The Current Status of Linux Driver of MiND PCI board and Results of Performance Tests of SAND/1 Neuro-chip

Ashot Chilingaryan , Ararat Vardanyan
H. Gemmeke*, W. Eppler*, T. Fischer*

Yerevan Physics Institute, Cosmic Ray Division, Armenia
*Haubtabteilung Procesdatenverarbeitung und Electronics,
Forschungszentrum Karlsruhe, Germany

Abstract:

The possibilities of the SAND neurochip, supporting hardware application of neural networks of different kind, are investigated as an intelligent device for fast decision making and critical applications in respect to reliability. The NeuroLution PCI board with four of these SAND chips integrated on-board was optimized in this work for use under Linux OS, in order to be able to use the specialized SAND chip in various scientific and industrial applications.

The choice if the Linux operating system is caused by the reliability and stability requirements of many applications from one and the open source advantages from the other hand, which allows to handle the system routines in low level and gives a possibility to construct the hardware driver operating in optimal way, providing features for real-time applications. The PCI bus-NeuroLution board with optimized Linux driver was investigated. The performance of the specialized SAND neurochip on different types of NN applications is estimated and the results are compared with powerful general purpose CPU's. The advantage of the system, with more than one SAND chip connected in parallel is highlighted. The short description of the driver routines and their interconnections is presented. The possibility of using the NeuroLution board in High Energy Cosmic Ray Physics experiments as

a third level trigger with pattern recognition and on-line data analysis is investigated.

As example the MAGIC cosmic ray experiments is discussed. The MAGIC Chernokov telescope will work in heavy background conditions. The low energy threshold of the telescope (30 GeV) combined with low pixel threshold will pose very strict requirements on pattern recognition of the signal image. Only the usage of all distinctive information contained in the telescope camera containing 500 pixels will provide a possibility to enlarge the signal-to-noise ratio. Working with such huge inputs new network training algorithms and powerful training accelerators are required. Tests, using simple models of signal and background for NN training prove that both pure software and hardware networks provide 98% of correct classifications, proving adequateness of the used ANI package net raining methods for such a huge amount of network weighs. Using the NeuroLution board 108000 iterations were done within 6 hours. The Celer on 433 MHz CPU needed 6 days for this amount of iterations. Such bit advantage is speed will provide the possibility for the training of very big networks and use the NeuroLution board as a very fast intelligent trigger.

Introduction:

The SAND (Simple Applicable Neural Device) chip designed for accelerating various neural net applications is a digital hardware realization (built with $0.8_{\mu m}$ CMOS technology) of NN models based upon the principle of systolic array [1, 2, 3].

For the stand alone operation it requires only a few external components, such as:

- Look Up Table LUT - for non-linear transfer function calculation
- Memory (WRAM) - for storing NN weights and intermediate data,
- Sequencer - for the overall memory management as well as the control of SAND itself.

The principal features of SAND/1 chip are:

- Maximum Operation Clock - 40MHz
- Cascadable architecture
- Calculation of scalar product and vector distance
- Extreme value search (minimum and maximum)
- CUT-function with over/underflow recognition
- On-line adaptation of arithmetic precision
- Activation function as look-up table external to chip
- Parallel processing support
- The following neural networks are supported:
  - Multilayered Perceptron (MLP)
  - Radial Basis Function (RBF)
  - Self Organizing Maps (SOM)
- Architecture:

- 16 bit weights and input activities
- 40 bit internal precision
- Processing of packets consisting of 4 data words
- Max. 65K weights for any configuration of NN

- Data I/O
  - Input activities normalized to the range $-1.0 \ldots +1.0$
  - 8 fixed-point formats available for the weights (0.25...128)
  - 2 scalable output formats: linear output or any transfer function
  - Continuous data flow on the weight and activity busses (max. 100Mb/s)

With a maximum clock frequency the single SAND achieves a performance 200 MCPS (Mega Connections Per Second).

The Linux Driver Status:

To take the advantage of such a specialized neuro-chip and to use it in real-time applications such as Pattern Recognition, Image Processing, Control Engineering, Fast Intelligent Triggers for High Energy Physics Experiments, the MiND (Multipurpose integrated Neural Device) was designed to integrate the SAND on a PCI bus based device to be compatible with today's advanced architectures and to provide a easy to use design via SAND to PC connection and hardware control via soft routines. Another great advantage of MiND board is that it integrates four SAND chips connected in parallel and attains a further increase of performance up to 800 MCPS.

