Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Un Classificatore Neurale basato sulla rete neurale EαNet per l'analisi di dati provenienti da esperimenti di fisica alle alte energie (HEP)

A. Gentile, G. Pilato, S.M. Siniscalchi, F. Sorbello, G. Vassallo, S. Vitabile

RT-ICAR-PA-24-03 **Dicembre 2003**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Un Classificatore Neurale basato sulla rete neurale EαNet per l'analisi di dati provenienti da esperimenti di fisica alle alte energie (HEP)

A. Gentile[2], G. Pilato[1], S.M. Siniscalchi[2], F. Sorbello[1-2], G. Vassallo[2], S. Vitabile[1]

Rapporto Tecnico N:
RT-ICAR-PA-24-03

Data:
Dicembre 2003

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sezione di Palermo
[2] Università degli Studi di Palermo Dipartimento di Ingegneria Informatica

Introduction

High Energy Physics (HEP) experiments generate huge amounts of data that require classification and event discrimination. As example, a run on the Collider Detector at Fermilab generates a dataset of events characterized by the generation of top/anti-top quark couples.
The top/anti-top quark couples was discovered at the Fermilab Tevatron in 1995. This was either the culmination of the nearly two decades of intense research at particle accelerators around the world, or the major triumph for the Standard Model of particle physics since it predicted the top quark existence. In HEP experiments, along with interesting events, background noise is generated by the collision, which occurs in a very small time lapse. Different backgrounds have very different kinematics properties, so HEP data classification is a very complex tasks.
Neural networks have been applied in HEP experiments as function approximators to obtain a functional form which describes some distribution [1], [2], or for event classification, combining information from different variables [3], [4], [5]. On the other hand, neural based high speed triggering devices, normally organized in a hierarchy, are then required to discriminate useful data from background noise [6], [7].

Background

Notoriously, data sets used in typical neural network applications are characterized by large cardinality and unknown statistical distribution. There is in fact no guarantee that input-output pairs be statistically significant when considered under neural network testing, which makes the traditional test-set validation procedure potentially incorrect.
The authors have previously introduced three ”quality factors” to give a measure, without using the test set, of the generalization capability of a feed-forward neural network. Based on the properties of these quality indexes, the E-αNet architecture has been developed and successfully employed in several application contexts [8], [9].
In other application arenas, the authors have developed a simulation environment for a Multi-Layer Perceptron (MLP) design showing large performance ratings in terms of both recognition rate and classification speed. This design uses sinusoidal shaped activation functions for hidden layer neurons and linear functions for output layer neurons. Successful applications of the design have been reported in the area of handwritten character recognition [10] and road sign recognition [11], [12].
The EαNet is a feed forward neural architecture capable to learn the activation function of its hidden units during the training phase. These networks are characterized by low quality factors when compared to traditional feed-forward networks with sigmoidal activation functions. Network learning capability has been obtained through the combination of Powell modified Conjugate Gradient Descent (CGD) [13] and the Hermite regression formula. Hidden layer activation functions are based on the first R Hermite orthonormal functions where R is a priori chosen before the learning process.

Experiments and Results

Balanced 10-fold cross validation

The performances of our artificial neural network, E-αNet, were evaluated on a 10-fold cross validation strategy, balanced version, referred to as 1B Strategy. In this strategy, ten (10) different groups of events comprised four hundred and nineteen (419) background pattern, and four hundred and nineteen (419) "top" patterns were available. Hence, both train and test are balanced. These ten groups were stored onto ten files that are referred to as S1B_01.txt, S1B_02.txt, ..., S1B_10.txt. Each of these files has eight hundred and thirty-eight (838) rows, and nine (9) columns. The first eight columns represent the features characterizing an event, whereas the last column defines the class which an event belongs to (zero for a background event, and one for a top one). Furthermore, nine sets were used to carry out the train phase, and only one set for the test phase. All possible combinations of the ten (10) files were gathered by pooling together the data according to the following equation

(1) $$\binom{n}{k} = \frac{n!}{(n-k)!\,k!}, \text{ where n=10 and k=9.}$$

Each of the above combinations was labelled as G#, where # goes through 1,2, …, 10, and Figure 1 gives a picture of the generating procedure.

