving a given classification problem is NP-complete, so its complexity grows faster than polynomial (see section 2.3.2). Blum & Rivest [13] showed that even with n inputs, two hidden units and one output, the problem of finding the weights, when it is known that a set of them exists, is also NP-complete (they showed this for threshold units and it is suggested to be true for logistic units). Until obtaining more results about these problems , the only way to solve them is empirically for the design problem, and using backpropagation for the learning problem because this algorithm has been used successfully up to date.

THEORY: There are very little theoretical results about neural networks. Besides, most of them consist on representation by networks rather than about learning algorithms. The first result was a Kolmogorov theorem [15]:
Any continuous function $g : [0,1]^n \to I\!R^m$ can be represented exactly by a three-layer network, with the inputs connected to n each of $2n + 1$ elements in the hidden layer, and with m elements in the output layer.
But the network used in this theorem is not like the standard nets. For this kind of networks, there are no results about exact representation. But it can be asked if they can give us good approximations.
Cybenko [17], Funahashi [16], Hornik et al. [18] have showed that neural networks with linear outputs can approximate any continuous function f uniformly on compact sets, by increasing the size of the hidden layer. This result implies the approximation of measurable functions in measure in L_p. For discontinuous functions it is known that at

most two hidden layers are needed [19] but the approximation is not uniform.
The classification problem is a little easier because it allows discrete outputs. There is a formal proof that a neural network with one hidden layer of sigmoid units and one output layer of threshold units can represent exactly any boolean function [8]. However, I inputs can need a number of hidden units that grows exponentially with I. Very little is known about sigmoid units networks, except that threshold units are a limit case of sigmoid ones, then their power should be greater.
These results do not tell us neither the size of the net nor if it is possible to learn the weights. The proofs of the approximation results need that the number of hidden units tends to infinity in order to the error in the approximation tends to zero.

2.2.2 Self-Organizing Neural Networks

HOPFIELD NETS: A Hopfield type neural network [20], [1], [2], [21] is a N-dimensional dynamic system with a discrete time evolution. These networks are designed to be used as associative memories (they can recognize an object from a partial or deteriorated version). The network is fully connected, and the weights ω_{ij} are symmetric ($\omega_{ij} = \omega_{ji}$). Each neuron i at the time t has one of the two possible states $\{+1, -1\}$, so the network state can be described by a binary vector:

$$x(t) = (x_1(t), ..., x_N(t)) \tag{13}$$

where N is the number of neurons in the network, and by a weights matrix (ω_{ij}).
The neuron i fires at $t^{'} = t + 1$ according to a probability that is a function of the total input:

$$Prob(x(t+1) = +1) = g(\sum_{j=1}^{N} \omega_{ij} x_j(t) - \theta_i) \tag{14}$$

where g is a sigmoid function. The system evolution is described by

$$x_i(t+1) = sign[\sum_{j=1}^{N} \omega_{ij} x_j(t) - \theta_i] \tag{15}$$

A Hopfield net has neither input layer nor output layer,so, one has to simulate them by the initial and final network state,respectively. The targets are the fixed points of eq. (15). The set of all initial configurations that go to a given fixed point after the evolution is called "basin of attraction" of that fixed point (or memory state). Clearly, a wide enough basin of attraction around every pattern to be stored is a very good feature for the network. This ensures that all input patterns sufficiently close to the fixed point will be associated to it, and then the reconstruction from a partial or deteriorated information is possible.
To study the time evolution of a dynamical system and to prove the existence of fixed points, a classical method is to define an energy function for the system. The minima of this function are the fixed points (stable states). Hopfield networks have just the same evolution as a spin glass (Ising model). The energy function is then:

$$E(t) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\omega_{ij}x_i(t) + \sum_{i=1}^{N}\theta_i x_i(t) \tag{16}$$

and each pattern to be stored corresponds to a local minimum of E. This function has many local minima, so one has a very large storage capability.

BOLTZMANN MACHINES: Boltzmann machines were introduced by Hinton & Sejnowski [22]. They are stochastic models of binary random variables on a graph $\mathcal{G}$, and the selfconnections are allowed. Each neuron is valued on $\{0, 1\}$. The connection that links two neurons is said to be activated if both nodes have value 1. By definition, $\omega_{ij} = \omega_{ji}$. The consensus function is defined by

$$C(x) = \sum_{connections} \omega_{ij}x_i x_j \tag{17}$$

the evolution steps of a Boltmann machine form a Markov chain. A given node i changes its state with probability $\frac{1}{1+e^{-\frac{\Delta C(x)}{T}}}$ for a control parameter T. Note that $\Delta C(x)$ depends only on the random variables at neighbours on the graph of the chosen node.
In a Boltzmann machine used as a classifier, the nodes are separated in three groups: inputs,outputs and hidden (this one can be empty). Then, the network approaches its equilibrium with the input set to the pattern to be classified. The problem fixes the inputs and the outputs, but the weights and the topology are free. Usually, the weights are chosen to minimize the Kullback-Leibler function.
The Boltmann machines are extremely slow, sometimes too slow to be run on real problems, but they are very effective.

SELF-ORGANIZING NETS: The Kohonen self-organizing map [23] is considered as part of the neural network subject, although it is rather different. It is like a n-dimensional lattice.
When a self-organizing map is used as a classifier, some nodes are associated to a codebook made of vectors m_i in the pattern space. The codebook vectors are grouped in classes. Their positions in the classes are not important. Only boundaries between classes are considered. The Learning Vector Quantization fit the vectors in order to find boundaries very close to the best decision. The decision rule is nearest neighbour, so an input x belongs to the same class as the nearest m_i.

2.3 Neural Networks in Particle Physics

2.3.1 Event selection

This application of neural networks in particle physics is only a especial case of classification problems. The different step can be identified, in the particular case of a feed-forward neural network:

-The mapping T is the detection and measurement step. The x_i variables in the pattern space correspond to the available measurements for each event.
-The mapping E (feature extraction) is implemented from the input layer to the hidden layer. Here can be seen what is the physics that the network has chosen in order to do the classification.
-The mapping C (classification) is done from the hidden layer to the output layer. Here the network classifies the inputs.
The set of known patterns for trainnig and testing the neural network comes from the Monte Carlo simulation of the events and the detector. It is desirable to have a good simulation for the following reasons:
It enables calculate the efficiency of the selection and the background contamination in the sample, because both of them are computed from the test set.
If the simulation is good, there are no differences between the training sample and the real data sample, then the classification task is easier for the network.
If the MC simulation and the real data are different, there is a systematic error in the measurements, however the neural network procedure should be stable to small discrepancies.

2.3.2 Track Reconstruction

The track reconstruction problems have two fundamental features: First, algorithms to achieve global solutions are NP, and second, the reconstruction is a two stage process [24]. To give an idea of the complexity of the problem, notice that three or more labels can be associated to each measured point: new detection (ND), false detection (FD) or member of an existing track (T_i). Then for a set of two measured points, the cardinality of the hypothesis set is 3^2. In general, the cardinality of the hypothesis set for N measured points is 3^N. The growth of this quantity is exponential (non-polynomial, this is the reason why the problem is named NP).
The solution to the tracking problem is a two stage process. In the first step, the cardinality of the hypothesis set is reduced according to some criteria (tracks from trajectories of constant curvature, etc.). The second step assigns a probability to each surviving hypotheses and the hypothesis with the largest probability is selected as the solution of the problem.
Usually, a typical detector measures a set of N points in space along a track [1]. From each of these points, a segment can be defined:
- A segment $i \to j$ links two points i, j with $i \neq j$.
- The length of the segment r_{ij} must be smaller than some fixed value R_{cut}.
To apply the neural networks algorithm, a natural choice is to assign a binary neuron to each segment, whose state is one if the segment is in the track and zero if the segment is not in the track. This is the way to use Hopfield nets for this problem.
The fundamental point of the neural network approach is the possibility of implementing it as parallel hardware for a quick trigger. This approach can be used also for a secondary vertex trigger [1].

2.3.3 Neural Multigrid

This is a theoretical application of neural networks, in order to study the behaviour of some theories in the lattice [25], [26]. There is a stochastic multigrid and a deterministic one. The stochastic version is used to compute high dimensional integrals in euclidean quantum field theory or in statistical mechanics. The deterministic version resolves discretized partial differential equations. One hopes to use both of them for simulation of QCD in the lattice, in the updating of gauge fields and for the computation of fermion propagators for given gauge fields.

MULTIGRID: One starts with a problem in a fundamental lattice Λ^0 of lattice spacing a_0. One introduces a sequence of lattices $\Lambda^1, \Lambda^2, ..., \Lambda^N$ of increasing lattice spacings $a_j = L_b^j a_0$ and interpolation operators A^j and restriction operators C^j that map the lattices one into another.
The idea is to translate the problem to coarser lattices up to be trivial, and then come back using the A^j operators.
A feed-forward neural network can compute the problem. The units are identified with points in the multigrid, so the optimization of the operators is equivalent to the learning problem in the network.

2.4 Example

The neural network algorithm will be used for the identification of a signal distribution in a background. The signal will be events with distribution

$$f_{signal}(x,y) = \frac{1}{N_1} e^{-\frac{1}{2}((\frac{x}{\sqrt{2}}+\frac{y}{\sqrt{2}})^2+\frac{1}{4}(-\frac{x}{\sqrt{2}}+\frac{y}{\sqrt{2}})^2)} \tag{18}$$

and the background will be events with distribution

$$f_{background}(x,y) = \frac{1}{N_2} e^{\frac{1}{200}(x^2+y^2)} \tag{19}$$

where N_1 and N_2 are the normalization constants.
To identify the signal, a closed region in the plane must be chosen, task that, in fact, is absolutely non-trivial.
Two different methods will be used: neural networks and cuts in the two variables. The most important difference between the two methods is that neural network will take into account the correlation among variables automatically while the cuts will not. [2]
The neural network used in this problem is a feed-forward one, with two nodes in the input layer, four nodes in the hidden layer and one node in the output layer. The activation function is sigmoid both in hidden and output layer, and the units are linked to all the

[2] The obvious objection, that the correlation can be handle by hand before applying the cuts, can become imposible or extremelly difficult if the pattern space dimensionality is large or if the correlations are non linear.

units in the next layer. The public domain software Aspirin/Migraines 6.0 [38] has been used in all this work.
10000 patterns have been used for training, 5000 for signal with target one and 5000 for background with target zero.The training algorithm is backpropagation. During the training the learning rate decreases from 0.2 to 0.1 and the momentum is 0.9. The training was stopped at 263992 iterations. The test sample has also 10000 patterns, 5000 for signal and 5000 for background.
The applied cuts to the test sample were $-x_{cut} \leq x \leq x_{cut}$ and $-y_{cut} \leq y \leq y_{cut}$. For each value of x_{cut}, the value of y_{cut} was varied in a wide range.
The following definitions are considered:
-Efficiency= N^o of selected signal events/Total n^o of signal events
-Purity= N^o of selected signal events/Total n^o of selected events

The relation Efficiency versus Purity for the two approaches is shown in figure 2. Cuts are applied by taking a value of x_{cut} and varying y_{cut}. This implies that there are different curves for cuts. The neural network is better than cuts because for a given purity, its efficiency is higher. This is the consecuence of taking into account the correlation among variables in the pattern space. But if a very high purity must be reached , there can be some problems for neural networks. This is due to training algorithm. Signal is required to map to one and background to zero, but the network is only an approximation, so the neural net maps background very close to zero and signal very close to one but not exactly. When a sample with very high purity is required, the output must be chosen as close to one as possible and some problems can appear. This can be improved with better learning algorithms.
In figure 3 it can be seen how the neural network identifies the signal. Each unit in the hidden layer can be associated to a hyperplane in the pattern space , see eq.(1): these hyperplanes are the points in the pattern space which are mapped to 0.5 in each unit of the hidden layer.
This is a dual interpretation of the feature extraction. The network has foundd the correlation for choosing the hyperplanes.
If the number of neurons in the hidden layer is increased (up to 50) there is no improvement in the classification. Remember that it was suggested that the size of the hidden layer must increase exponentially to improve the approximation. Besides, the efficiency and purity have no dependence on the signal to background ratio in the training sample. Furthermore, if some extra random variables for each event are introduced (up to 4) the efficiency versus purity function does not change.
Finally, it is important to note that, if the network is trained with the training sample, a very accurate measurement of the fraction of the signal in a given test set can be made. This allows to measure the branching ratios in some particle physics problems.

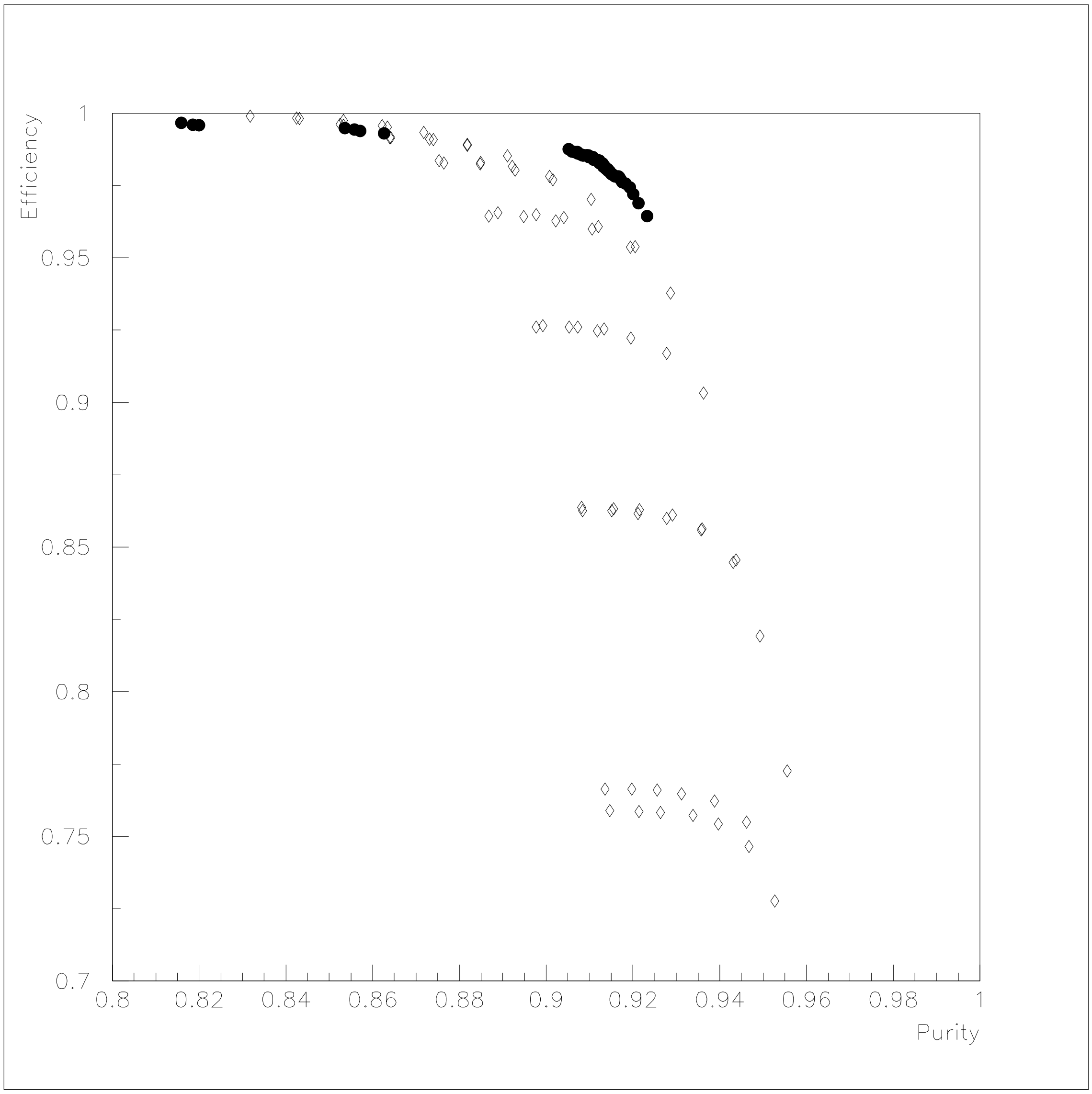

Figure 2: Efficiency versus Purity for the example discussed in the text. Dots are the results of the neural network technique and the diamonds are the results of the approach based on imposing selection cuts.

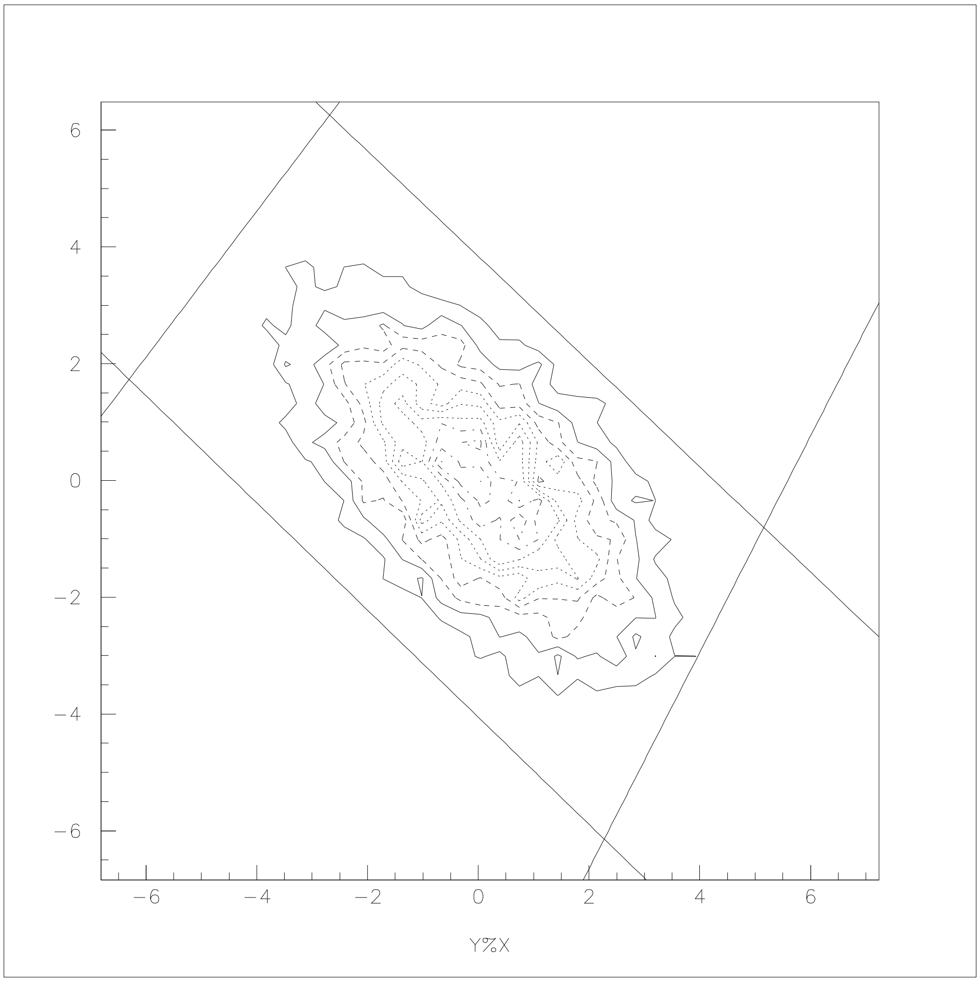

Figure 3: Selection in the hidden layer. The figure is a contour plot of the distribution signal and the lines are associated to the hidden units. The neural network selects a closed region in the plane.