To drive the MiND PCI board installed on a PC, and to program applications for SAND chip, it is necessary to develop a hardware driver as an addition o the OS (Operating System), which runs on a PC and controls its operation. Among many existing OS's the Linux OS is intensively developed and used worldwide for various scientic purposes and industrial applications due to its stability and reliability. So it is very important to drive the MiND board under Linux OS to be able to implement it in many time and reliability critical applications.

Due to the very fast development of Linux OS, some changes were required in existing driver of MiND board to tune it with new system facilities and keep it up to date in terms of optimized solutions in controlling overall PC components. Also some bug fixing and optimizations were done in order to be able to use the extreme possibilities provided by SAND chip and to attain as high performance as possible.

Particularly the following was done in attempt to drive the MiND boar under Linux in optimal way and to take the full advantage of SAND neuro-chip:

1. Added support for new 2.2.x kernel
2. Added support for IRQ sharing with other devices
3. Kernel-level communication with device is handled via IRQ, which increases the efficiency of driver itself
4. Added read ahead mode (Input data for next event are uploaded into MiND board, while for the current event calculations are in process)
5. Added delay calibration function for reading results from SAND, which enables to use the MiND board on PC's with different performance, and

increases the performance as compared with fixed timeout system implemented in old driver

6. Some unnecessary parts of source code are removed in order to avoid performance losses
7. Corrected errors with memory management, which were preventing normal operation of large networks
8. Added debugging capabilities for all network types
9. Added support of different CUT Mask selections for different layers, increases the precision
10. Optimized intermediate data conversions from system memory to MiND board increases the I/O performance
11. Added support for linear output in hidden layers
12. Error handling system is fully rewritten
13. Added data swap support for off-line processing of very large input data
14. New software emulator for testing and comparisons
15. New easy to use high level library (SAND LIB.CC) can be used to write user defined applications for SAND chip
16. Present status of driver:
    - Supports MLP networks with max. 512 nodes in each layer, the number of hidden layers is limited only by total number of 65535 weights in any network topology
    - New user friendly configuration file to load and run different networks
    - RBF and SOM networks are also supported, but not fully tested yet
17. New routine library to attain the on-board learning possibility is developed and tested
18. New SAND class library for use in object oriented programs is under development

In Figure 1 the block diagram of current driver is plotted. The different layers of programming are filled by different colors and represent the different levels of driver routines. The short description of routines and levels see bellow:

1. Top Level:
   SAND.CC SAND class library for use in object oriented programming
   SAND LIB.CC high level user interface to SAND
2. High Level:
   SAND CONFIG.CC *SAND* configuration file processing
   SAND EMU.CC software emulator for checking and testing purposes
   SAND UTILS.CC different utilities like memory management, etc…
   SAND ERR.C error handling routine
   SAND NN.C main SAND operations performing: Load Weights, Load LUT, *SAND* input, SAND output, Init Net
3. Low Level:
   SAND TRANSFER.C builds SAND commands and sends to board via driver
   SAND DRV.C SAND driver, main communicator with KERNEL MODULE
4. Kernel Level:
   KERNEL MODULE sends requests to, and reads response from MiND board