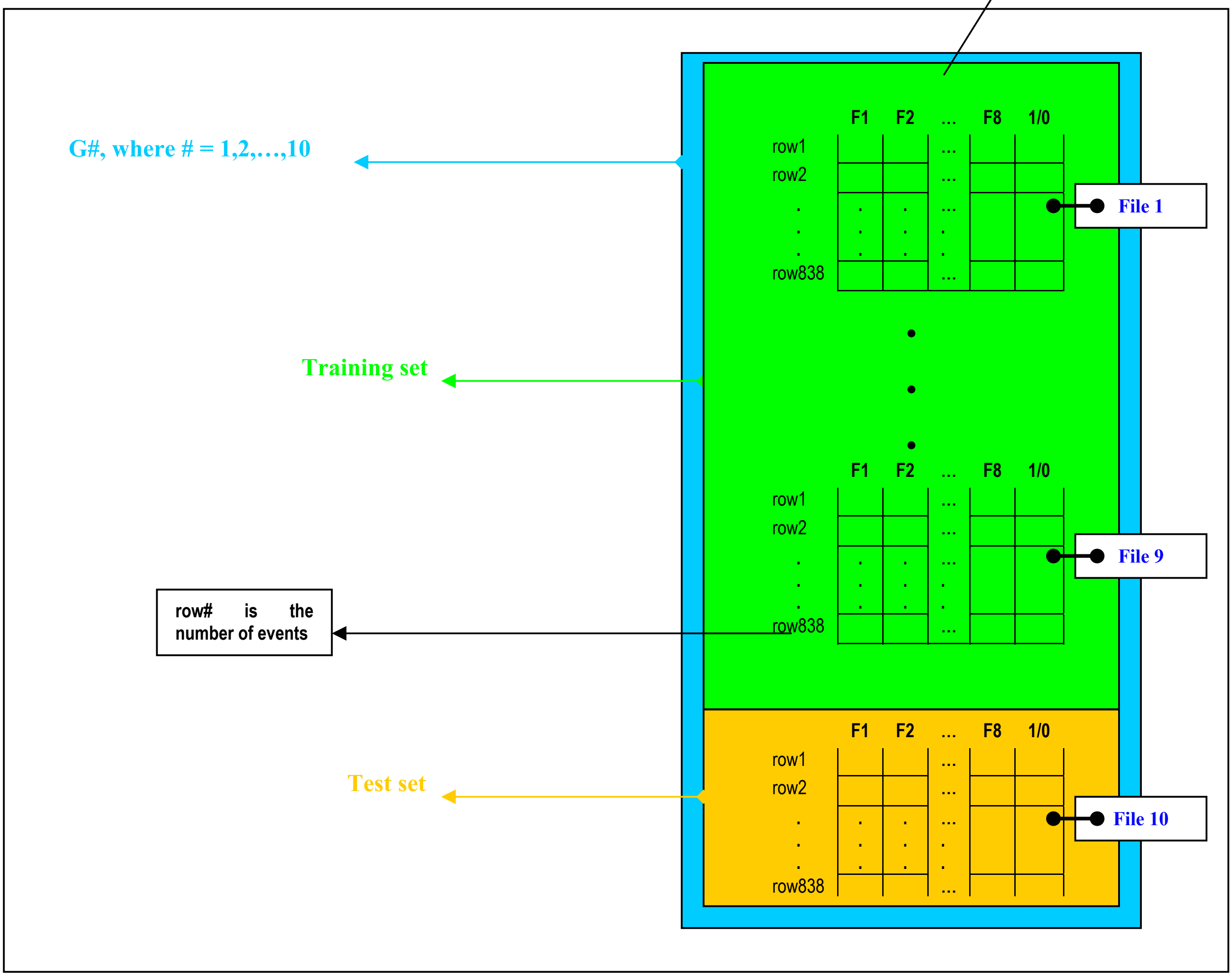


Figure. 1

The classification quality is given averaging over the ten test seasons the efficiency and purity factors, which are defined as follows,