3 The L3 experiment

L3 is one of the four experiments at LEP. During its first five years of operation it has cover a wide range of physics topics checking many predictions of the Standard Model and providing the determination of its parameters [27]. The aim at present for the L3 collaboration is to accumulate the largest possible statistics and to improve the understanding of the detector and analysis techniques to reach higher precision of the Standard Model parameters measurements.
The L3 detector [27], [29] is placed in the experimental area P2 of the LEP collider. Its fundamental goal is the precise measurement of photons and leptons with energies up to 50 GeV. Figure 4 shows a general view of the L3 detector, which consists in a large cylinder of 12 m long and centred on the interaction point with its axis along the beam and 6 m of radius.
The main components of L3 are, following a particle which escape from the interaction point:
- The beam pipe of 8 cm. of radius.
- The vertex chamber, a very precise drift chamber that works on Time Expansion Mode. It reaches 50 cm radius.
- The electromagnetic calorimeter, made of a very dense crystal scintilator (BGO), that is a homogeneus barrel up to 80 cm radius.
- The plastic scintillator counters, that work for TOF and trigger purposes. They are about 2 cm thick.
- The hadronic calorimeter, made of alternate uranium plates and proportional chambers. This calorimeter reaches the 170 cm radius.

The set of the previous subdetectors make up the inner detector, which is placed into a huge steel tube, called the support tube.
The luminosity system, based in BGO crystals and drift chambers, is also situated inside this tube.
Around the support tube and up to 5.7 m radius, there is the muon spectrometer. It consists of three layers of precision drift chambers.
Finally, and including all the previous detectors inside, is placed the magnet that using conventional technology provides a magnetic field of 0.5 T parallel to the beam direction.
Figure 5 shows how the different particles are observed in the detector:
-PHOTONS: They cross the vertex chamber without interaction, and then produce an electromagnetic shower contained in the BGO calorimeter. This subdetector provides a very precise information about photon energy and good measurements of the azimuthal and the polar angles.
-ELECTRONS: The electrons are observed as a track in the vertex chamber pointing to a shower in the BGO, completely contained in this calorimeter. The combined information provides very precise measurements of energy, orientation and charge identification.
-HADRONS: As in the case of electrons, they might give tracks in the TEC (if they are charged) and showers in the BGO, but a part of their cascade goes out from the BGO and it is measured in the hadron calorimeter. Their energy is not measured as precisely as

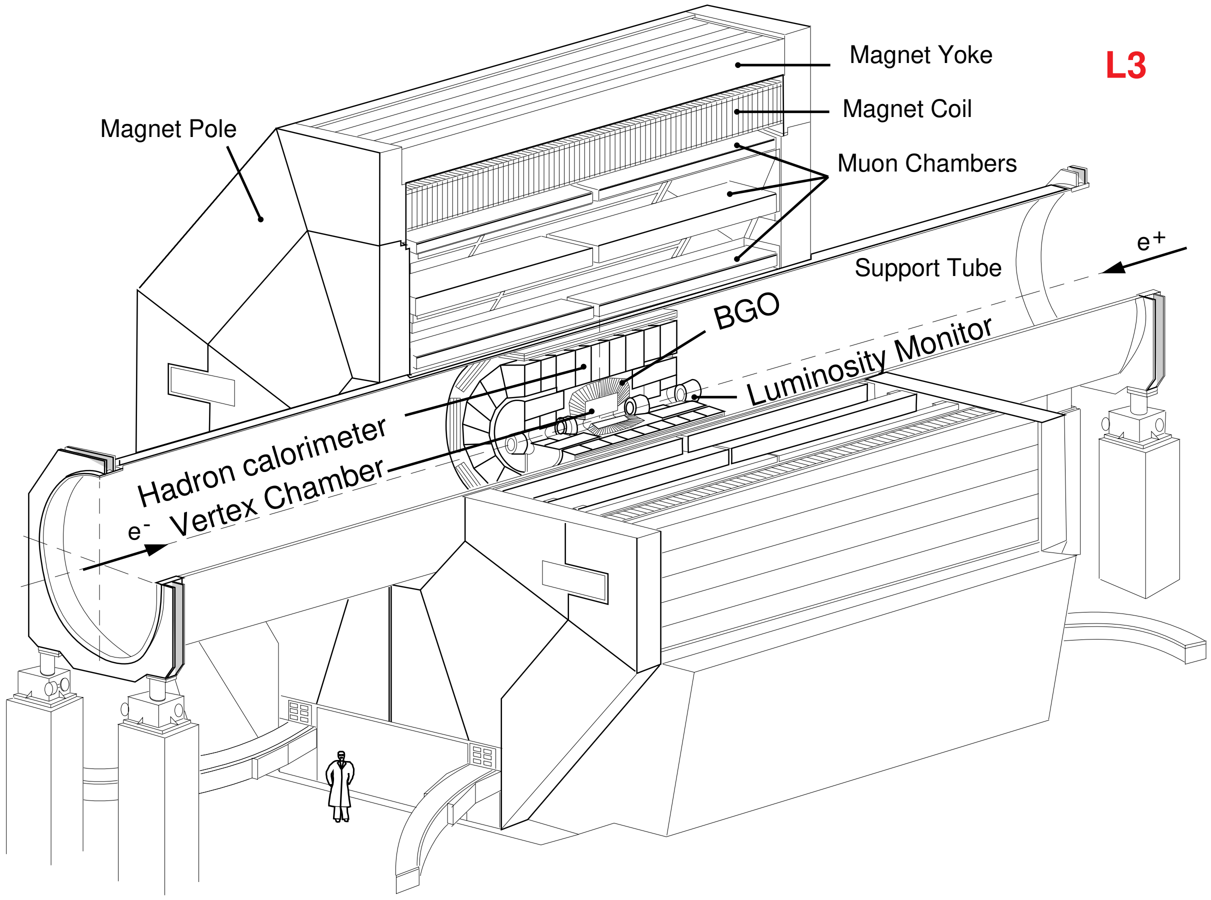

Figure 4: L3 detector view

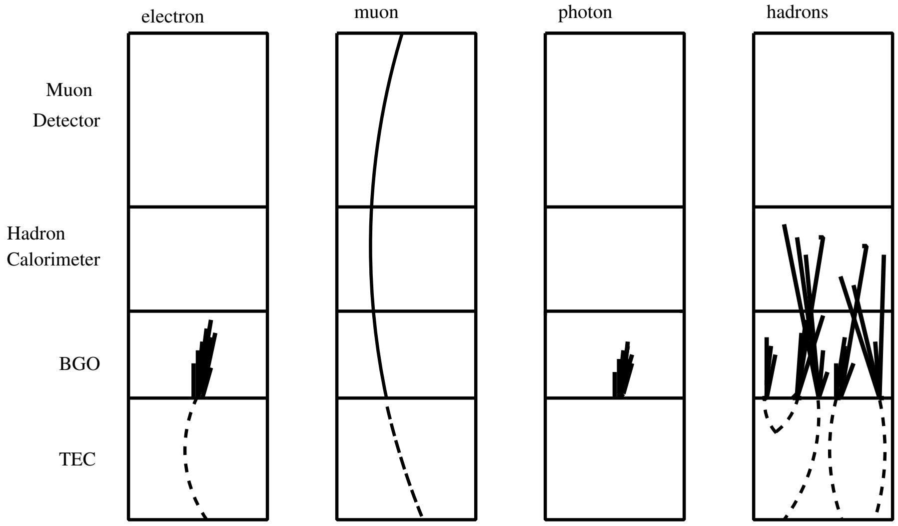

Figure 5: view of events in L3 detector.

the electrons one, but a very good solid angle coverage is reached, and there are enough absortion lengths to fully contain the complete hadronic cascade.
-MUONS: They are particles of minimum ionization in all the detectors, so they produce a track in the vertex detector, that provides the angular information. They also leave some energy in the BGO calorimeter, and produce ionizatiob in the hadronic calorimeter, what allows to follow the trajectory of the muons. Finally, muons are the only particles reaching the muon spectrometer, exception made for neutrinos, where their momenta is measured with precision.

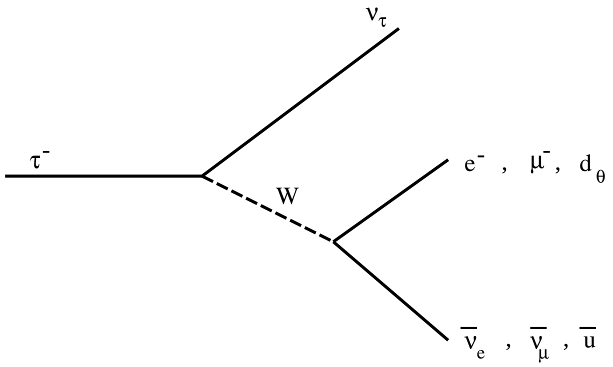

Figure 6: First order Feynman diagram for τ decays

4 The τ lepton

The τ particle is the heaviest known lepton. It is the only one that may decay into hadrons. Its decays are a very good framework to test the structure of the weak currents, the universality of their couplings to gauge bosons and also some aspects of strong interactions. In the Standard Model, the τ lepton decays through the Feynman diagram shown in figure 6. The coupling of the W boson to a charged current is universal, with the following expression

$$L_{cc} = \frac{g}{2\sqrt{2}} W_\mu^+ \{ \sum_l \bar{\nu}_l \gamma^\mu (1-\gamma_5) l + \bar{u} \gamma^\mu (1-\gamma_5) d_\theta \} + h.c. \tag{20}$$

and there are five equal contributions to the τ^- decay width. Two of these contributions come from $e\bar{\nu}_e\nu_\tau$ and $\mu\bar{\nu}_\mu\nu_\tau$ decays and three of them come from the possible colours of the quark-antiquark pair in the final state $\nu_\tau \bar{u} d_\theta$, ($d_\theta = \cos\theta_c d + \sin\theta_c s$, θ_c is the Cabbibo angle). Then, in the first order approximation, the expected leptonic branching ratios are:

$$Br(\tau \to l\bar{\nu}_l\nu_\tau) \approx \frac{1}{5} = 20\% \tag{21}$$

4.1 τ leptonic decays and universality

The partial width for τ leptonic decays can be easily computed, with the result

$$\Gamma(\tau \to \nu_\tau l \bar{\nu}_l) = \frac{G_F^2 m_\tau^5}{192\pi^3} f\left(\frac{m_l^2}{m_\tau^2}\right) r \tag{22}$$

where it has been assumed that neutrinos are massless, $f(x) = 1 - 8x + 8x^3 - x^4 - 12x^2 \ln x$, and the r factor includes the radiative corrections. These corrections are small [30], ($\alpha = \alpha(m_\tau) \approx 1/133.3$):

$$r \approx [1 + \frac{\alpha}{2\pi}(\frac{25}{4} - \pi^2)][1 + \frac{3}{5}\frac{m_\tau^2}{m_W^2}] \approx 0.9960 \tag{23}$$

Precise measurements of the τ leptonic branching ratios and the τ lifetime can be used to test the $e - \mu - \tau$ universality. If the coupling constant is allowed to depend on the lepton flavour the ratio of the leptonic widths can be written as:

$$\frac{Br(\tau \to \nu_\tau \mu \bar{\nu_\mu})}{Br(\tau \to \nu_\tau e \bar{\nu_e})} = \left(\frac{g_\mu}{g_e}\right)^2 f\left(\frac{m_\mu^2}{m_\tau^2}\right) \tag{24}$$

In a similar way, using the $\tau \to \nu_\tau e \bar{\nu_e}$ and $\mu \to \nu_\mu e \bar{\nu_e}$ widths, the $\tau - \mu$ lepton coupling universality can be tested

$$Br(\tau \to \nu_\tau e \bar{\nu_e})\frac{\tau_\mu}{\tau_\tau} = \left(\frac{g_\tau}{g_\mu}\right)^2 \left(\frac{m_\tau}{m_\mu}\right)^5 \frac{r}{r_\mu} \tag{25}$$

where τ_μ, τ_τ are the lifetimes and r_μ is the correction factor in muon decay (with the same expression of equation 23, changing m_τ to m_μ).

4.2 $\Lambda_{\overline{MS}}$ and α_s measurement

The measurement of α_s at low energies may be competitive with the high energies measurements. For instance, a measurement with a 10% error rate at 1.7 GeV corresponds to a measurement with a 4% error at the Z^0 mass, due to the behaviour of the evolution of the coupling. The inclusive hadronic decay rate of the τ lepton expressed as the ratio

$$R_\tau = \frac{\Gamma(\tau \to \nu_\tau + hadrons)}{\Gamma(\tau \to \nu_\tau e \bar{\nu_e})} \tag{26}$$

can be used to test QCD and gives a measurement of α_s at low energies [31].
This ratio has electroweak and QCD (both perturbative and non-perturbative) contributions [31], [32]:

$$R_\tau = 3 s_{ew}(1 + \delta_{PQCD} + \delta_{NPQCD}) \tag{27}$$

where $s_{ew} = 1.0215 \pm 0.0050$ is a known electroweak correction which includes the mixing factor ($\mid V_{ud} \mid^2 + \mid V_{us} \mid^2$). The perturbative QCD contribution is known up to third order [33], [34], [32] in the $\overline{MS}(\mu = m_\tau)$ renormalization scheme:

$$\delta_{PQCD} = a + 5.20a^2 + 26.37a^3, a \equiv \frac{\alpha_s}{\pi} \tag{28}$$

The non-perturbative contribution has been estimated too [35], [36], and it is small:

$$\delta_{NPQCD} = -0.007 \pm 0.004 \tag{29}$$

The largest source of theoretical uncertainty is the higher order contribution of δ_{PQCD}.
From the measurement of the τ leptonic branching ratios, R_τ can be derived.

$$R_\tau = \frac{1 - Br(\tau \to \nu_\tau e \bar{\nu}_e) - Br(\tau \to \nu_\tau \mu \bar{\nu}_\mu)}{Br(\tau \to \nu_\tau e \bar{\nu}_e)} \tag{30}$$

and from R_τ, $\alpha_s(m_\tau)$ can be computed. Using the expression for α_s running, $\Lambda_{\overline{MS}}$ and $\alpha_s(m_{Z^0})$ can be extracted.

5 Selection of $e^+e^- \to \tau^+\tau^-$

From here to the end, the neural network method will be applied to select leptonic decays of τ. The aim is to measure the leptonic branching ratios of the τ. The analysis will have two parts: in the first one $e^+e^- \to \tau^+\tau^-$ events are selected, and in the second one the τ leptonic decays will be identified. All the selections will be made according to the neural network method.
The sample used in this work corresponds to the data collected by the L3 detector in 1992, with a total integrated luminosity of 21.6 pb^{-1}. The events have been selected after some preselection cuts that remove most of the cosmic rays, two photon processes and $e^+e^- \to$ hadrons background. After preselection, the data sample includes more than 98% of the charged leptonic Z^0 decays and a background of 5% mainly from two photon interactions and hadronic events [37],[27].
The selection is based on calorimetric and tracking information. Data collected in a fiducial volume $|\cos\theta_{th}| < 0.76$ will be used, where θ_{th} is the polar angle of the event thrust axis. Each event is divided in two hemispheres by a plane perpendicular to the thrust axis.

5.1 Variables used for event selection

The following variables have been used for event selection:

1) N_1: Number of tracks inside a 20° of semiaperture cone around the thrust axis in hemisphere 1 (the most energetic one).
2) N_2: Number of tracks inside a 20° cone around the thrust axis in hemisphere 2.
3) N_T: N_1+N_2.
4) N_{had}:Number of tracks outside the previous cones.
5) p_{tmax}: Maximum transverse momentum measured in the TEC.
6) N_{bumps}: Number of BGO bumps (clusters of BGO crystals with deposited energy).
7) $\cos\theta_{th}$
8) p_{tsum}: Sum of transverse momenta of tracks inside a 20° cone around thrust axis.
9) E_{BGO}: Total energy deposited in the BGO calorimeter.
10) Acopl: Acoplanarity, $||\phi_1 - \phi_2| - \pi|$, where ϕ_1 and ϕ_2 are the azimuthal angles of the track with largest momentum in hemisphere 1 and 2 respectively.
11) Acol: Acolinearity,
$\arccos(-\sin(\theta_1)\sin(\theta_2)\cos(\theta_1-\theta_2)+\cos(\theta_1)\cos(\theta_2)$ where θ_1 and θ_2 are the polar angles of the track with largest momentum in hemisphere 1 and 2 respectively.
12) $\max(E_{BGO}+p_\mu)$: For each hemisphere the sum of the total energy collected in that hemisphere and the largest momentum measured in the muon chambers in the same hemisphere has been computed. The maximum of the two hemispheres is selected.

These variables were chosen folowing a previous analyses [37], [27] which have shown that these quantities contain enough information for the identification of $e^+e^- \to \tau^+\tau^-$ events. The variables have been normalized to be enter to the neural network, to simplify the

Input variable	Normalization
1	$N_1/10.$
2	$N_2/10.$
3	$N_T/10.$
4	$N_{had}/10.$
5	$p_{tmax}(GeV)/1000.$
6	$N_{bumps}/10.$
7	$\cos\ \theta_{th}$
8	$p_{tsum}(GeV)/1000.$
9	$E_{BGO}(GeV)/1000.$
10	Acopl. (mrad)
11	Acol.
12	$\max(E_{BGO} + p_\mu(GeV))/100.$

Table 1: Normalization of input variables.

training phase, according to the table 1.