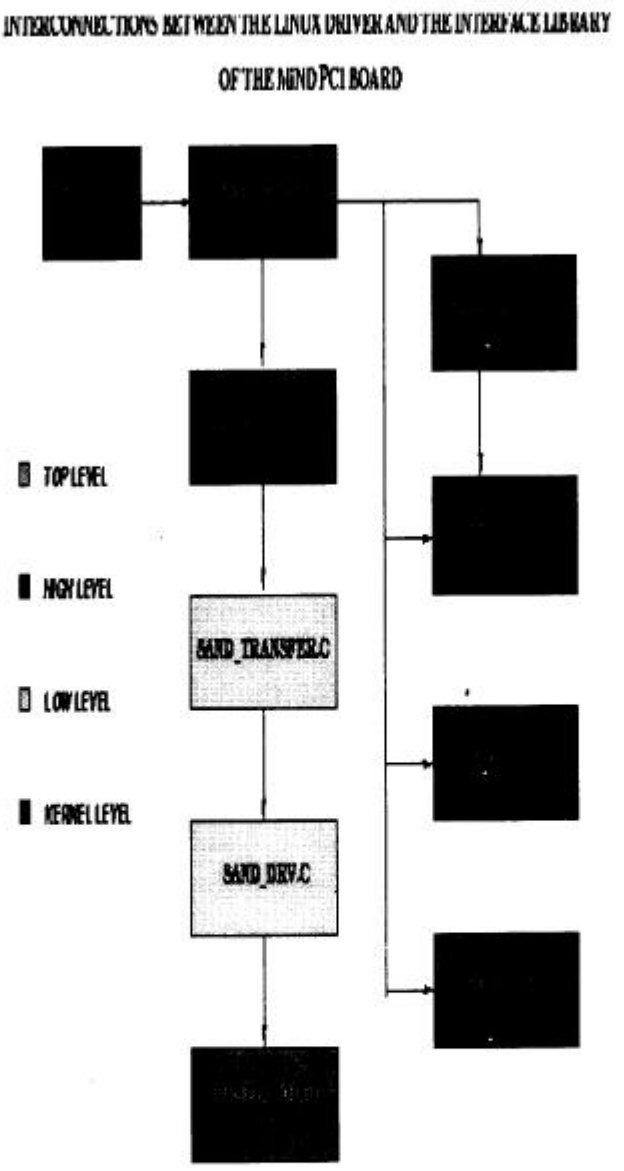

Figure 1: The block diagram of MIND PCI board driver

Testing and Comparisons:

The amount of changes, add-ons and optimizations, which were described above shortly, made on the skeleton of the existing Linux driver, have brought forth the completely new driver kit. Taking into account the crucial importance of several innovations and optimizations of a major (basic) character, such as *real ahead mode, automatic delay calibration, kernel level communications via IRQ, error handling and debugging, etc..., the complete checks and new tests were required.*

Successful checks and tunings were followed by testing of the SAND chip performance and precision. Performance tests are very important, in order to highlight the benefits and limitations of use of such a specialized device in different applications. Furthermore, comparisons with general purpose microprocessors are interesting, because of the intensive development, and very large augment of the performance of now-days general purpose microchips (CPU's). Comparisons were done with well known and widely used in many scientific and industrial fields microprocessor of INTEL corporation. Particularly, it was a system with 6^{th} generation INTEL Celeron 433MHz CPU, on which the MiND PCI board was installed and running.

The results of comparisons are presented in graphs bellow. All tests were done for MLP networks, using 2000 input vectors for any configuration of NN. Different kind of performance tests are presented in this paper:

1. Dependence of the calculation time on the number of total weights in NN, when the rest of all NN parameters are fixed (number of input neurons, number of output neurons, number of hidden layers and the total number of neurons in all hidden layers).
2. Dependence of the calculation time on the total number of neurons, when all other parameters are fixed.
3. Dependence of the calculation time on the total number of Input/Output neurons, when the number of Output/Input neurons is fixed.
4. And finally, the general case, when nothing is fixed and the network configuration is smoothly growing up.

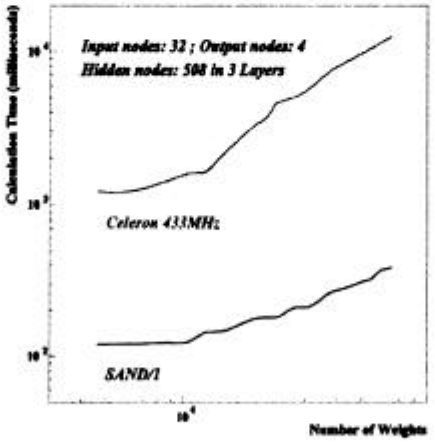

Figure 2: Dependence of calculation time on the total number of weights for medium networks

The next important point is the precision of calculations, SAND results are compared with CPU results (CPU results in double precision mode were considered as true, when obtaining the SAND precision). Due to the CUT-Mask function, implemented in SAND chip, two comparisons are interesting:

- The precision dependence on the number of layers in neural net

- The precision dependence on the activation function of neurons (linear of sigmoid)

Figure 2 displays the dependence of calculation time on the total number of weights of NN. As one can see, for SAND chip the calculation time is approximately constant and 2 - 3 times less than CPU calculation time, at the minimal number of weights (600) for medium networks. While the calculation time for CPU is increasing with increase of number of weights by factor of 2, thus the advantage of SAND chip for networks with 2000 weights is more than 4 - 6 times.