(2) $$Efficiency = \frac{TruePositive}{TruePositive + FalseNegative}$$

(3) $$Purity = \frac{TruePositive}{TruePositive + FalsePositive}$$

Data Pre-elaboration

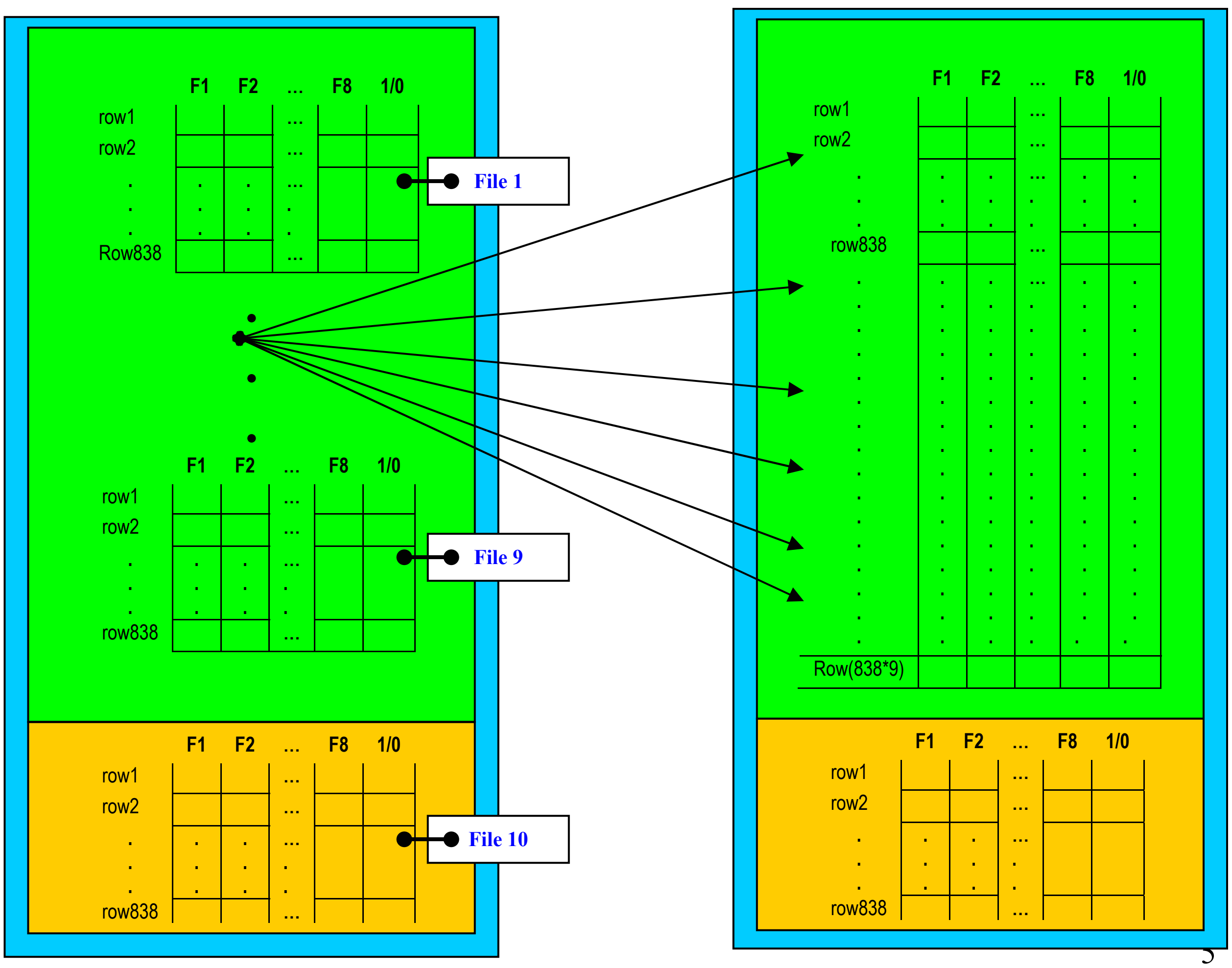
Since E-αNet doesn't work well with the given raw data, a pre-elaboration of the G# groups was performed. For each of the G# a new group, labeled G^{temp}#, was generated according to the bello criteria,

- For each G#: merge together nine training files in only one file, called *set9*
- Suppose each feature (F_j) of the *set9* as belonging to a Gaussian with means μ_j and variance σ_j^2, where j runs from one (1) to eight (8). Each column is then normalized as follows,