5.2 The network structure

For this analysis a feed-forward neural network with an input layer of 12 units, a hidden layer of 6 units and an output layer of 1 unit has been used.
The activation functions have been chosen to be sigmoid, both in the hidden layer and in the output layer. Each unit is connected to all units in the following layer. The target is 1 if the event is $e^+e^- \to \tau^+\tau^-$ and 0 in any other case.
The number of units in the input layer is fixed by the number of variables used in the analysis. The number of output units is one because there is only one cathegory ($e^+e^- \to \tau^+\tau^-$) of good events. Six hidden units had been used by applying the following thumb rule: *the number of hidden units can be chosen to be the average of the number of input units and the number of output units* [2]. The possible influence of changing this number wil be described later.

5.3 The training

91059 MC events have been used for the network training. The events have been showed in random order to the network, to avoid generalization troubles. The composition of this sample was 32.94% of $e^+e^- \to \tau^+\tau^-$ events, 21.96% of $e^+e^- \to e^+e^-$ events, 21.96% of $e^+e^- \to \mu^+\mu^-$ events, 14.87% of two photon events and 8.26% of $e^+e^- \to q\bar{q}$ events.
During this training, the learning rate η decreases from 0.2 to 0.0001 and the momentum α has been fixed at 0.9. A total of 1330967 backpropagation iterations were required by the network before reaching convergence.

input	unit 1	unit 2	unit 3	unit 4	unit 5	unit 6
N_1	-0.41	6.46	-7.38	-3.00	-0.26	-1.20
N_2	-1.52	7.92	2.85	14.44	-1.10	2.61
N_T	-2.00	14.19	-4.58	11.45	-1.32	1.33
N_{had}	-3.16	2.53	32.29	48.34	-2.03	-54.38
p_{tmax}	-7.15	-64.23	0.73	20.16	-4.52	-3.35
N_{bumps}	-5.60	26.82	-0.69	12.28	-3.39	0.88
$\cos\theta_{th}$	0.06	-1.02	0.02	0.81	0.02	0.04
p_{tsum}	-7.74	-53.52	-0.63	32.50	-4.90	3.35
E_{BGO}	-0.06	-1.48	-4.80	-5.26	-0.03	0.41
Acopl	0.72	12.97	-28.12	27.26	0.50	9.19
Acol	-3.53	10.99	6.80	35.05	-2.42	-6.40
$\max(E_{BGO} + p_\mu)$	1.56	0.58	-58.10	-63.73	0.84	-48.11
Biases	-3.82	16.84	2.15	-31.08	-3.69	19.80

Output	0.84	-2.57	-9.17	-5.07	0.69	10.04
Output bias	-4.11					

Table 2: Connection weights for $e^+e^- \to \tau^+\tau^-$ neural network.

The possible influence of varying η and α values has been considered as a source of systematic uncertainties.

5.4 Analysis of the network

The variables in the hidden layer can be analized to find the best combinations of initial variables for doing the classification (see chapters 1 and 2). In figures 7, 8 it is shown what kind of events are discriminated in each hidden unit for MC events. The most important variables are those whose combination of connection weights to the hidden layer (table 2) and distribution of input variables are larger. These are, in each case, the following ones:
1) $e^+e^- \to l^+l^-$ events are rejected in hidden units 3 and 6, based on $\max(E_{BGO} + p_\mu)$,

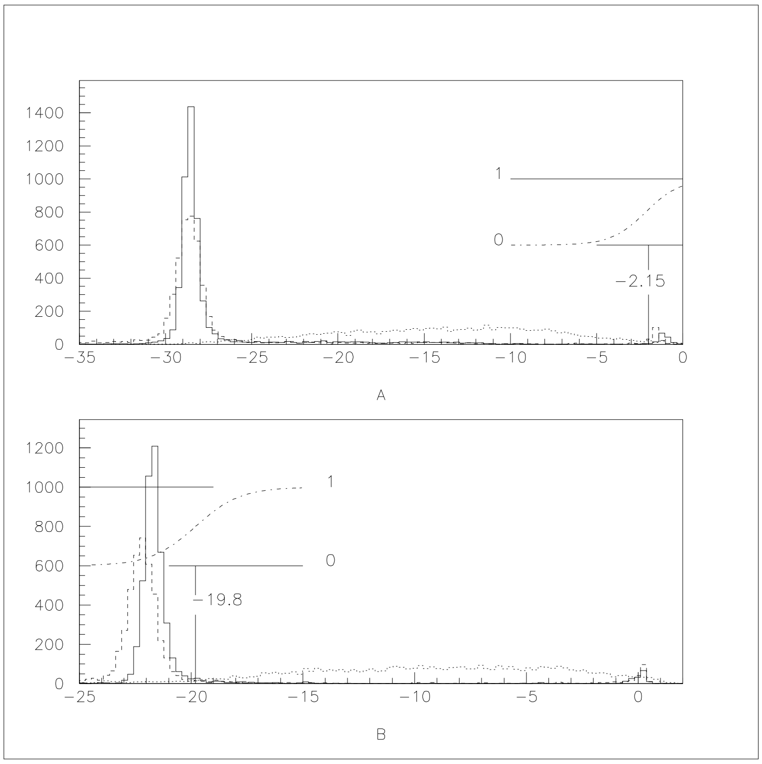

Figure 7: Hidden unit 3 (A) and 6 (B) distributions for $e^+e^- \to \tau^+\tau^-$ (dotted line), $e^+e^- \to e^+e^-$ (continuous line) and $e^+e^- \to \mu^+\mu^-$ (dashed line).

Acopl, N_{had}. These are the input variables whose connection weights with the hidden units 3 and 6 are larger (table 2), but for this type of events $N_{had} = 0$, the same behaviour than the $e^+e^- \to \tau^+\tau^-$ events. Then, the variable N_{had} is not important in the discrimination of $e^+e^- \to l^+l^-$ events.

The hidden unit 3 rejects the leptonic events with one bad reconstructed lepton, the momentum is not well measured, and the most important variable is Acopl. These are a very small number and they can be seen as a peek close to 0 in the figure 7A, and the bias of this unit is 2.15 as can be seen in the table 2. This means that if the value in the hidden unit 3 is smaller than -2.15 then the output of this unit is close to 0. In this case, this means that the event is probably accepted because the connection weight between the hidden unit 3 and the unit in the output layer is negative. If the value in the hidden unit 3 is larger than -2.15 then the output of this unit is close to 1 (the event is probably rejected). The uncertainty region for accepting or rejecting a given event is around -2.15 and it is described by the sigmoid function (chapter 1) of the figure 7A.

The hidden unit 6 rejects the leptonic events that have a lepton with an energy close to $45GeV$, mainly using $\max(E_{BGO} + p_\mu)$. These kind of events are the peek with center in -21.65 in the figure 7B, because if $\max(E_{BGO} + p_\mu)_{initial} \approx 45GeV$, with the normalization, the input given to the network is $\max(E_{BGO} + p_\mu) \approx 0.45$. The product of this input with its connection weight (-48.11, see table 2) is -21.65. This is the central value of the peek. In this case, the bias of the hidden unit is in 19.8. The connection weight between the hidden unit 6 and the output layer unit is positive, so if the output of the hidden unit is

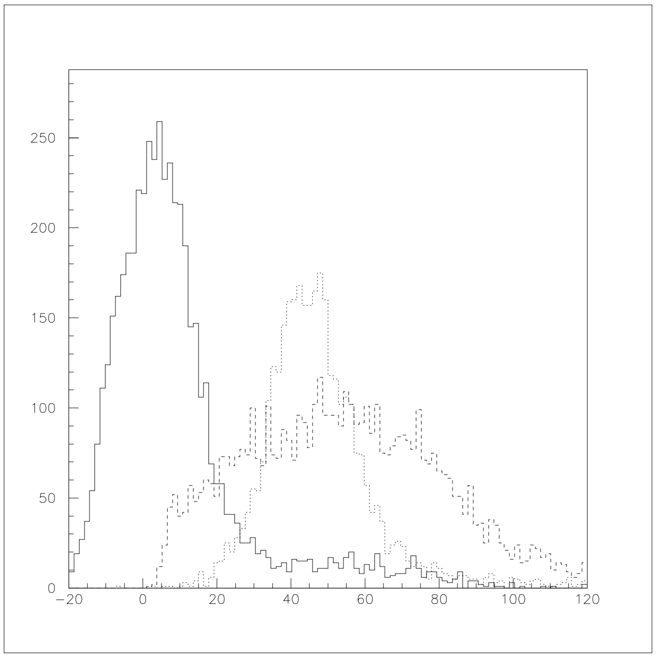

Figure 8: Hidden unit 4 distributions for $e^+e^- \to \tau^+\tau^-$ (continuous line), two photon (dashed line) and $e^+e^- \to q\bar{q}$ (dotted line). The number of events is not normalized.

close to 0, this means that the event is probably rejected and if the output of the hidden unit is close to 1, this means that the event is probably accepted. The uncertainty region is described by the sigmoid function that can be seen in the figure 7B.

2) Two photon events are rejected in hidden units 2 and 4, and the most important variables are p_{tmax}, p_{tsum} and Acopl because the distributions of the input variables for two photon events make less important variables such as max($E_{BGO} + p_\mu$), N_{had}, N_T, N_{bumps} or Acol, although their connection weights are large. In both cases, the values of p_{tmax} and p_{tsum} for two photon events are very small,and the value in the hidden units larger than the $e^+e^- \to \tau^+\tau^-$ events is due to Acopl distribution. This can be seen in figure 8, while the $e^+e^- \to \tau^+\tau^-$ events are between $-20.$ and 20., the two photon events are between 0. and 120.

3) The rejection of hadronic events is done in hidden unit 4 with the variables N_T, N_{had}, Acopl, max($E_{BGO} + p_\mu$), N_{bumps}. The rejection is very similar to that of the two photon events but now the most important variables are different, because the distributions of the input variables are different. The distributions in the hidden units for this kind of events are dominated by the high multiplicity. The clean separation between hadronic events and $e^+e^- \to \tau^+\tau^-$ events can be seen in the figure 8.

		ee	$\mu\mu$	$\gamma\gamma$	$q\bar{q}$	$\tau\tau$
Hid. unit 1	bias-3	0.09	0.11	0.0	0.0	0.017
ac. %	bias	0.0	0.05	0.0	0.0	0.013
weight to out 0.84	bias+3	0.0	0.036	0.0	0.0	0.011
Hid. unit 2	bias-3	89.68	88.57	99.99	99.25	97.64
rej. %	bias	88.67	87.46	99.99	99.25	97.45
weight to out -2.57	bias+3	87.41	86.10	72.42	99.19	97.19
Hid.unit 3	bias-3	3.41	5.00	92.71	20.78	10.76
rej. %	bias	3.14	4.31	85.29	12.12	5.55
weight to out -9.17	bias+3	0.30	0.84	72.42	6.22	3.58
Hid. unit 4	bias-3	7.36	8.01	89.18	94.72	10.86
rej. %	bias	6.86	7.58	86.85	91.51	9.15
weight to out -5.07	bias+3	6.38	7.13	84.61	86.62	9.15
Hid. unit 5	bias-3	0.004	0.08	0.0	0.0	0.01
ac. %	bias	0.0	0.03	0.0	0.0	0.01
weight to out 0.69	bias+3	0.0	0.02	0.0	0.0	0.06
Hid.unit 6	bias-3	92.07	75.49	80.96	79.30	97.15
ac. %	bias	6.33	8.02	73.85	70.31	94.66
weight to out 10.04	bias+3	4.24	5.94	66.34	57.80	87.64

Table 3: Acceptance or rejection power of the $e^+e^- \to \tau^+\tau^-$ neural network for a cut in bias-3 , bias and bias+3 on each hidden unit.

The acceptance in each hidden unit can be seen in the table 3. This table shows that the error in the distributions for $e^+e^- \to l^+l^-$ in the hidden unit 6 is very important, due to the large change in the acceptance. If the width of the distributions of the hidden unit 6 for data is larger than the width for the MC, the error in the efficiencies will be large. The values of the acceptance or rejection are showed in bias-3, bias and bias+3, because in these points the values of the sigmoid function associated are $\approx$0, 0.5 and $\approx$1 respectively. This is the region where the discrimination of events is done.

This classification method can be described approximately as a set of effective cuts in the input variables, computed from the hidden units distributions and the corresponding biases. These effective cuts are:

-For rejection of $e^+e^- \to l^+l^-$ events:

max(E$_{BGO}$ + p_μ)<42 GeV, max(E$_{BGO}$ + p_μ)>0.3 GeV.

-For rejection of two photon events: Acol<0.4.

-For rejection of $e^+e^- \to q\bar{q}$ events: N$_{had}$ <3, N$_{bumps}$ <20, Acol< 0.3.

These cuts are only a reference, because the neural network takes into account the correlations among the input variables, but they are a useful tool for computing systematic errors.

The most important backgrounds are going to be those from $e^+e^- \to l^+l^-$ events, because

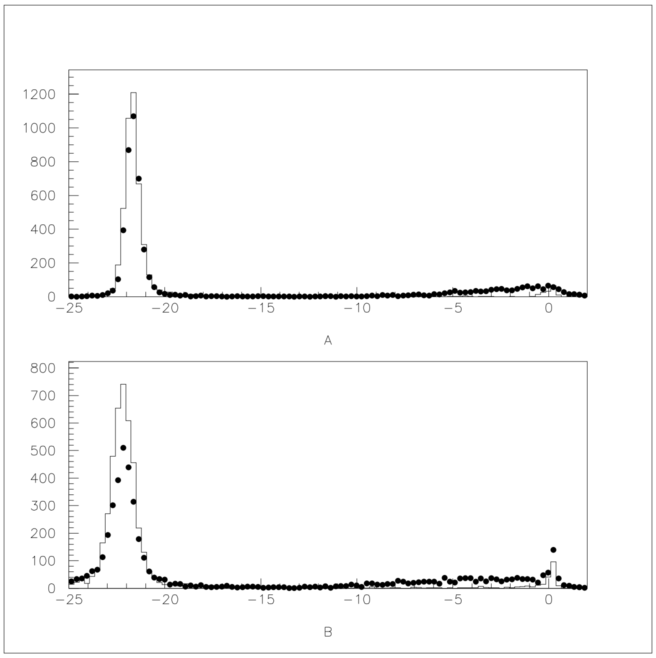

Figure 9: Antiselection of two lepton events: (A) $e^+e^- \to e^+e^-$ and (B) $e^+e^- \to \mu^+\mu^-$. The continuous line is MC (only $e^+e^- \to e^+e^-(\mu^+\mu^-)$ events) and the points are data. The discrepancy in the right side for data is due to two photon events.

these events are not rejected in the preselection. As a self-consistency check, an antiselection of such events is done by using the neural networks described in the next chapter. This is, from the initial data, the $e^+e^- \to l^+l^-$ events are selected requiring a lepton of the same type in each hemisphere.

Using this antiselection, the data and MC distributions have been compared. Of particular relevance is the width of the low end peak in the hidden variable 6 distribution. Any diference between data and MC will corrupt the efficiencies and purities computed from the MC.

The results of these fits are presented in the table 4 and the figures 10 and 11. These fits show a good MC description of the data.

	$e^+e^- \to e^+e^-$		$e^+e^- \to \mu^+\mu^-$	
	DATA	MC	DATA	MC
Mean	-21.66±0.04	-21.70±0.03	-22.17±0.01	-22.19±0.01
Width	0.37±0.06	0.39±0.06	0.64±0.01	0.59±0.09

Table 4: Mean and Width for the hidden variable 6 distribution, data and MC, for antiselected $e^+e^- \to l^+l^-$ events.

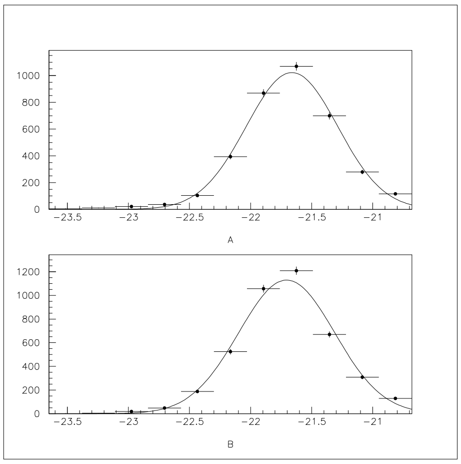

Figure 10: Expanded region of figure 9A around the peak. The curves are the results of the fits discussed in the text for data (A) and MC (B).

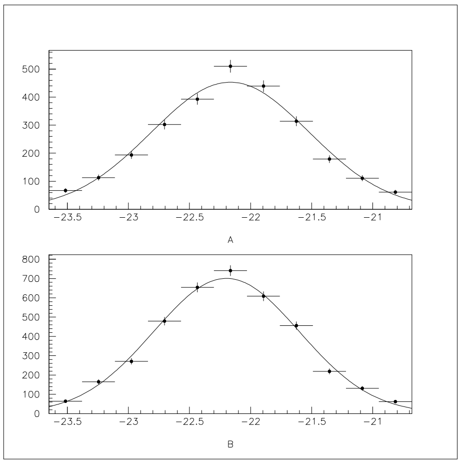

Figure 11: Expanded region of figure 9B around the peak. The curves are the results of the fits discussed in the text for data (A) and MC (B).

In figures 12, 13 can be seen the values of the input variables for MC and data. In the MC all the background has been included.

type of event	σ^{Fid}_{total}	acc.(%)	presel. eff.(%)	sel. eff.(%)	final composition
$e^+e^- \rightarrow \tau^+\tau^-$	1.001	69.18	96.70	79.84	90.67
$e^+e^- \rightarrow e^+e^-$	1.186	79.92	99.90	1.03	1.44
$e^+e^- \rightarrow \mu^+\mu^-$	1.002	72.56	96.00	1.27	1.43
$\gamma\gamma$	0.596	—	58.00	1.17	0.48
$e^+e^- \rightarrow q\bar{q}$	20.59	68.04	6.0	4.12	5.97

Table 5: Efficiencies for all the selection steps for $e^+e^- \rightarrow \tau^+\tau^-$.$\sigma^{Fid}_{total}$ is in nbarns and it is the cross section in the fiducial volume measured at L3 and acc. is the acceptance in the fiducial volume.