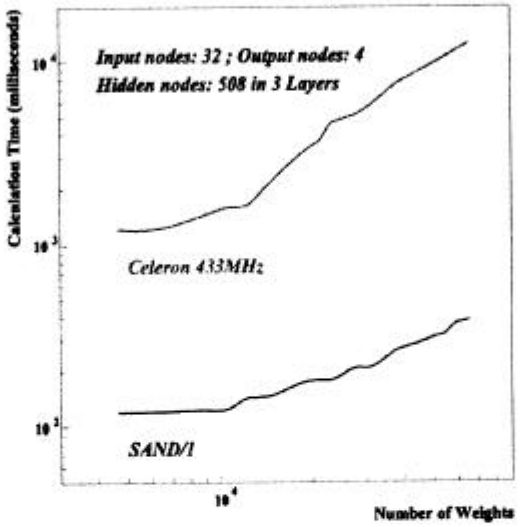

Figure 3: Dependence of calculation time on the total number of weights for large networks

Figure 3 shows the same dependence for large networks, is easy to recognize the difference as compared with medium networks. The calculation time for SAND increases, with increase of weights, because the range of weights is very wide, from 4000 up to 65000 weights, which is very near of SAND's upper limit. The difference of calculation time for largest and smallest networks in this graph is about 200 milliseconds (the factor is 3), while the calculation time for Celeron CPU changes dramatically, approximately by factor of one order. And the difference of calculation time for SAND chip and CPU is about 30 times, while for the case of 5000 weights it is 10 – 12 times. So, for very large networks the SAND chip gives a superior performance as compared with 433MHz CPU. What is also notable in this figure, is the stable value of calculation time for SAND chip up to 10000 weights, this value changes rather slow (only by 20%) for CPU too. From Figure 4 one can see, that the dependence of calculation time on the total number of neurons, when the other parameters of NN are fixed, the behavior of SAND chip and CPU are similar. This test gives the estimate of benefit of using lookup table for sigmoid activation function calculation. The rough estimation looking on this graph is done in the following way: the calculation time for SAND from minimum number of neurons to maximum number of neurons increases from 100 milliseconds up to about 200 milliseconds.

For CPU this change is about 2500-1500=1000 milliseconds. So, the difference by factor 10 is of course notable.

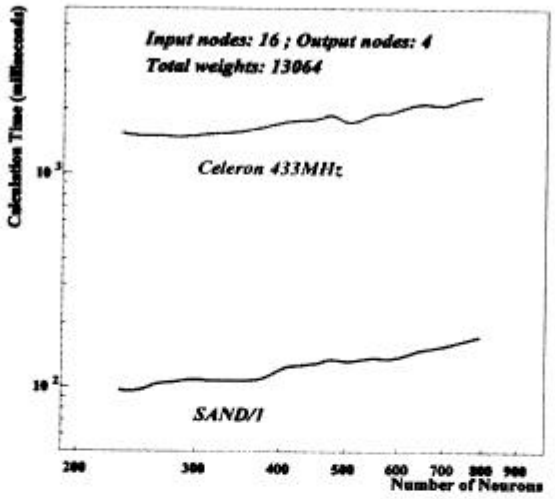

Figure 4: Dependence of calculation time on the total number of neurons in NN

Table 1: The precision of SAND chip depending on the number of functional layers (FL)

ctivation Function	FL	2 FL's	3 FL's	4 FL's
igmoid	.0001	0.00015	0.0002	0.004
inear	.0005	0.001	0.003	0.02

In Table 1 the precision of SAND chip is presented. As one can see, the results are more precise, when using sigmoid activation function, and it decreases with increase of functional layers. This is caused by the multiple appliance of the CUT-Mask function to the outputs of all neurons from layer to layer.