(4) $$z_j = \frac{(F_j - \mu_j)}{\sigma_j} \sim N(0,1)$$

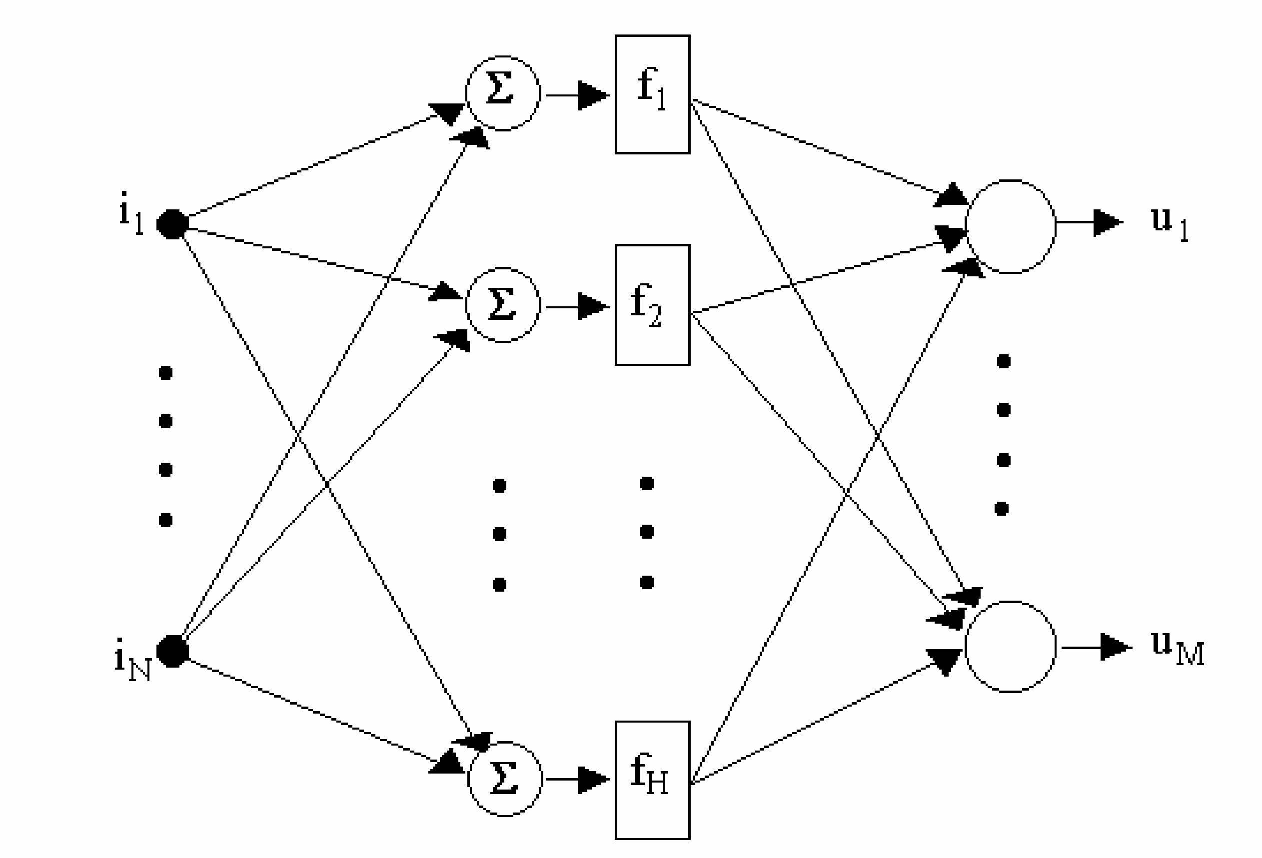
- All features of the test set are normalized using μ_j and σ_j^2 and (4).

The new groups are referred to as G°#, and Fjs are the new features.



Architecture set-up

E-αNet is an artificial neural network characterized by different parameters, some of them are tied up to the problem others have to be assigned by the developers.
E-αNet parameters are input/output numbers, Hermite polynomials numbers, and number of the hidden units.
The number of Input/Output is established and in our problem is eight (8) for inputs and two (2) for output, indeed fourteen Hermite polynomials have been chosen to catch the best performance.
Setting the number of hidden layer is more difficult than others parameters, then CSAI developers have decided to extract from experimental method (try and error?). In the following of this paragraph the above strategy will be shown.