5.5 The test

After training, 183048 MC events from a sample independent of the training one have been used for network testing. The composition of the test sample was 29.03% of $e^+e^- \rightarrow \tau^+\tau^-$ events, 35.14% of $e^+e^- \rightarrow e^+e^-$ events, 29.12% of $e^+e^- \rightarrow \mu^+\mu^-$ events, 4.87% of two photon events and 1.84% of $e^+e^- \rightarrow q\bar{q}$ events. This is the composition expected in the data sample. The relation efficiency versus purity reached by this network is plotted in figure 14. An event is taken as $e^+e^- \rightarrow \tau^+\tau^-$ if its network output y is $y > y_{cut}$. The point reached by the selection of ref. [37], using cuts, has been included in the figure 14 as a reference. The efficiency and the purity versus y_{cut} are shown in figure 15.
For this analysis point A is used in $e^+e^- \rightarrow \tau^+\tau^-$ selection, what means an efficiency of 79.84% and a background of 9.33% coming from $e^+e^- \rightarrow e^+e^-$ (1.44%), $e^+e^- \rightarrow \mu^+\mu^-$ (1.43%), two photon events (0.48%) and $e^+e^- \rightarrow q\bar{q}$ (5.97%). This is the same problem as in example 2.4, the purity that can be reached with the neural network method is limitted by the network complexity and the size of the training and test samples. The criterion to select this point was to have a background as small as possible, because the identification of the τ leptonic decays is simpler and more reliable in a sample with small background. Finally, with this analysis, 18429 events were selected as $e^+e^- \rightarrow \tau^+\tau^-$ from the data sample, that corresponds to a cross section of 0.974 ± 0.043 nbarns (compatible with the cross section measured in L3 [27], see table 5).

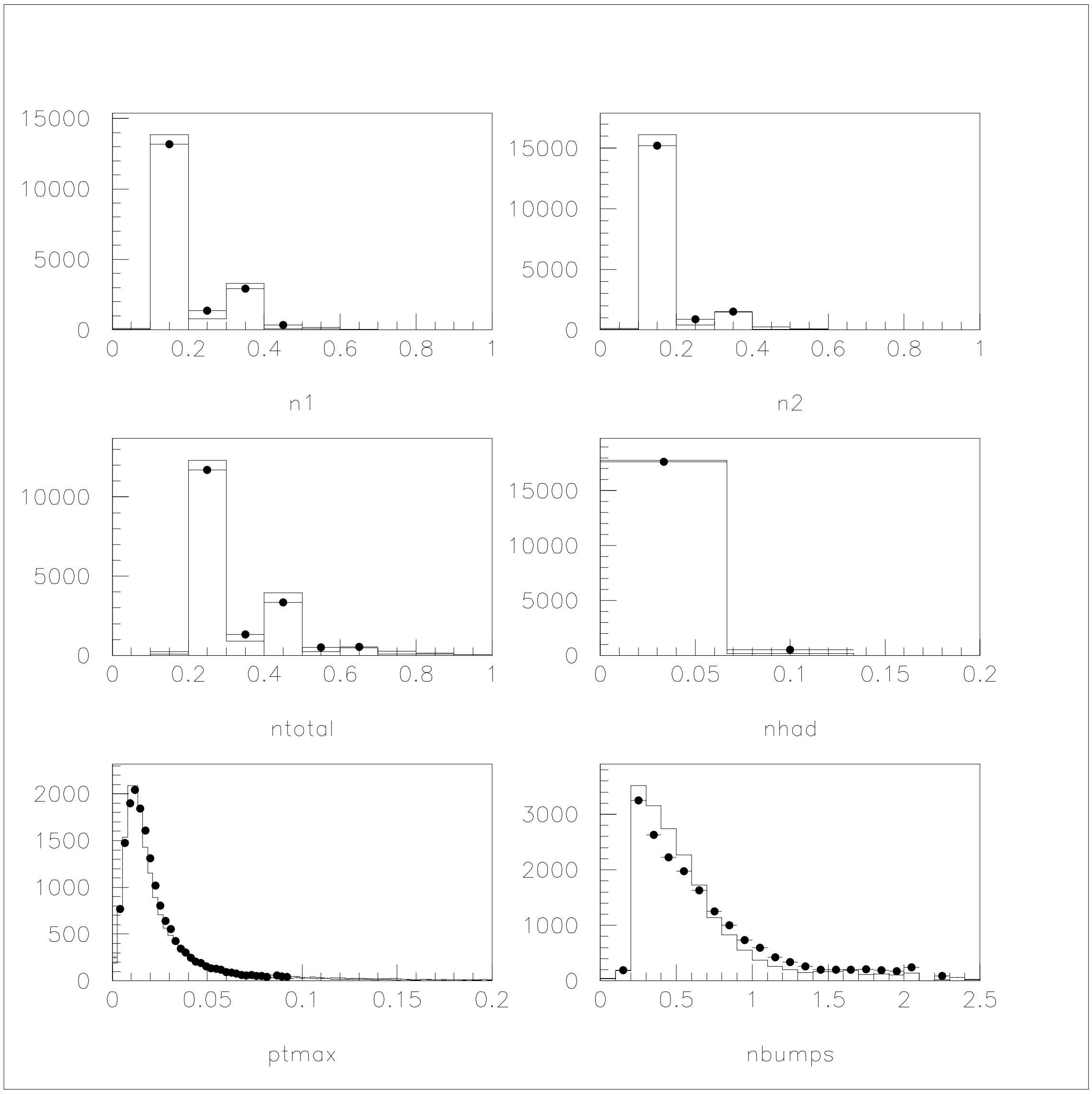

Figure 12: Input variables 1-6 for the events selected as $e^+e^- \to \tau^+\tau^-$. The dots are data and the histograms are MC (all the background is included in the MC).

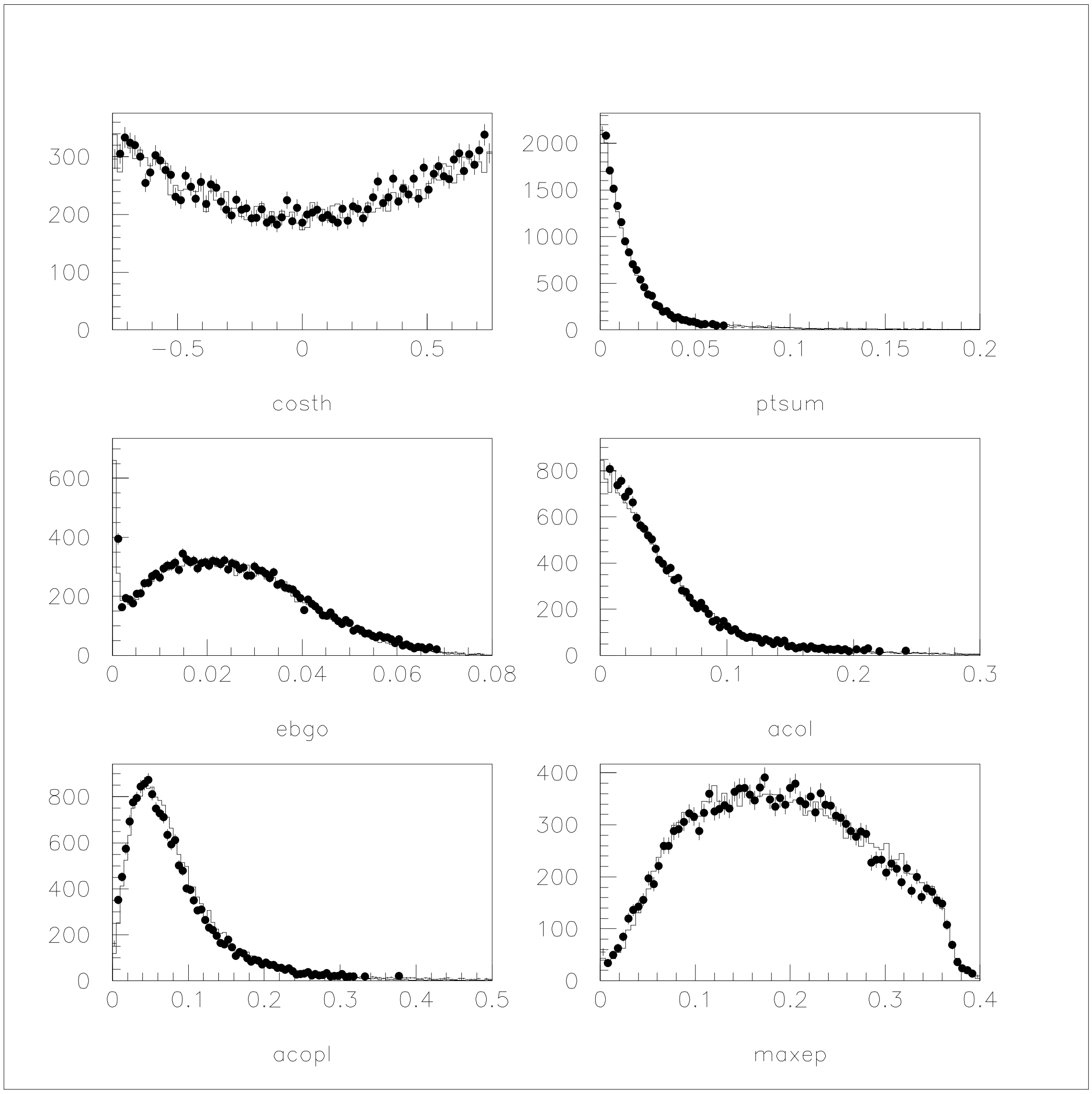

Figure 13: Input variables 7-12 for the events selected as $e^+e^- \to \tau^+\tau^-$. The dots are data and the histograms are MC (all the background is included in the MC).

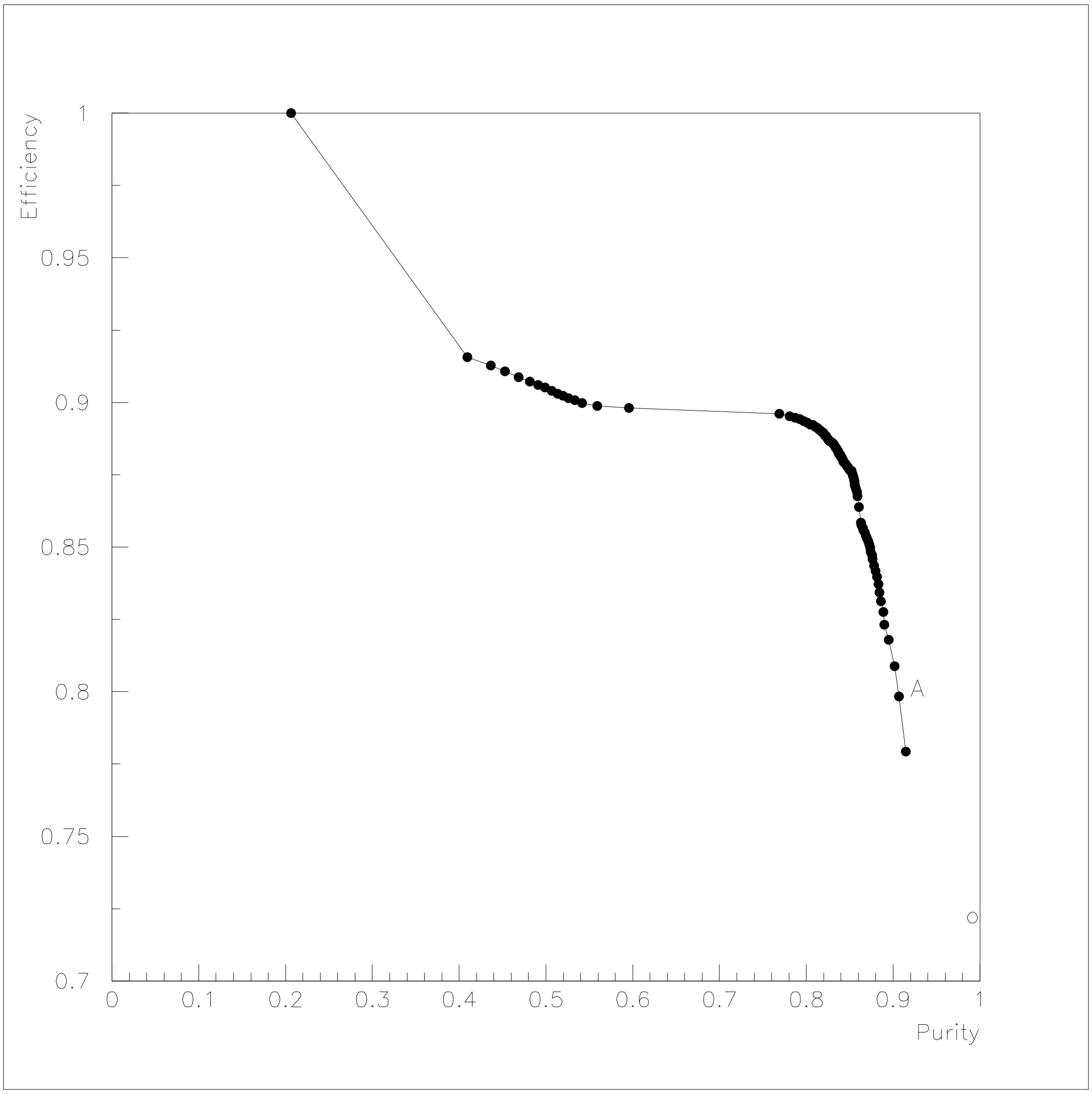

Figure 14: Efficiency.vs.Purity for the reaction $e^+e^- \rightarrow \tau^+\tau^-$ using the neural network (dots), and for selection cuts used in reference [37] (open circle). Point A was selected for the analysis.

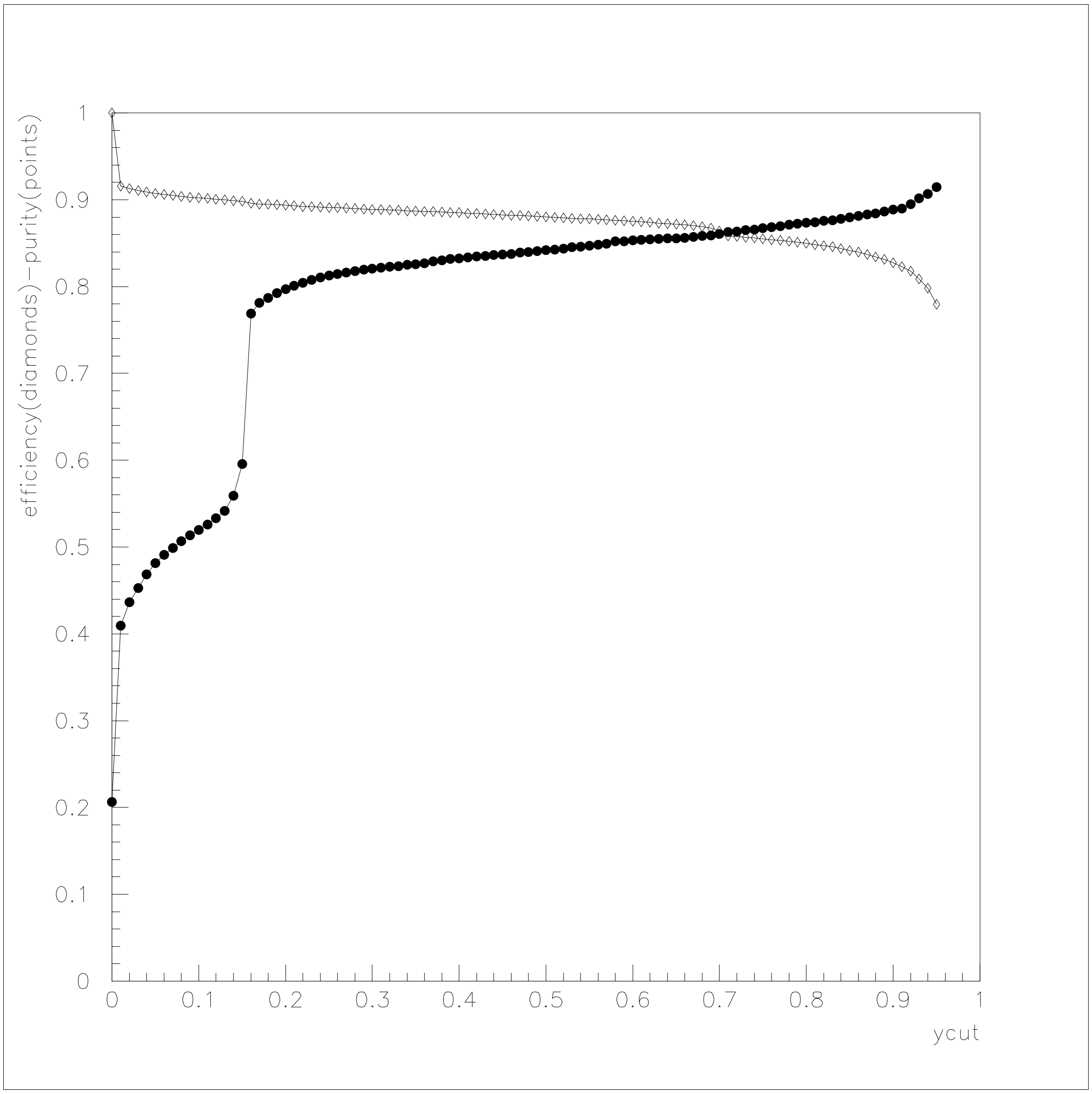

Figure 15: Efficiency (diamonds) and Purity (dots) as functions of the y_{cut} for the $e^+e^- \rightarrow \tau^+\tau^-$ channel using the neural network.

6 $\tau \longrightarrow l\nu_\tau\bar{\nu}_l$ selection

After the $e^+e^- \rightarrow \tau^+\tau^-$ event identification, two neural networks were made to identify the leptonic decays of the τ from the previously selected events.