Among all comparisons the most interesting is the last one, because nothing is fixed and this gives an overall information on SAND performance. Figures 5 and 6 display the dependence of the calculation time on the total number of weights and CPU time to SAND time ratio dependence from number of weights respectively. One can recognize, that the CPU is faster for small networks(up to the 150 weights), but with increase of net size SAND is taking advantage quickly, and for networks with about 1000 weights (medium networks) SAND is 6 – 8 times as faster as the CPU, furthermore, for very large networks the factor reaches up to 30. This can be explained easily, taking into account the difference of operations of CPU and SAND chip. For small networks the calculations are not much, but for the SAND chip the most time is spent for uploading input data (4 input vectors at a time) and reading results (I/O time), while CPU is free of this problem (problem of very small memory on MiND board).

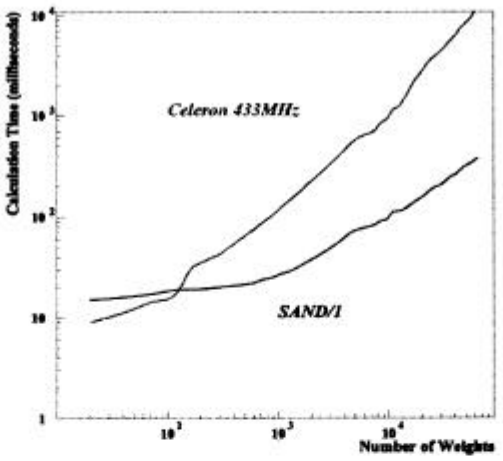

Figure 5: Calculation time versus number of weights for SAND chip and CPU

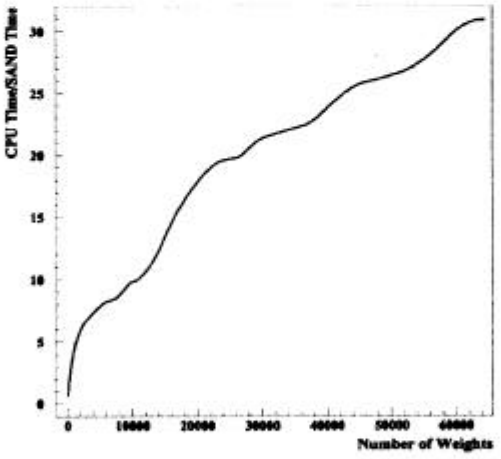

Figure 6: CPU time/SAND time ratio versus number of weights

For medium and large networks, the calculation time is dominating, where the SAND chip benefits very much, and the I/O time is overlapped by grater calculation time by dint of *read ahead mode* (see changes and innovations in the driver above). Figure 7 shows the Input/Output time for different number of input and out neurons in NN (all curves in presented figures are approximations by 30 points).

So, from the comparisons and tests described above, one can conclude, that the SAND/1 specialized neuro-chip has a great advantage of calculation speed (from 5 to 25 times faster) over now-days general purpose PC-class CPU's, when using medium and large configurations of neural networks.

The precision of calculations is also acceptable compared with the 32 bit CPU.

Having in mind the fact, that before applying any neural net for some problem solution, first of all the net should be trained, the real benefit of using SAND chip will be in its application for training purposes. This idea has led out the development of the special libraries and routines in driver kit, for on-board learning application. In present paper we don't bring the detailed description of these libraries and routines. Due to the large abundance of them, and independence from the MiND driver at the same time, they require separate and subsequent description, including not only technique of realization, but different learning algorithms as well. We only mention, that these libraries and interface routines allow to use the system CPU for training (checking, testing, etc..) purposes as well, as the MiND board via its driver for training process acceleration. Specifics of second kind operation are, that the main (time consuming) part of training procedure, iterative processing of input data, is done using SAND chip, and the rest of algorithm (changing current weights and calculating the quality function) is done by system CPU.

The another aim of developing learning libraries and routines was to implement them in ANI non-parametric statistical analysis package [8], developed in Cosmic Ray Division of Yerevan Physics Institute (CRD, YerPhI), in which, different non-parametric statistic methods including Neural Networks are implemented with emphasis of analysis of multivariate data from the High Energy Astroparticle Physics experiments. The most time-consuming Network training will be accelerated by using

hardware realization of NN (MiND board). Such combined soft-hardware system is of vital importance especially for very high dimensionality data.