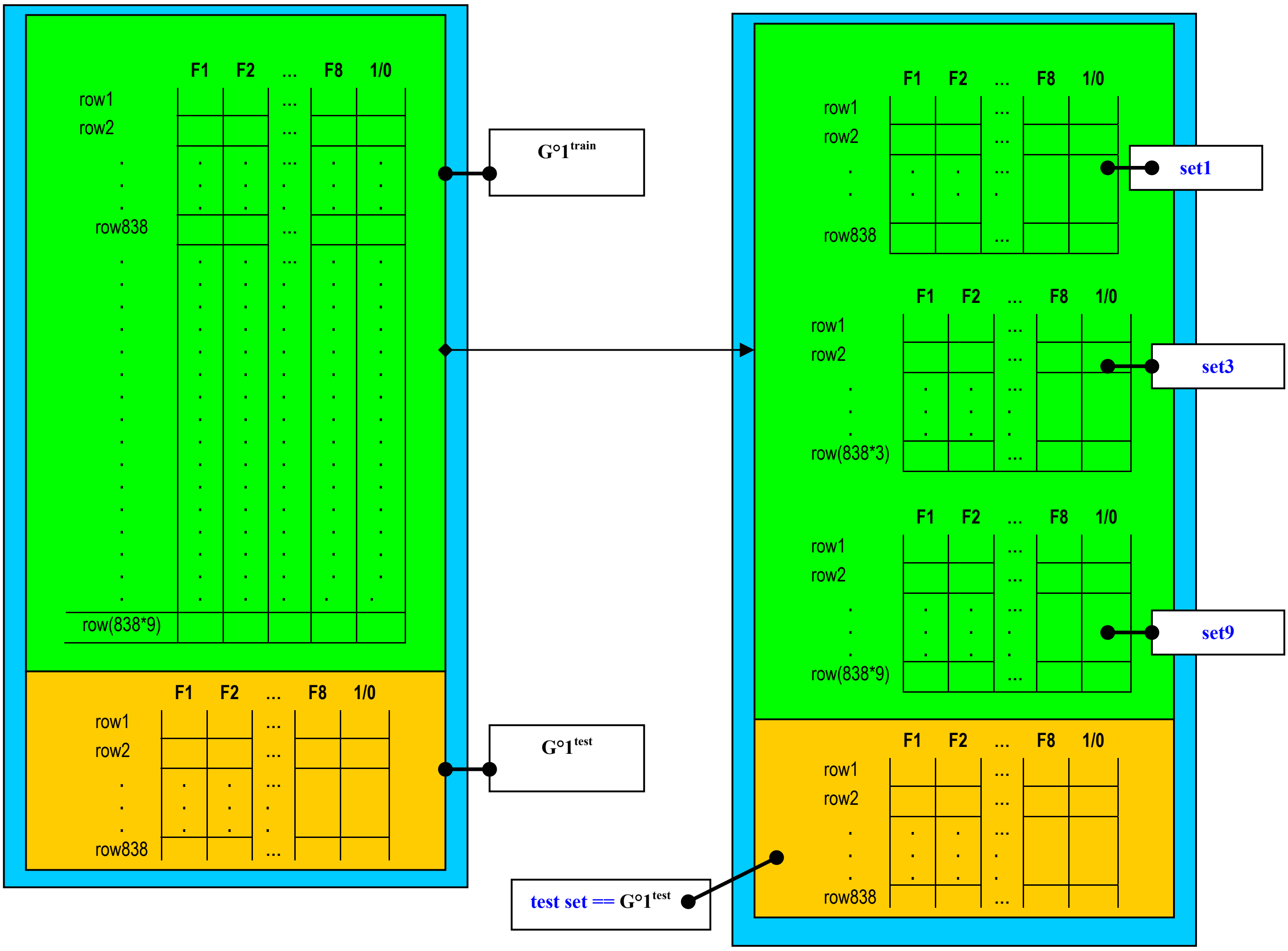


- **Try and error strategy**

G°1 is the only group among the ten ones that will be considered; it will not use directly but it is still processed to obtain a new training set; indeed the test set is the same.
Exactly $G°1^{train}$ has split in three sub-sets that we will call set1, set2 and set3. Set1 is compounded by the first 838 rows of $G°1^{train}$; set2 is compounded by the first 838*3 rows of $G°1^{train}$ and at last set9 corresponds to the whole $G°1^{train}$
After the pre-elaboration of the train data set the actually training starts:

I. E-αNet has trained with the events of the set1. After 100000 iterations a temporary train error and temporary weights will be obtained.
II. Starting from the weights of the phase I, the ANN has trained using set2. Also now, after 100000 iteration, a temporary train error and temporary weights will be obtained.