6.1 Variables used for event selection

The τ leptonic decays should have one track ($e^\pm\mu^\pm$) of momentum p_{TEC} measured in the TEC and produce a well defined bump in the BGO of energy E_{BGO}.
The variables used for this selection are:

1) DPHI = $\Delta\phi$ for the matching between the TEC track and the BGO bump.
2) DTHE = $\Delta\theta$ for the matching between the TEC track and the BGO bump.
3) EPCHI = χ^2 for TEC-BGO matching, defined as

$$\chi^2 = \frac{\frac{1}{2}(E_{BGO}\sin\theta - E_{gss})^2}{(\Delta E_{BGO}\sin\theta)^2} + \frac{\frac{1}{2}p_{TEC}^4(1/p_{TEC} - 1/E_{gss})^2}{(\Delta p_{TEC})^2} \tag{31}$$

, where

$$E_{gss} = \frac{E_{BGO}p_{TEC}^4(\Delta 1/p_{TEC})^2 + p_{TEC}\Delta E_{BGO}^2}{(\Delta p_{TEC})^2 + (\Delta E_{BGO})^2} \tag{32}$$

4) E925 = E_9/E_{25}, where $E_{9(25)}$ is the energy deposited in an array of 9 (25) crystals of the BGO calorimeter around the crystal with largest energy deposition.
5) EMCHI = χ^2 for the BGO cluster to fit an the electromagnetic shower profile.
6) EHAD = Hadronic energy in the hemisphere.

These variables are optimized for the electrons as seen in L3.
The variables have also been used for the muon identification, using the fact that muons are minimum ionization particles (notice that the muon chambers have been not used for this analysis, so this is an independent test for the selection based in the muon chambers). The purpose is to achieve a high efficiency for muon selection without using the muon chambers. Anyway, this analysis can not be considered as definitive. It is only a preliminary result, and a more detailed analysis could be done by combining this one with new variables coming from muon chambers.
For an easier training of the network, these variables have been normalized, according to table 6.

Input variable	Normalization
1	10 DPHI
2	10 DTHE
3	EPCHI/10.
4	E925
5	EMCHI/10.
6	EHAD

Table 6: Normalization of input variables.

6.2 $\tau \rightarrow e\nu_\tau\bar{\nu}_e$ identification

6.2.1 Neural Network

A feed-forward neural network with 6 units in the input layer, 3 units in the hidden layer and 1 unit in the output layer has been used.
The activation functions are sigmoid, both in the hidden layer and in the output layer. Each unit is connected to all the units in the next layer. The target is 1 if the event is $\tau \rightarrow e\nu_\tau\bar{\nu}_e$ and 0 in any other case.
As in the previous chapter, the number of units in the hidden layer has been taken using the rule of thumb of the input/output average.

6.2.2 The training

A sample of 79977 MC events has been used. The composition of this sample was 15.00% of $\tau \rightarrow e\nu_\tau\bar{\nu}_e$ events, 65.00% of other τ decays, 6.25% of $e^+e^- \rightarrow e^+e^-$, 6.25% of $e^+e^- \rightarrow \mu^+\mu^-$, 3.75% of two photon events and 3.75% of $e^+e^- \rightarrow q\bar{q}$. The events were showed randomly to the network, in order to avoid generalization problems. During the training $\alpha = 0.9$ and η decreases from 0.05 to 0.01. The network needed 147683 backpropagation iterations to converge.

6.2.3 Analysis of the network behaviour

The analysis of the neural network has been done using the test sample, described in the next chapter. In this network there is no selection of each type of event in a hidden unit. The net only identifies electrons and rejects any other type of events, without separating among backgrounds. Due to this fact, the hidden units have similar distributions.
The most important variables are EHAD and E925, and the largest backgrounds come from other τ decays and from $e^+e^- \rightarrow e^+e^-$. The network connection weights can be seen in table 7 and the hidden variable 3 in figure 16. EHAD =0 for $\tau \rightarrow e\nu_\tau\bar{\nu}_e$, but it has other values for the other events. The most important variable for the distribution of $\tau \rightarrow e\nu_\tau\bar{\nu}_e$ in the hidden units is E925, close to 1 for the signal and with other values for the background.

input	unit 1	unit 2	unit 3
DPHI	-11.22	-13.00	-5.11
DTHE	-2.94	0.68	0.61
EPCHI	-6.73	-4.58	-3.02
E925	8.54	13.50	3.78
EMCHI	-0.22	-0.79	-4.37
EHAD	-51.66	-51.53	-39.02
Biases	-4.10	-0.18	-0.50

Output	3.28	2.84	2.06
output bias	-2.79		

Table 7: Connection weights for $\tau \rightarrow e\nu_\tau\bar{\nu}_e$ neural network.

This analysis can be interpreted as the following set of effective cuts:
- E925 $>$ 0.87
- EMCHI $<$ 0.9
- EHAD $<$ 0.04

As in the previous chapter, these cuts are only a useful reference and they have been computed from the hidden units distributions, taking into account the biases and the sigmoid functions.

6.2.4 The test

88064 MC events were used to test the network. The composition of the test sample was 17.39% of $\tau \rightarrow e\nu_\tau\bar{\nu}_e$ events, 78.94% of other τ decays, 1.25% of $e^+e^- \rightarrow e^+e^-$, 1.85% of $e^+e^- \rightarrow \mu^+\mu^-$, 0.24% of two photon events and 0.31% of $e^+e^- \rightarrow q\bar{q}$ events. The efficiency versus purity function is plotted in figure 17. Some points obtained using previous analyses [37], [27] are also included. In figure 18 the efficiency and the purity versus y_{cut} can be seen.

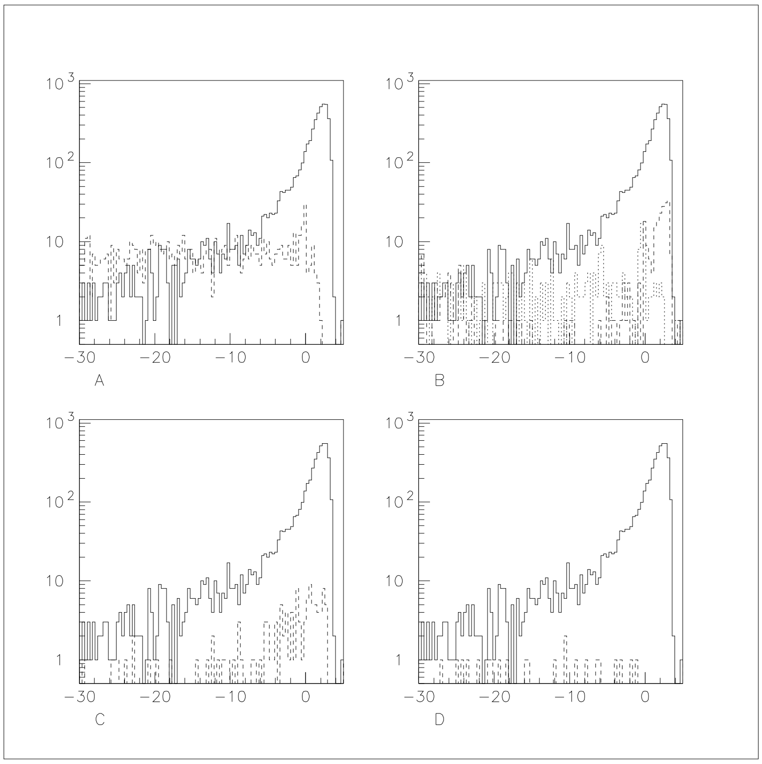

Figure 16: Hidden unit 3 distributions for $\tau \to e\nu_\tau\bar{\nu}_e$ neural network (continuous line). A) $\tau \to$ no e (dashed line), B) $e^+e^- \to e^+e^-$ (dashed line) and $e^+e^- \to \mu^+\mu^-$ (dotted line), C) two photon (dashed line), D) $e^+e^- \to q\bar{q}$ (dashed line).

The point selected for this analysis provides an efficiency of 90.12% and a background of 9.56%, coming from other τ decays (4.73%), $e^+e^- \to e^+e^-$ (2.61%) and two photon events (2.06%). This point has been selected looking for a small systematic error. The MC is a good description of the data near to this point ($y_{cut} = 0.85$). If the cut point is selected looking for a smaller background (larger values for y_{cut}) then the MC is not so good description of the data, and the systematic error grows. This is due to the fact that if the value of y_{cut} is changed, then the effective cuts are different and correspond to zones were the MC is not a good description of the input variables for the data.

type of event	sel. eff. (%)	final composition
$\tau \to e\nu_\tau\bar{\nu}_e$	90.12	90.44
Other τ decays	1.03	4.73
$e^+e^- \to e^+e^-$	29.71	2.61
$e^+e^- \to \mu^+\mu^-$	1.77	0.15
$\gamma\gamma$	47.14	2.06
$e^+e^- \to q\bar{q}$	0.00	0.00

Table 8: Efficiencies for the selection of $\tau \to e\nu_\tau\bar{\nu}_e$ events.

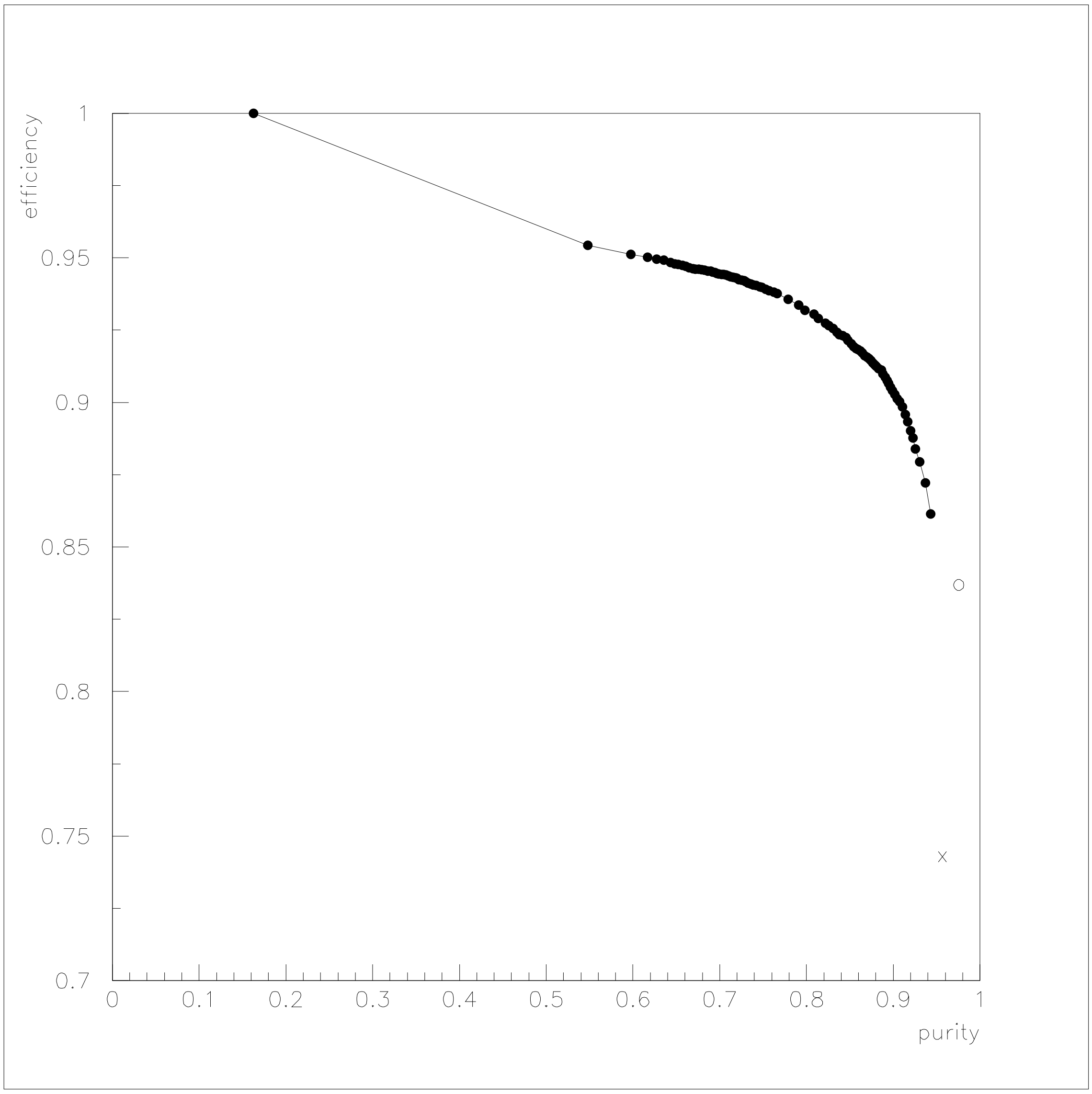

Figure 17: Efficiency.vs.Purity for $\tau \to e\nu_\tau\bar{\nu}_e$ using the neural network (dots), the analysis of ref. [37] (open circle) and the analysis of ref. [27] (cross).

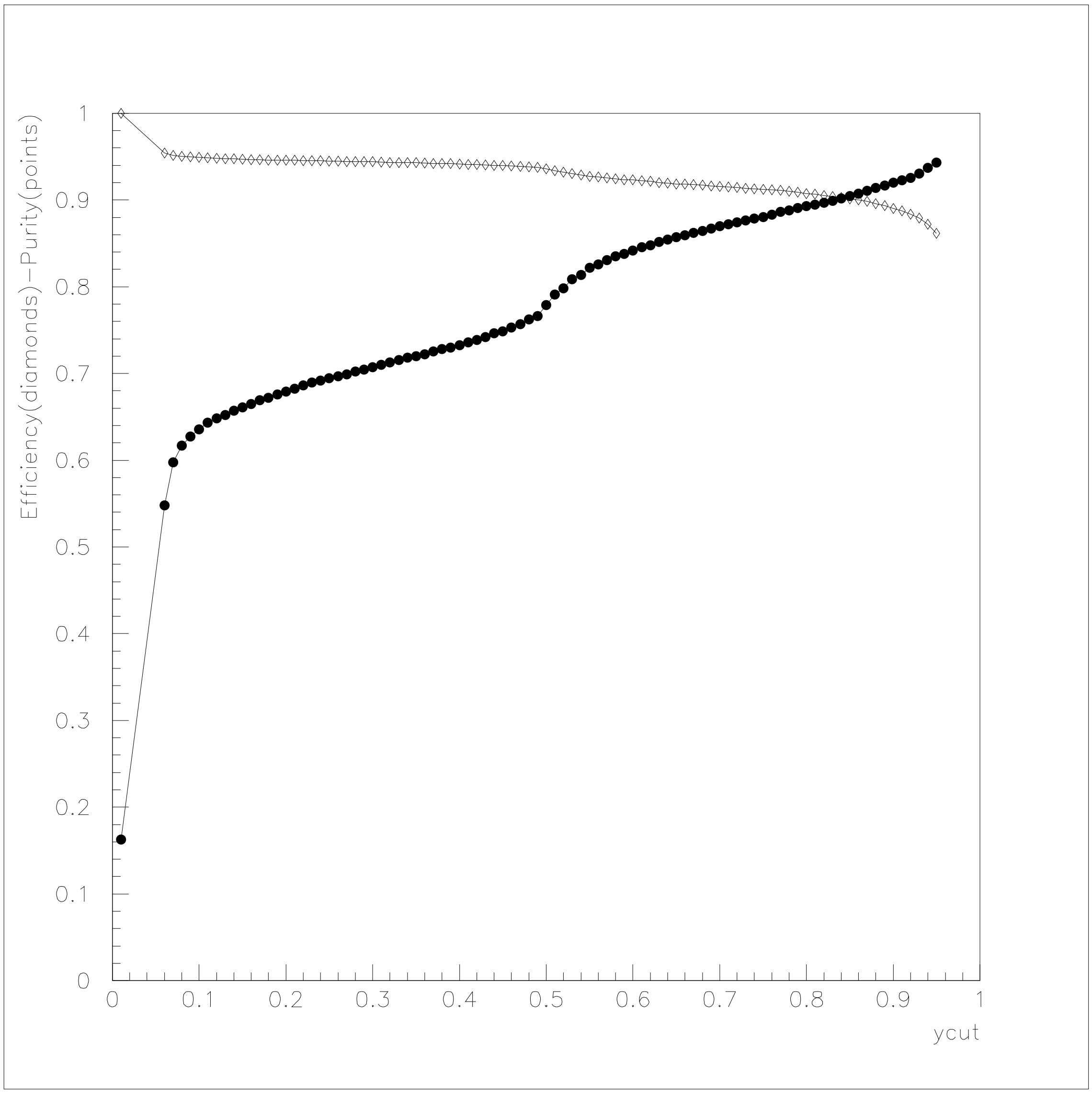

Figure 18: Efficiency (diamonds) and Purity (dots).vs. y_{cut} for $\tau \rightarrow e\nu_\tau\bar{\nu}_e$ using the neural network.

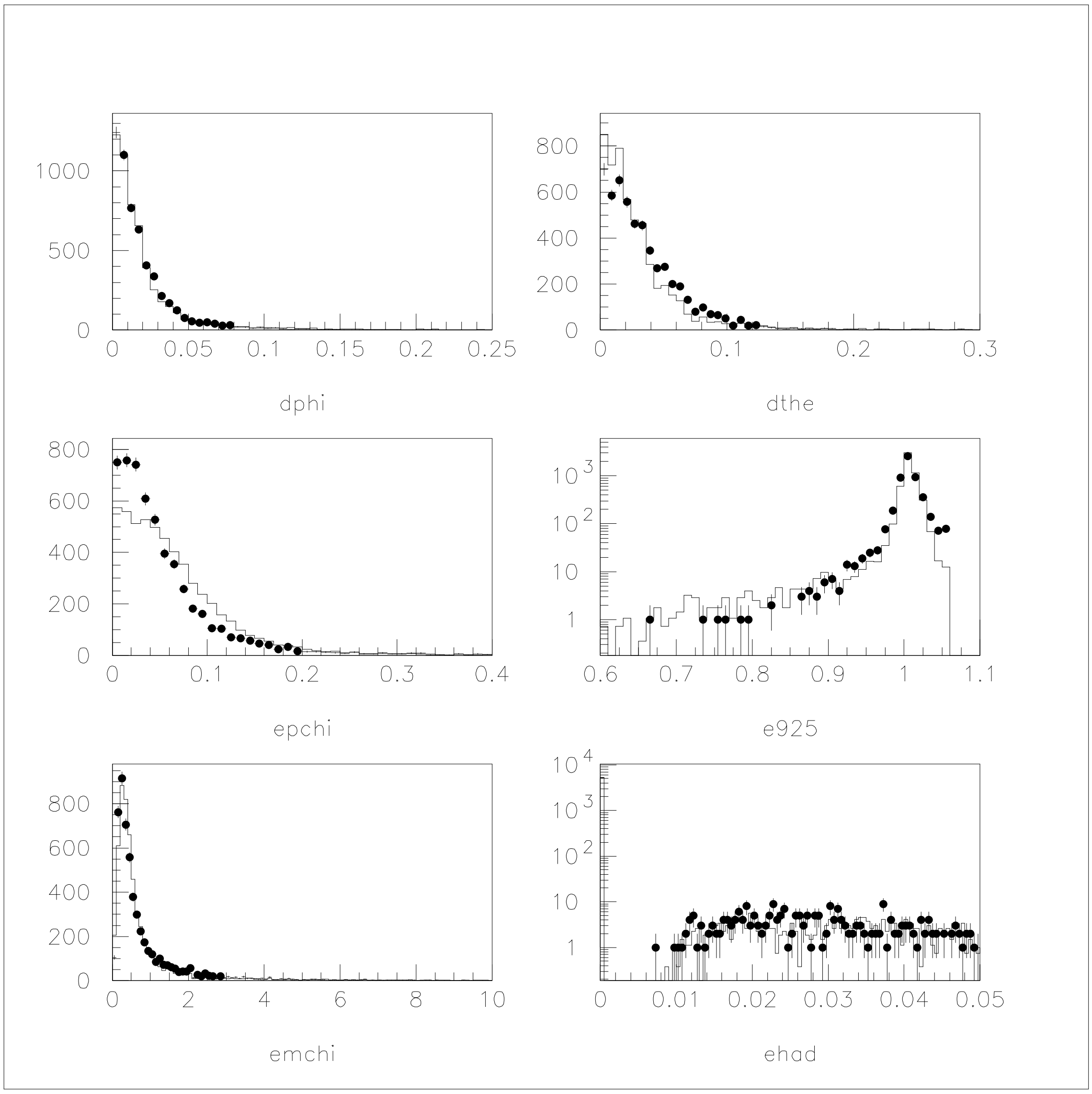

Figure 19: Input variables for events selected as $\tau \to e\nu_\tau\bar{\nu}_e$ both data (points) and MC (histograms). All backgrounds are included in the MC events.

type of event	sel. eff. (%)	final composition
$\tau \to e\nu_\tau\bar{\nu}_e$	81.10	87.93
Other τ decays	1.01	5.16
$e^+e^- \to e^+e^-$	0.27	0.00
$e^+e^- \to \mu^+\mu^-$	51.71	5.03
$\gamma\gamma$	29.05	1.40
$e^+e^- \to q\bar{q}$	1.07	0.44

Table 9: Efficiencies for the selection of $\tau \to \mu\nu_\tau\bar{\nu}_\mu$ events.

6.3 $\tau \to \mu\nu_\tau\bar{\nu}_\mu$ identification

6.3.1 Neural network

The neural network used for this analysis is also feed-forward, with 6 units in the input layer, 10 units in the hidden layer and 1 unit in the output layer.
The activation functions and the targets have been taken as in the previous cases.
The number of units in the hidden layer is 10 because the variables used are not the best ones. So, the network is allowed to use more combinatons of variables for the identification.

6.3.2 The training

In this case, 84421 MC events were showed randomly to the network. The composition of the training sample was 14.21% of $\tau \to \mu\nu_\tau\bar{\nu}_\mu$ events, 66.85% of other τ decays, 5.92% of $e^+e^- \to e^+e^-$, 5.92% of $e^+e^- \to \mu^+\mu^-$, 3.55% of two photon events and 3.55% of $e^+e^- \to q\bar{q}$ events. η decreases from 0.01 to 0.005 and $\alpha = 0.7$. The number of backpropagation iterations was 191955.

6.3.3 The test

The same 88064 events of the preceding chapter have been used for testing this network. Obviously, these events were independent of the events in the training sample. Now the composition was 17.12% of $\tau \to \mu\nu_\tau\bar{\nu}_\mu$ and 79.22% of other τ decays, the other backgrounds have the same proportion than in the $\tau \to e\nu_\tau\bar{\nu}_e$ case. The figures 20 and 21 show the efficiency and purity obtained in this analysis. The results obtained with cuts are also showed, but here this is only a reference and no comparision should be made because the variables used in the two analyses are completely different.
The point selected for the identification provides an efficiency of 81.10% and a background of 11.81%, coming from other τ decays (5.16%), $e^+e^- \to \mu^+\mu^-$ (5.03%), and two photon events (1.40%).

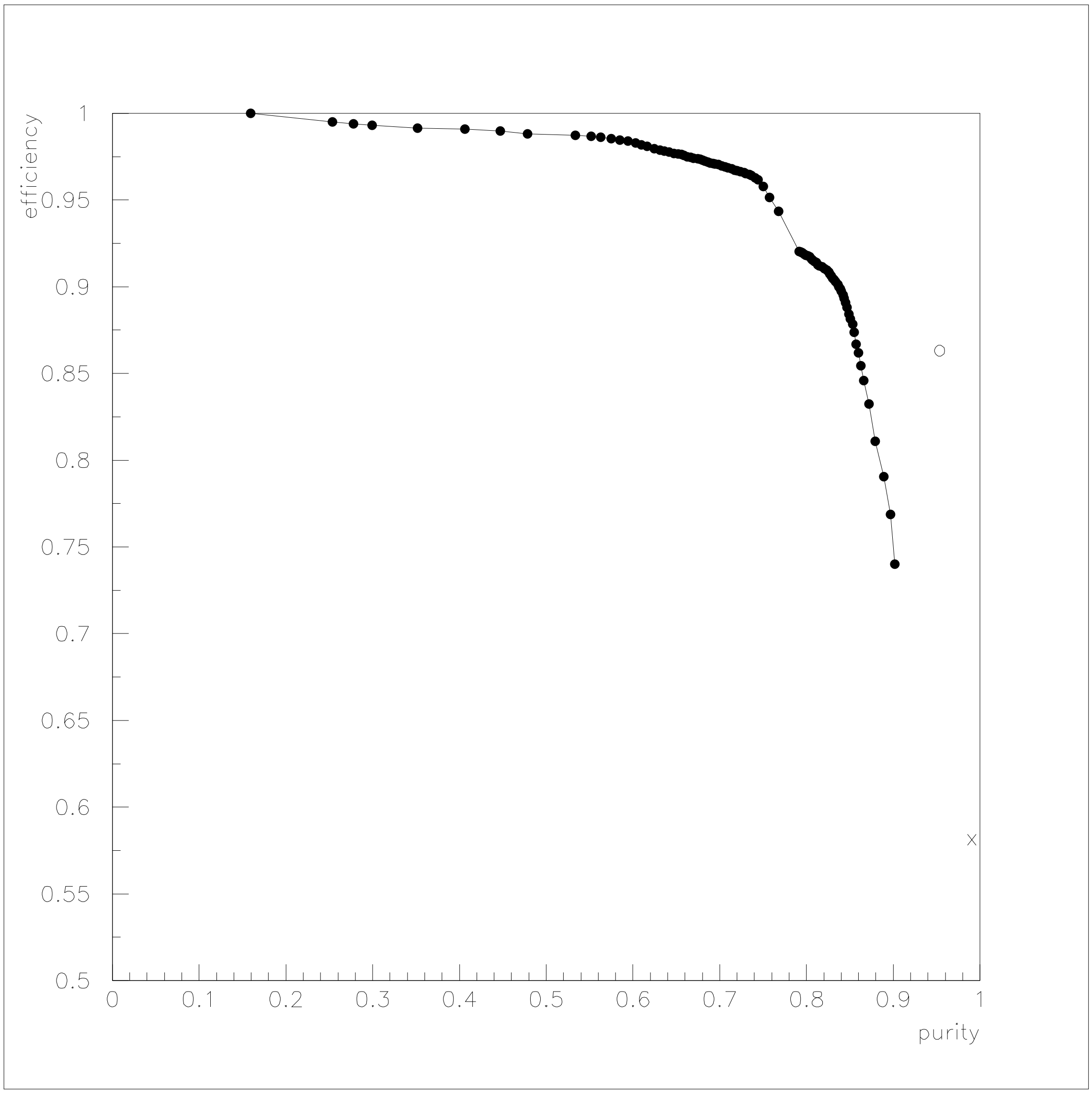

Figure 20: Efficiency.vs.Purity for $\tau \rightarrow \mu\nu_\tau\bar{\nu_\mu}$ using the neural network (dots), for analysis of ref. [37] (open circle) and for analysis of ref. [27] (cross).

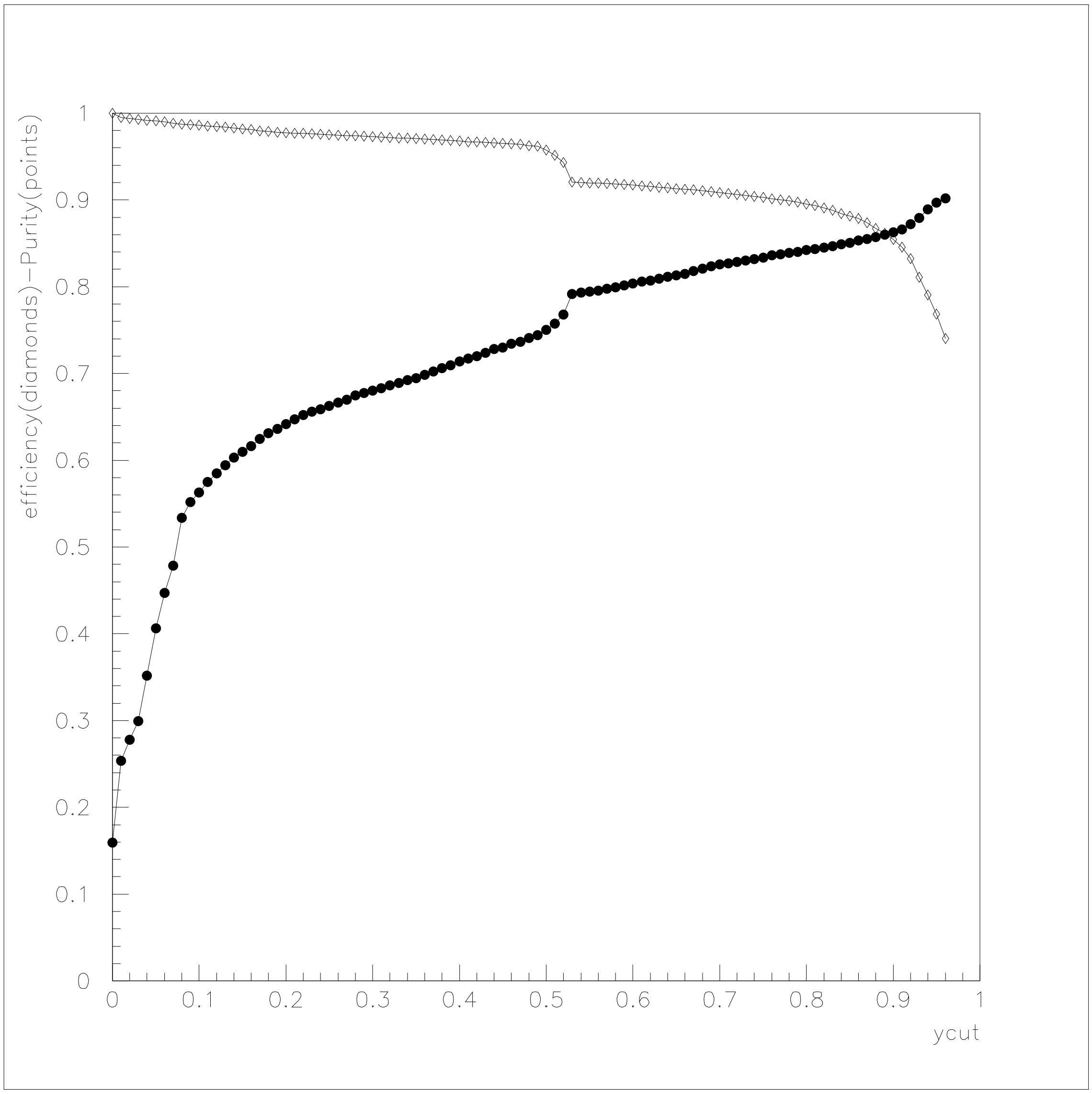

Figure 21: Efficiency (diamonds) and Purity (dots).vs. y_{cut} for $\tau \rightarrow \mu\nu_\tau\bar{\nu_\mu}$ using the neural network.

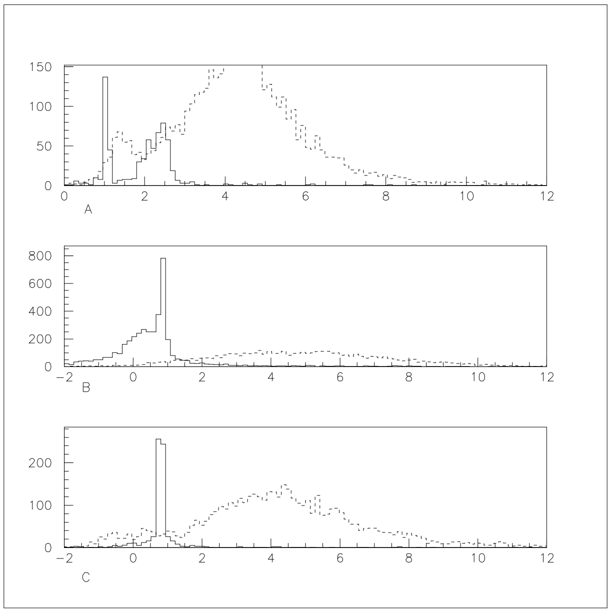

Figure 22: Hidden units for $\tau \rightarrow \mu\nu_\tau\bar{\nu_\mu}$ (dashed lines). A) Hidden unit 1,continuous line is for $e^+e^- \rightarrow e^+e^-$, B) Hidden unit 2,continuous line is for other τ decays, C) Hidden unit 3,continuous line is for $e^+e^- \rightarrow e^+e^-$.

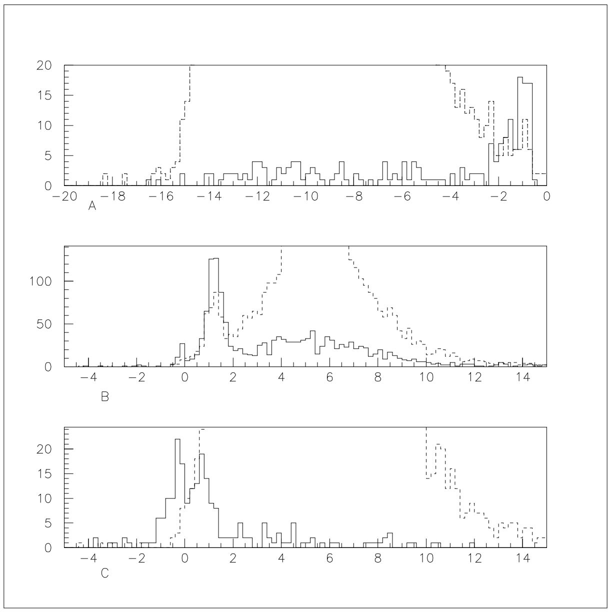

Figure 23: Hidden units for $\tau \rightarrow \mu\nu_\tau\bar{\nu_\mu}$ (dashed lines). A) Hidden unit 4,continuous line is for two photon events, B) Hidden unit 5,continuous line is for $e^+e^- \rightarrow \mu^+\mu^-$, C) Hidden unit 5,continuous line is for $e^+e^- \rightarrow q\bar{q}$.

input	1	2	3	4	5	6	7	8	9	10
DPHI	-0.64	-0.28	-0.39	0.56	-0.68	0.29	-0.26	-0.73	1.17	0.01
DTHE	-1.05	-0.45	-0.75	0.94	-1.08	0.59	-0.56	-1.22	1.44	-0.05
EPCHI	-0.12	1.44	-0.28	-0.83	0.18	-2.72	1.55	-0.53	-0.26	0.34
E925	1.03	0.80	0.88	-0.45	0.99	-1.24	0.96	1.69	-3.27	-0.44
EMCHI	0.14	-0.02	-0.01	-0.92	-0.01	0.19	-0.14	0.55	0.22	-0.42
EHAD	0.49	-0.61	0.80	0.21	0.65	0.38	-0.51	-0.21	-0.04	-0.25
Biases	-0.67	-0.45	-0.62	1.62	-0.82	1.75	-0.61	-0.40	0.98	0.015
Output	1.21	0.99	1.05	-2.62	1.38	-3.42	1.41	1.60	-3.41	-0.66
output bias	-2.94									

Table 10: Connection weights for $\tau \to \mu\nu_\tau\bar{\nu_\mu}$ neural network.

6.3.4 Analysis of the network behaviour

In figures 22 and 23 can be seen some hidden variables for this network. The most important variables are:
To reject other τ decays, EPCHI and E925.
For $e^+e^- \to e^+e^-$, DPHI, DTHE, E925, EHAD.
For $e^+e^- \to \mu^+\mu^-$, DPHI, DTHE, E925.
For $e^+e^- \to q\bar{q}$, DPHI, DTHE, E925.
For two photons events, DTHE, EPCHI, EMCHI.

6.4 Results of the selection

In these selections 6106 $\tau \to e\nu_\tau\bar{\nu_e}$ events and 5335 $\tau \to \mu\nu_\tau\bar{\nu_\mu}$ events are found.
In figures 19 and 24 the input variables for events selected as $\tau \to l\nu_\tau\bar{\nu_l}$ both for MC and data can be seen. The figure 25 shows the energy distribution of the selected events.