III. It is the last phase, it starts from phase II, the train set corresponds with set9. The outputs are a train error and the weights for the given architecture.



For each train phase a corresponding test phase is run, so efficiency and purity parameters and test error can be computed. These values and train error are used to choose the right architecture.

	8-12-2		8-20-2		8-25-2		8-30-2		8-64-2	
	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %
Set1	0	33,05	0	32,43	0	33,67	0	32,70	0	29,00
Set2	4	31,26	12	29,50	6,32	32,46	3,38	32,46	0	33,41
Set3	22,3	21	22,1	22,05	19,5	22,32	19,1	24,82	12,6	30,43

The symbol 8-N-2 will be used to indicate the architecture that has got N hidden layers.
The tested architecture are: 8-12-2, 8-20-2, 8-25-2, 8-30-2 and 8-64-2. The above table shows all results. According with the CSAI researcher knowledge the best compromise between test, train error and complexity of the architecture is given by 8-25-2 one.

Balanced 10-fold cross validation results

Now to test the 8-25-2 architecture by (with) 10-fold strategy it is needed that every G°# group is processed as it is done for G°1. Then each group is submitted to the I, II and III try and error phases, and besides for each group is run a test phase. Table summarized all results.

	Group 1		Group 2		Group 3		Group 4		Group 5	
	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %
Set1	0	32,7	0	34,84	0	33,65	0	35,68	0	32,34
Set2	9,62	31,86	7,59	36,16	8,81	32,34	9,14	34,13	9,42	32,34
Set3	19,53	22,32	18,59	27,09	19,37	28,04	18,85	26,49	19,76	24,94
Efficiency set3		0,773		0,732		0,727		0,747		0,773
Purity set3		0,778		0,727		0,717		0,731		O,738

	Group 6		Group 7		Group 8		Group 9		Group 10	
	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %	Train error %	Test error %
Set1	0	34,84	0	36,87	0	32,22	0	32,22	0	31,03
Set2	9,22	31,61	8,91	31,86	8,15	30,07	8,27	33,05	8,47	32,22
Set3	19,96	23,87	19,39	26,25	19,61	25,78	19,57	24,22	18,81	25,78
Efficiency set3		0,758		0,732		0,758		0,782		0,735
Purity set3		0,762		0,739		0,734		0,745		0,745

train average error %			**test average error %**		
set1	**set2**	**set3**	**set1**	**set2**	**Set3**
0	8,77	19,3	33,6	32,5	25,4

efficiency average	0,752
purity average	0,742

The efficiency is high when the system is biased towards the positive class (its class estimate is likely to be positive). On the other hand, when the system faces with high purity, it classifies a point as positive (the true target will probably be positive). Figure 2 shows the purity variation as a function of efficiency.