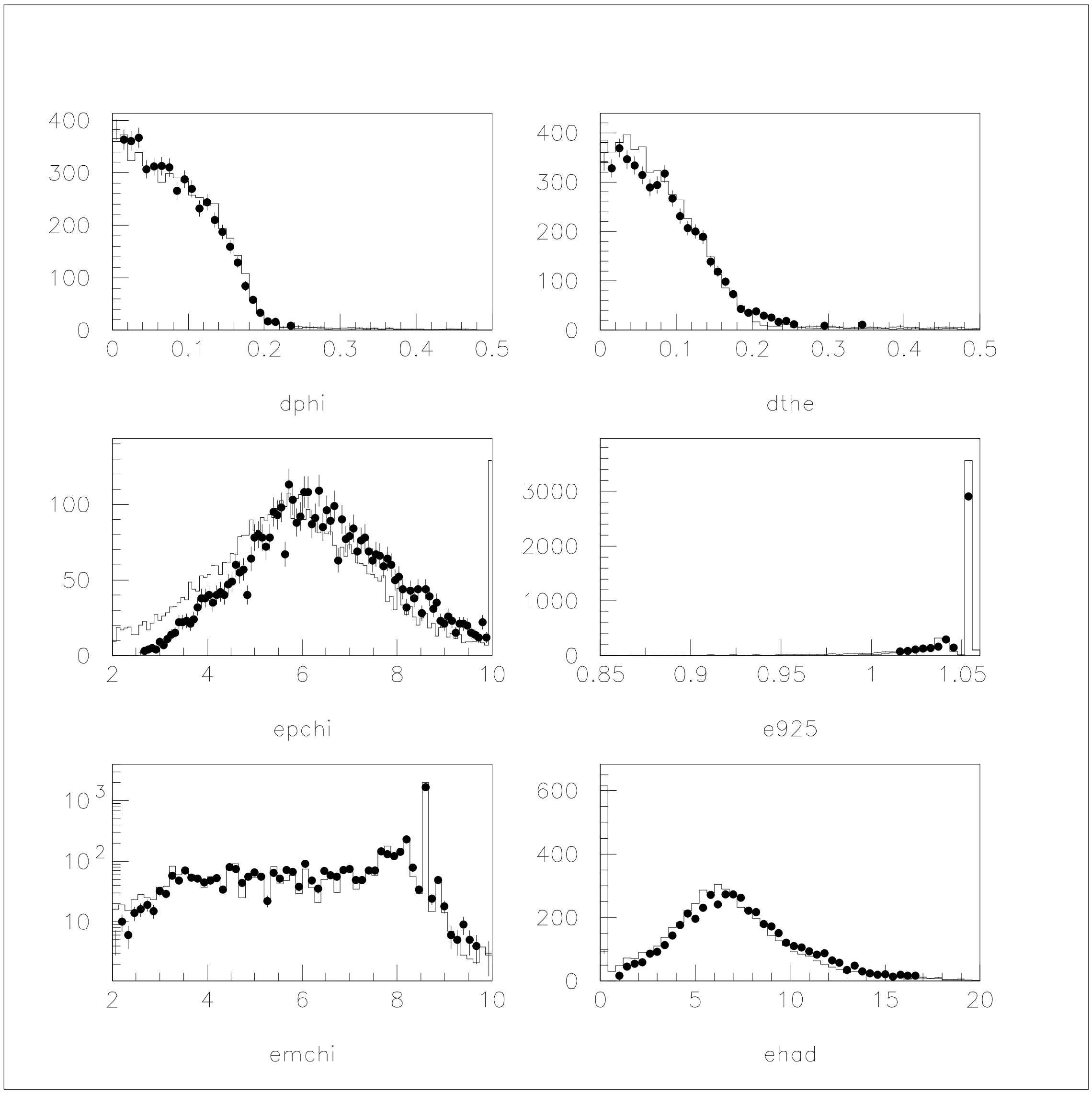

Figure 24: Input variables for events selected as $\tau \to \mu\nu_\tau\bar{\nu}_\mu$ both data (points) and MC (histograms). All backgrounds are included in the MC events.

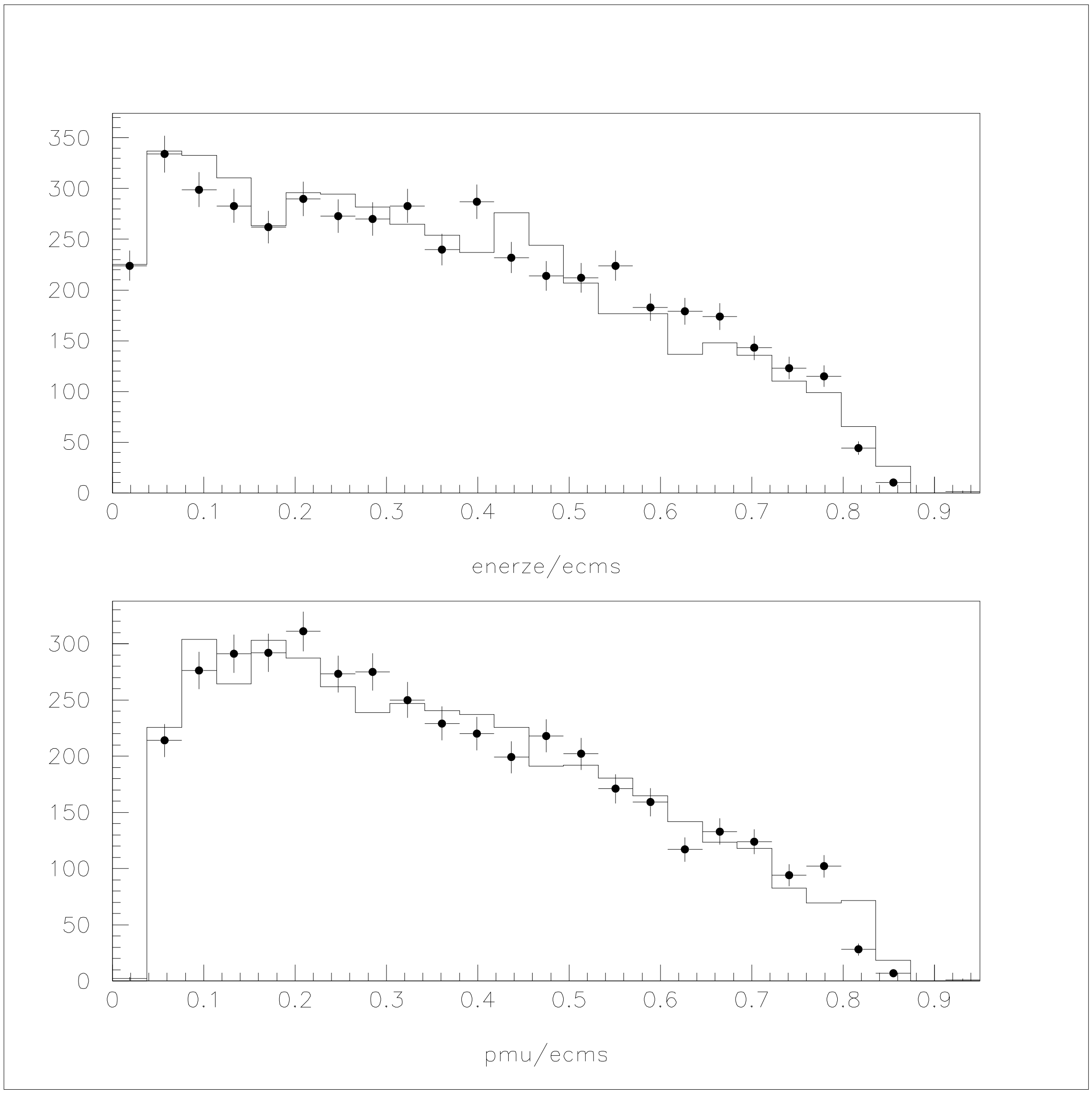

Figure 25: Electron energy normalized to E_{beam} and muon momentum normalized to E_{beam} for events selected as $\tau \rightarrow l\nu_\tau\bar{\nu}_l$. Points are data and histograms are MC. All the background is included in the MC events.

7 Results

τ leptonic branching ratios are determined from the ratio of number of $\tau \to e\nu_\tau\bar{\nu}_e$ events selected to the number of $\tau \to \mu\nu_\tau\bar{\nu}_\mu$ events selected. After efficiency and background substraction corrections, the results are:

$$BR(\tau \to e\nu_\tau\bar{\nu}_e) = 0.1824 \pm 0.0026 \pm 0.0025 \tag{33}$$

$$BR(\tau \to \mu\nu_\tau\bar{\nu}_\mu) = 0.1752 \pm 0.0023 \pm 0.0031 \tag{34}$$

where the first error is statistical and the second one is systematic.
From these values the following conclusions can be obtained (see chapter 4):

$$\frac{BR(\tau \to \mu\nu_\tau\bar{\nu}_\mu)}{BR(\tau \to e\nu_\tau\bar{\nu}_e)} = 0.961 \pm 0.033 \tag{35}$$

and

$$\frac{G_\tau}{G_\mu} = 0.997 \pm 0.027 \tag{36}$$

Both values are in good agreement with leptonic universality. The L3 τ lifetime [27] $\tau_\tau = 293 \pm 9 \pm 12$ fs and the value of τ mass $m_\tau = 1784.1^{+2.7}_{-3.6}$ MeV [28] have been used.
For the QCD related quantities the results are,

$$R_\tau = 3.52 \pm 0.12 \tag{37}$$

This value implies

$$\alpha_s(Q^2 = m_\tau^2) = 0.288^{+0.047}_{-0.041} \pm 0.03 \tag{38}$$

where the first error comes from the experimental uncertainties and the second one comes from the theoretical uncertainties.
Extrapolating the α_s value from $Q^2 = m_\tau^2$ to m_Z^2 [39] the following value is obtained,

$$\alpha_s(Q^2 = m_Z^2) = 0.115 \pm 0.005 \pm 0.004 \tag{39}$$

and

$$\Lambda^{(5)}_{\overline{MS}} = 152^{+62}_{-53} \pm 39MeV \tag{40}$$

These values are in agreement with the values found from the topology of hadronic Z^0 decays at L3 [27].

y_{cut}	0.70	0.75	0.80	0.85	0.90
0.93	0.1816	0.1820	0.1824	0.1824	0.1823
0.94	0.1818	0.1815	0.1823	0.1824	0.1825
0.95	0.1814	0.1816	0.1821	0.1823	0.1821

Table 11: Variation of the branching ratio of $\tau \to e\nu_\tau\bar{\nu_e}$ for the variation of y_{cut}, $\tau \to e\nu_\tau\bar{\nu_e}$ (horizontal), and $e^+e^- \to \tau^+\tau^-$ (vertical).

y_{cut}	0.92	0.93	0.94	0.95	0.96
0.93	0.1755	0.1756	0.1764	0.1761	0.1766
0.94	0.1753	0.1752	0.1761	0.1758	0.1764
0.95	0.1756	0.1760	0.1761	0.1762	0.1759

Table 12: Variation of the branching ratio of $\tau \to \mu\nu_\tau\bar{\nu_\mu}$ for the variation of y_{cut}, $\tau \to \mu\nu_\tau\bar{\nu_\mu}$ (horizontal), and $e^+e^- \to \tau^+\tau^-$ (vertical).

8 Systematic errors

The main problem in neural network analysis is the evaluation of the systematic errors. In this work, the sources of systematic errors that have been found are the following ones: y_{cut} selection, finite MC statistics, number of hidden units, training parameters, disagreement between MC and data (efficiencies and backgrounds). The computation of the systematic errors associated to these sources has been done as follows.

-y_{cut} selection from output of neural network: The y_{cut} value in each neural network has been varied in the range showed in tables 11, 12. This error can be associated to that coming from choosing cuts in the standard analysis. Note that another y_{cut} value is not appropiated because the systematic uncertainty associated would be very large. This is because if y_{cut} is chosen larger than the values showed, the MC can not reproduce the data as can be seen in the figure 27. In this figure, the ratio between the integrated histograms for MC and data of the figure 26 for $\tau \to e\nu_\tau\bar{\nu_e}$. If y_{cut} is larger than 0.9, the MC and the data are not in good agreement and the ratio is different to 1. For example, if the value of y_{cut} for the neural network identifying $\tau \to e\nu_\tau\bar{\nu_e}$ is chosen in 0.95 the effective cut in E925 is E925 > 0.94 and the discrepances between data and MC are more important than for the value of y_{cut} given in table 11. When y_{cut} grows, this effective cut is nearer to 1. and the efficiency value computed from MC is not so accurate. The other values of the effective cuts are not so important. However, if y_{cut} is chosen in the region of tables 11, 12, the MC and the data are in good agreement and the systematic error is small.

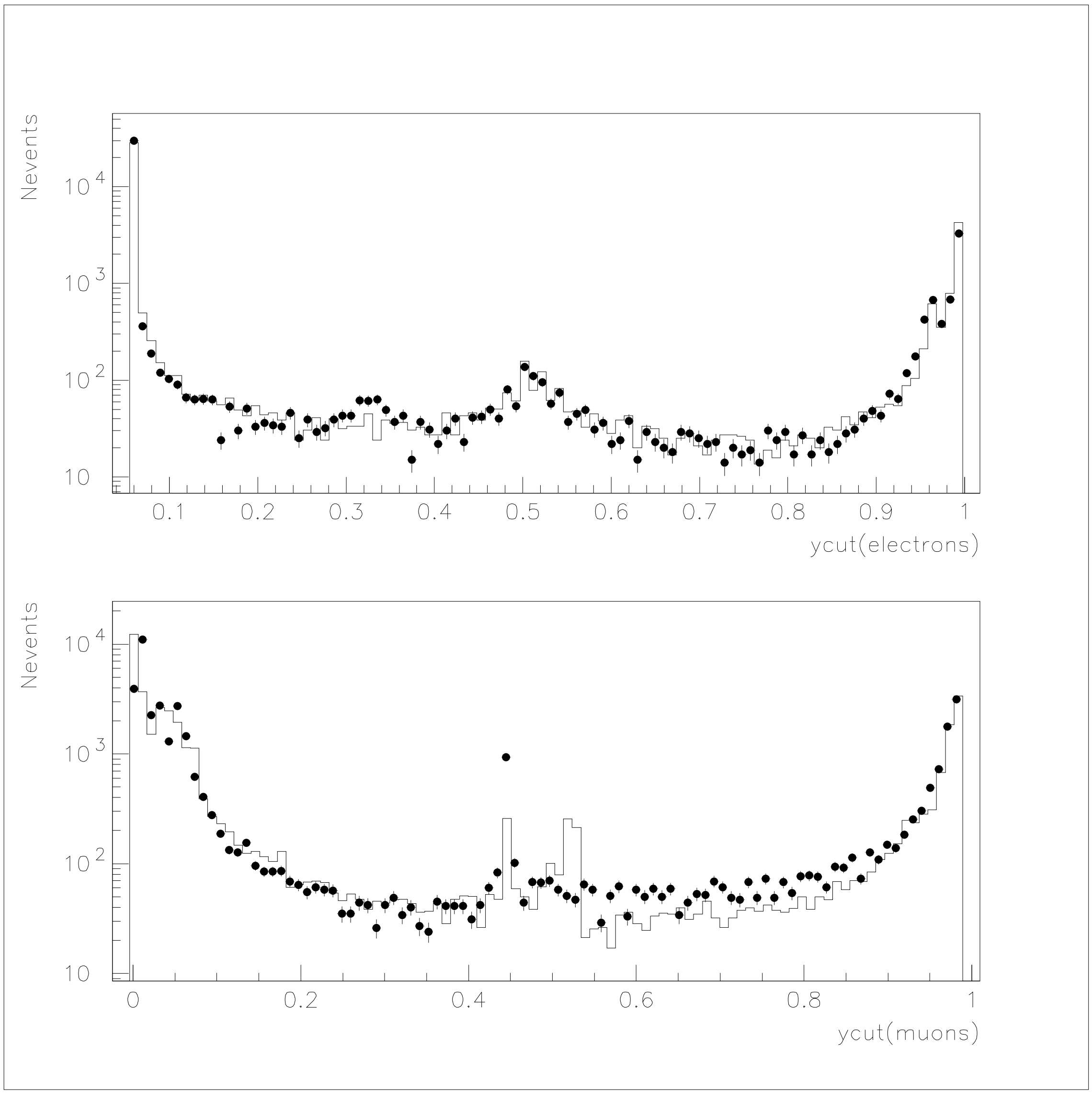

Figure 26: Neural network output for $\tau \to e\nu_\tau\bar{\nu}_e$ (top) and for $\tau \to \mu\nu_\tau\bar{\nu}_\mu$. Points are data and histograms are MC. All the background is included in the MC events.

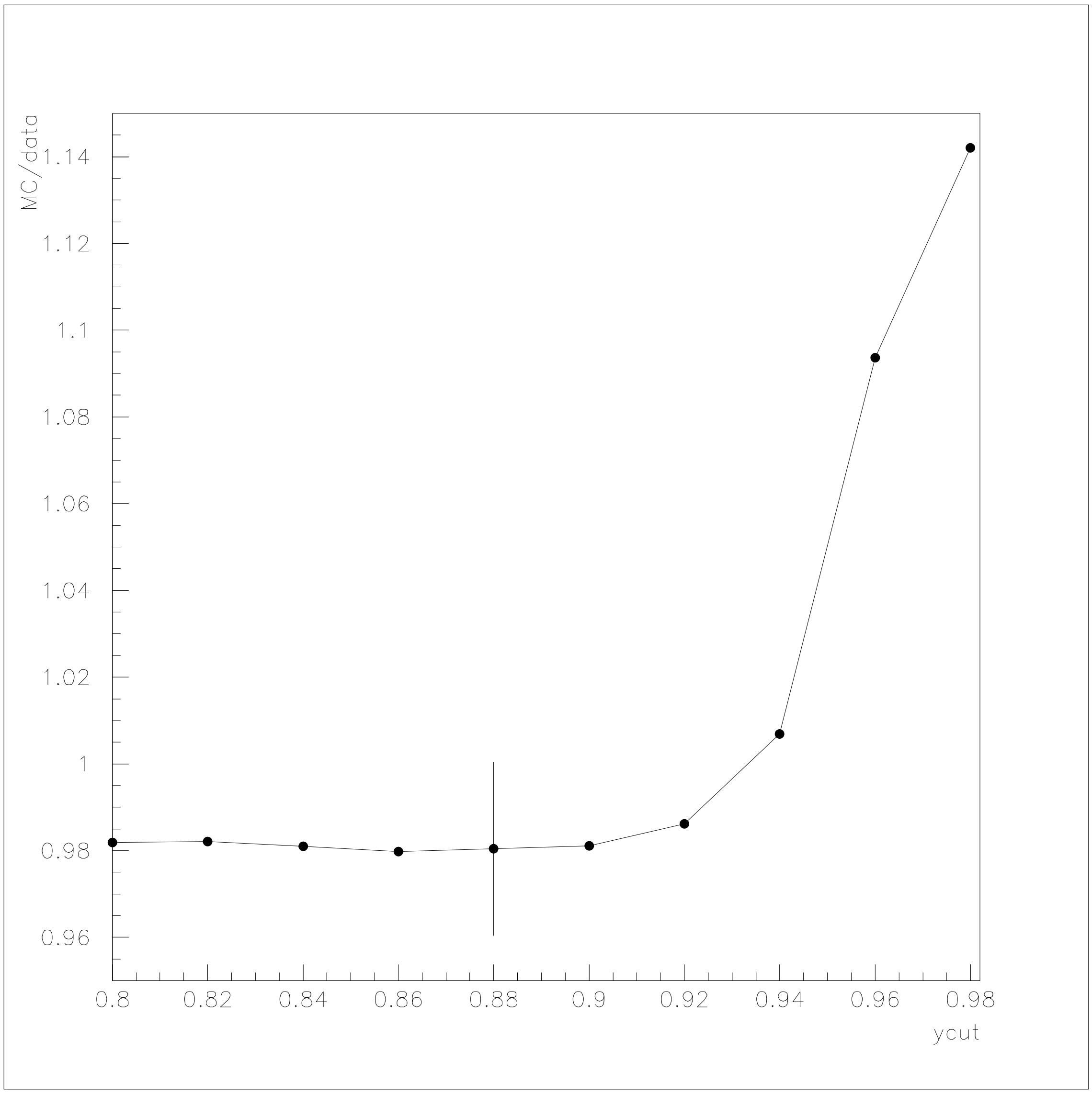

Figure 27: Ratio between the integrated histograms MC and data for the output of the neural network $\tau \rightarrow e\nu_\tau\bar{\nu}_e$ in the figure 26.

source	ΔB_e	ΔB_μ
y_{cut} in τ and $\tau \to l$ selection	0.0007	0.0012
finite MC statistics	0.0011	0.0012
N^o hidden layers	0.0011	0.0011
training parameters	0.0011	0.0011
background substraction in lepton sel.	0.0015	0.0020
background substraction τ sel.	0.0001	0.0002
total	0.0025	0.0031

Table 13: Systematic errors in τ leptonic branching ratios.

-Finite MC statistic: This is the easiest systematic error, and it has been computed as a statistical error from MC. Note that only the test sample has an influence in the branching ratio. The influence of the train sample is an uncertainty in the connection weights, but when the neural network is fixed, this has no contribution to the error in the branching ratio.
-Number of hidden units in the neural network: The computation of the error coming from this source has been done varying the number of hidden units (one or two more units) and computing the new efficiencies. Neural networks with direct connections from the input layer to the output layer have also been considered for computing this error (different topology).
-Training influence: The training parameters (η, α) have been varied within a 10% of their initial values. After the training procedure, the new efficiencies are used to compute the error rate. A larger change of these parameters is meaningless because in that case the neural networks can not learn solving the problem.
-Disagreement between MC and data (background substraction): For computing this error, the efficiencies have been varied within 1σ of their statistical error. The effects contributing to this error are either the detector effects or the effects of the composition of the training sample. The detector effects are very small because it is well simulated in the critical regions for the present analysis (near the effective cuts). The used training sample has not the same composition as the data sample. This is not impportant for the signal to background ratio because the neural network can separate them. But the ratios among the different backgrounds are important. For different compositions of the background, the neural network selects different features. Then, using a training sample distribution different to the data distribution leads to a systematic error. However, this effect should be small because it affects to the backgrounds, and they are a small proportion of the sample.
The systematic errors that could aim a reduction are those coming from background substraction. The other ones are very difficult to reduce, because they are associated to the neural network complexity.
The contributions of these sources to the systematic errors can be seen in table 13.

9 Conclusions

In this work Neural networks have been shown as a competitive tool for the analysis of events produced in high energy reactions.
The most important advantages of this method are:
-Automatic learning of correlations in a multidimensional space, which avoids the need of fine tuning for the cuts applied in the selection of the sample.
-the signal-to-background ratio can be smoothly adjusted by varying y_{cut}.

On the other hand, the most important problem is that:
-There is a limitation on the level of purity that can be achieved. This is due to the fact that the training method is an approximation. Then, the network output does not go from 0 to 1, but from $0+\epsilon$ to $1-\epsilon^{'}$ and is meaningless to select $y_{cut} < 0+\epsilon$ or $y_{cut} > 1-\epsilon^{'}$. This problem can be avoided with new training algorithms or larger MC samples.

All the results obtained in this work are in agreement with the values obtained by other experiments [28], but with larger error on $BR(\tau \rightarrow \mu\nu_{\tau}\bar{\nu_{\mu}})$, as espected because the original variables were not optimized for this channel.
The measured value of R_{τ} is consistent with the values obtained from the measurements of the mass and lifetime of the τ lepton.

10 Appendix: Fitted branching ratios

The aim of this part is to obtain the leptonic branching ratios of the τ lepton using a method different of the method used in the main body of this article. This new procedure consists in a fit of the output distribution of the data to the output distribution for the different MC samples.
The initial data sample are the events selected as $e^+e^- \rightarrow \tau^+\tau^-$ by the first neural network described in the text. Then, a fit is done to the outputs of the neural networks identifying $\tau \rightarrow l\nu_\tau\bar{\nu_l}$ using the function $\lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3$ where f_1 is the neural network output for MC of $\tau \rightarrow l\nu_\tau\bar{\nu_l}$, f_2 is the neural network output for MC of other τ decays (the relative branching ratios for the other τ decays are known) and f_3 is the neural network output for background, as is shown in figures 28 and 29.
After doing this fit using a least squares method, the results are:

A) $BR(\tau \rightarrow e\nu_\tau\bar{\nu_e}) = 18.43 \pm 0.23\%$
B) $BR(\tau \rightarrow \mu\nu_\tau\bar{\nu_\mu}) = 17.62 \pm 0.38\%$

In both cases, $BR = \lambda_1/(\lambda_1 + \lambda_2)$, and λ_3 is the background rate in the τ sample. Then λ_3 would have the same value in both cases, because the same initial sample is used. The results for λ_3 are:

A) $\lambda_3 = 9.24 \pm 0.53\%$
B) $\lambda_3 = 10.04 \pm 0.75\%$

These values are compatible between them and with the value of the background computed in the text (9.33%).
Anyway the errors showed are those from the fit procedure, and they do not include all the systematic errors quoted in the text. In all cases this method should be taken only as a consistency check.

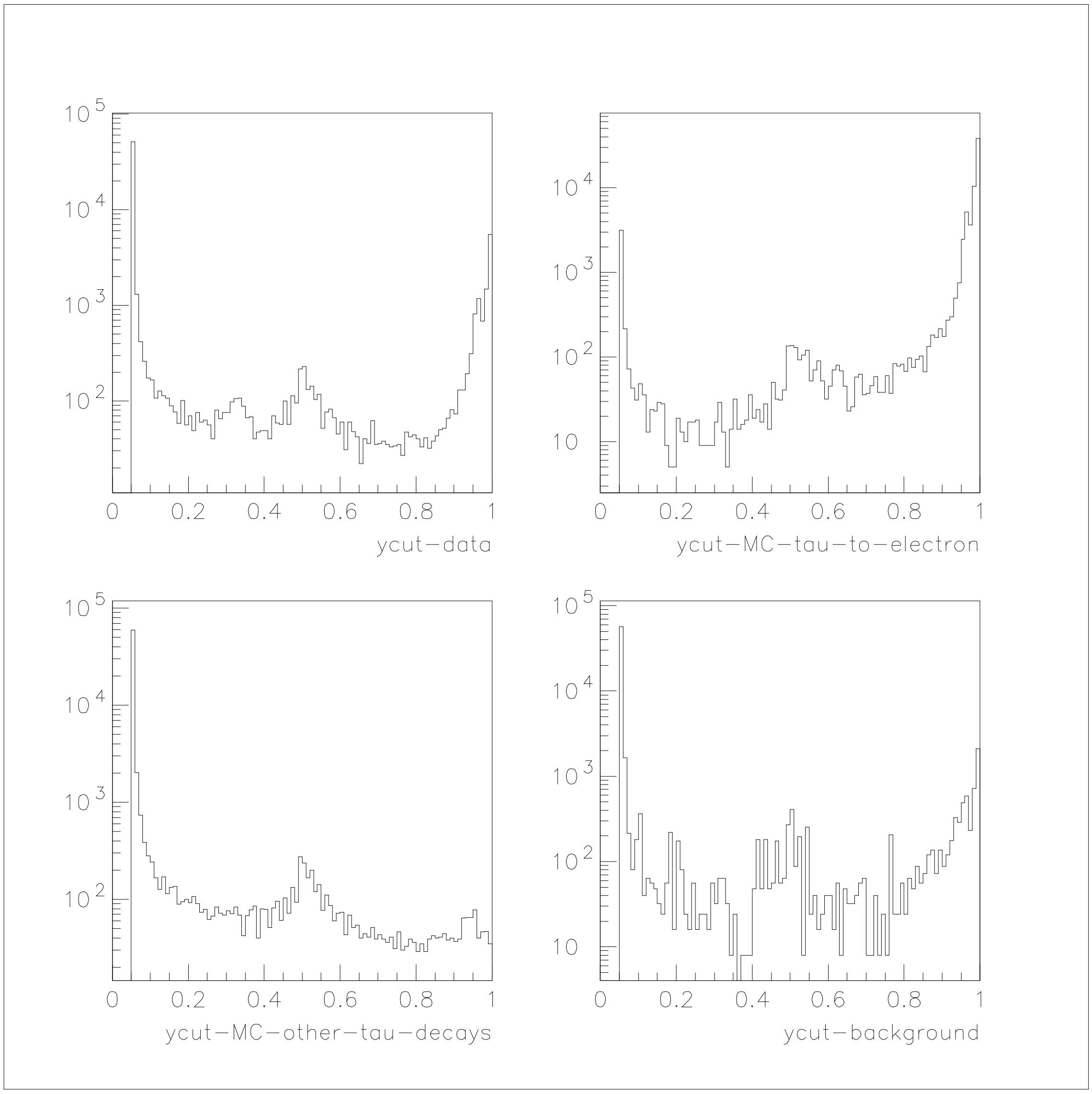

Figure 28: Data and MC distribution for the $BR(\tau \rightarrow e\nu_\tau\bar{\nu}_e)$ fit.

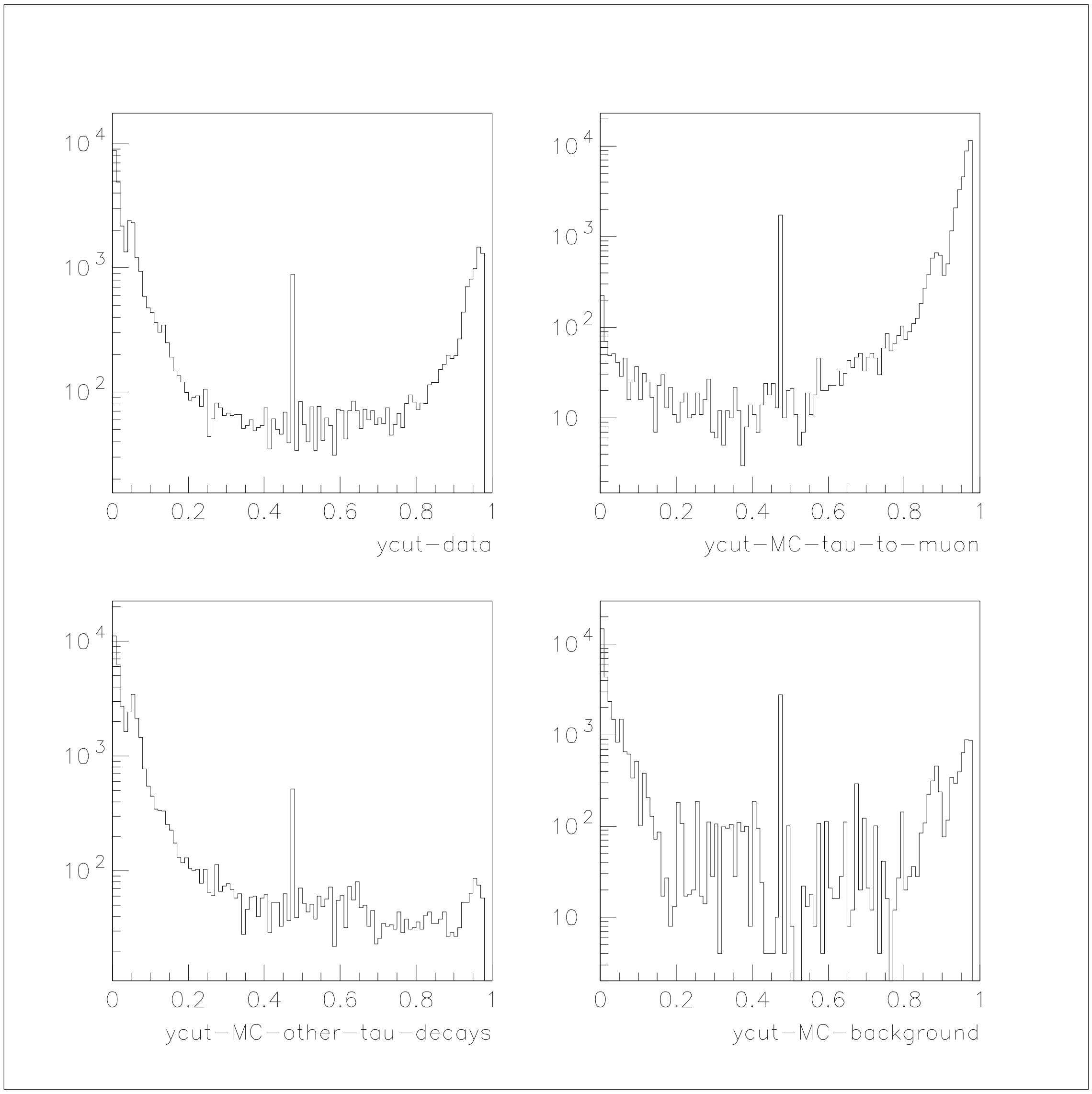

Figure 29: Data and MC distribution for the $BR(\tau \to \mu\nu_\tau\bar{\nu}_\mu)$ fit. The peek near $y_{cut} = 0.475$ are bad reconstructed events.

References

[1] C. Bortolotto, A. De Angelis, N. De Groot and J. Seixas Udine Report 91/09/CB-Rev. Universita'di Udine.

[2] B. D. Ripley, Statistical aspects of neural networks, in Chaos and Networks-Statistical and Probabilistic Aspects (eds. O. E. Barndorff-Nielsen, D. R. Cox, J. L. Jensen & W. S. Kendall), London, Chapman & Hall.

[3] S. Dennis & S. Phillips, available by anonymous ftp from pt.cs.cmu.edu (128.2.254.155), 'Analysis tools for Neural Networks'.

[4] L. Lönnblad, C. Peterson, T. Rögnvaldsson, Nucl. Phys. B **349**, 675(1991).

[5] R. Odorico, University of Bologna, DFUB 92/8, Talk at the XXVIIth Recontres de Moriond, QCD and High Energy Hadronic Interactions, March 22-28 1992, Les Arcs, Savoie, France.

[6] E. Fiesler, 'Neural Network Classification and Formalization', submitted to Computer Standards & Interfaces, special issue on Neural Network standards, John Fulcher editor, North-Holland, Elsevier Science Publishers. B. V. ; Amsterdam, The Netherlands.

[7] S. A. Solla, E. Levin, M. Fleisher, Complex Systems **2**, 625(1988)

[8] J. Hertz, A. Krogh, R. G. Palmer, 'Introduction to the Theory of Neural Computation', (Redwood City) Adisson-Wesley (1991).

[9] G. E. Hinton, Artificial Intelligence **40**, 185(1989)

[10] B. D. Ripley 'Neural Networks and Flexible Regression and Discrimination', in Statistics and Images, ed. K. V. Mardia. Advances in Applied Statistics, (Abingdon:Carfax).

[11] D. E. Rumelhart & J. L. McClelland eds. 'Parallel Distributed Processing: Exploration in the Microstructure of Cognition', vols. 1 & 2. The MIT Press.

[12] B. D. Ripley 'Neural networks and related methods for classification. J. Roy. Statist. Soc. B. **56**.

[13] A. L. Blum & R. L. Rivest, Neural Networks **5**, 1117(1992).

[14] J. S. Judd, 'Neural Network Design and the Complexity of Learning', The MIT Press.

[15] A. N. Kolmogorov, Dokl. Akad. Nauk. USSR **114** 953(1957) (In Russian).

[16] K. Funahashi, Neural Networks **2**, 183(1989).

[17] G. Cybenko, Math. Control Systems Signals **2**, 303(1989).

[18] K. Hornik, M. Stinchcombe, H. White, Neural Networks **2**, 359(1989).

[19] G. Cybenko, Technical Report, Dept. of Computer Science, Tufts University, 'Continuous valued Neural Networks with two Hidden Layers are sufficient'.

[20] J. J. Hopfield & D. W. Tank, Science **233** 625(1986).

[21] J. J. Hopfield & D. W. Tank, Biological Cybernetics, **52** 141(1985).

[22] G. E. Hinton & T. J. Sejnowski, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Washington,1983) pp. 448-453, 'Optimal perceptual inference'.

[23] T. Kohonen, Proc. IEEE **78**, 1464(1990).

[24] G. Stimpl-Abele, accepted for publication in Comp. Phys. Commm. (1991),'Fast track finding with neural networks'.

[25] M. Bäker, T. Kalkreuter, G. Mack, M. Speh, preprint DESY 92-126, september 1992.

[26] G. Mack, T. Kalkreuter, G. Palma, M. Speh, preprint DESY 92-070, may 1992.

[27] L3 collaboration, Phys. Rep. **236**, 1(1993).

[28] Particle Data Group, Phys. Rev. **D45**, 1(1992).

[29] L3 collaboration, Nucl. Inst. and Meth. in Phys. Res. A **289**, 35(1990).

[30] W. Marciano & A. Sirlin, Phys. Rev. Lett. **61**, 1815(1988).

[31] E. Braaten, Phys. Rev. Lett. **60**, 1606(1988).

[32] S. Narison & A. Pich, Phys. Lett. B **211** 183(1988).

[33] L. R. Surguladze et al., Phys. Rev. Lett. **66**, 560(1991).

[34] S. Gorishny et al., Phys. Lett. B **259**, 144(1991).

[35] E. Braaten & C. S. Li, Phys. Rev. D **42**, 3888(1990).

[36] L. R. Surguladze, INR preprint P-0639, (Moscow, 1989).

[37] L3 collaboration, L3 note 1446 CERN (1993).

[38] Aspirin/Migraines 6.0; The MITRE Corporation.
Available by anonymous ftp from pt.cs.cmu.edu(128.2.254.155) or from ftp.cognet.ucla.edu(128.97.50.19).

[39] G. Rodrigo & A. Santamaría, Phys. Lett. B **313**, 441(1993).