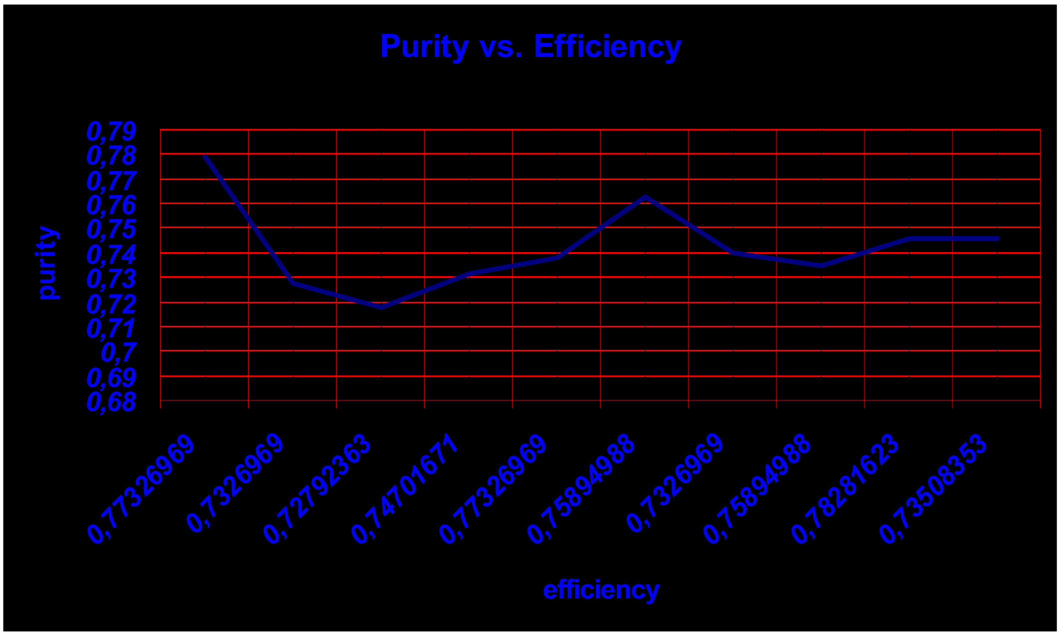

Purity vs. Efficiency
purity
0,79
0,78
0,77
0,76
0,75
0,74
0,73
0,72
0,71
0,7
0,69
0,68
0,77326969
0,7326969
0,72792363
0,74701671
0,77326969
0,75894988
0,7326969
0,75894988
0,78281623
0,73508353
efficiency

Figure 2

References

1. Suman B. Beri, Pushpalata Bhat, Rajwant Kaur, Harrison Prosper. Top Quark Mass Measurements Using Neural Networks. In VII International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2000)

2. C. Bhat, P. Bhat. Using Ensembles of Neural Networks in HEP Analysis. VII International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2000)

3. Chris Hays and Ashutosh V. Kotwal Using a Neural Network for Electron Identi.- cation. CDF Note CDF/DOC/ELECTRON/CDFR/5810, 2002

4. Joshua P. Tuttle, Chris Hays, Ashutosh V. Kotwal Neural Networks for Electron and Photon Identi.cation. CDF Note CDF/DOC/ELECTRON/CDFR/5791, 2001

5. Catalin I. Ciobanu Richard E. Hughes Brian L. Winer Using Neural Networks to Identify Single Top. CDF Note CDF/ANAL/TOP/GROUP/5370, 1999

6. Fent et al. The Realization of a Second Level Neural Network Trigger for the H1 Experiment at HERA. AIHENP'96, Lausanne, Switzerland, Sept.2-8, 1996

7. L. Janauscheck et al. Arti.cial Neural Networks as a Second Level Trigger at the H1 Experiment at HERA Performance Analysis and Physics Results. AIHENP'99 - University of Crete and FORTH, GREECE - April 12-16, 1999

8. Gaglio S., Pilato G., Sorbello F., Vassallo G.: Using the Hermite regression formula to design a neural architecture with automatic learning of the 'hidden' activation functions. Lecture Notes in Arti.cial Intelligence 1792, pages 226–237. Jan 2000 Springer-Verlag.

9. Pilato G., Sorbello F., Vassallo G.: An innovative way to measure the quality of a neural network without the use of the test set. International Journal of Artificial Computational Intelligence, Vol. 5 No 1, 2001, pp:31–36

10. F. Sorbello, G. Gioiello, and S. Vitabile. Handwritten Character Recognition using a MLP, chapter 5, pages 91–119. L.C. Jain and B. Lazzerini - CRC Press, 1999.

11. S. Vitabile, A. Gentile, F. Sorbello A Neural Network based Automatic Road Signs Recognizer. Proc. of 2002 IEEE World Congress on Computational Intelligence - International Joint Conference on Neural Networks (IJCNN)

12. S. Vitabile, A. Gentile, F. Sorbello Real-Time Road Signs Recognition on a SIMD Architecture. WSEAS Transactions on Circuits and Systems, Issue 3, Volume 3, May 2004, pp. 664-669, ISSN: 1109-2734.

13. M.J.D. Powell. Restart Procedures for the Conjugate Gradient Method. In Mathematical Programming, (1968), 12, pp. 241-254.