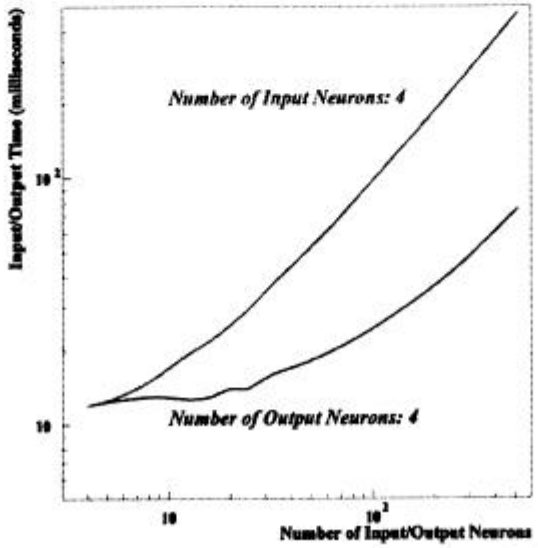

Figure 7: I/O time dependence on the number of input/output neurons for SAND chip

Possibility of Using SAND as the Second Level "Pattern Recognizing Trigger":

The low level trigger (L0) rate from planned Cherenkov Atmospheric Telescopes (ACT) of new Generation as MAGIC [4], Veritas [5] and HESS [6] is dominated by Night Sky Background (NSB) and the Photo-multiplier (PMT) after pulsing and at low thresholds (~ 4 photoelectrons) trigger rate can reach 1 MHz for each channel. For the practical telescope operation the trigger rate must be in the region less than 100 Hz. Therefore, 1 MHz rate has to be reduced down to the few of KHz and tenths of Hz by the higher level triggers (L1 and L2). L0 can be derived from four-fold majority coincidence of all pixels. L1 can be constructed using sophisticated coincidence chime, exploiting the topology of hits pattern requiring close-patched configurations. To obtain short trigger decision times (¡100 nsec) the *Altera 10K* family programmable logic devices can be used ([11, 12]). L1 trigger will provide reduction of the trigger rate down to several units of KHz. For further reduction we propose in ([7]) to use neuro-chip SAND as fast "intelligent" trigger. L2 Pattern Recognizing Trigger (PRT) can help to reject muon and hadron backgrounds which at present is only possible off-line.

In this section we will demonstrate that PRT, designed from MiND boards controlled by the new Linux driver will be able to treat initial pixel information and provide background rejection of Cherenkov images initiated by the isotropic flux of Cosmic Ray (CR) hadrons incident on the telescope aperture.

Unfortunately the realistic MAGIC simulation isn't yet available, so we construct very simple signal (showers initiated by the γ -quanta) and background model. We use 2-dimensional Gaussian distribution incident on the matrix of 20 x 20. In each of the 400 pixels also NSB was simulated by uniform distribution. For the signal "images" the dispersion of the Gaussian was taken less than for the background "images", mimicking in such way longer and broader hadron shower Cherenkov images. So, only shape information of images was used for the discrimination. 1000 simulated events per class were generated and used for net training.

Using such simplified simulation and very big networks we yet want to demonstrate that SAND will give reasonable results for 400 input patterns and very big networks. The MAGIC telescope will work in heavy background conditions. The low energy threshold of the telescope (30 GeV) combined with low pixel threshold will pose very strict requirements on pattern recognition of the signal image.

Table 2: Processing time for single event, with net configuration: 400:64:1, for SAND chip and

SAND/1	Celeron 433MHz
0.97 milliseconds	8.8 milliseconds

Therefore, we can't expect that common technique of pedestal extraction and image reduction to second order moments will provide necessary level of the background rejection. Only usage of all distinctive information contained in the pixels will provide possibility to enlarge signal-to-noise ratio. Working with such huge inputs

(compared with previous analysis of Wipple telescope data, when only 4-5 Hillas parameters were used [9, 10]), new network training algorithms and powerful training accelerators are required.

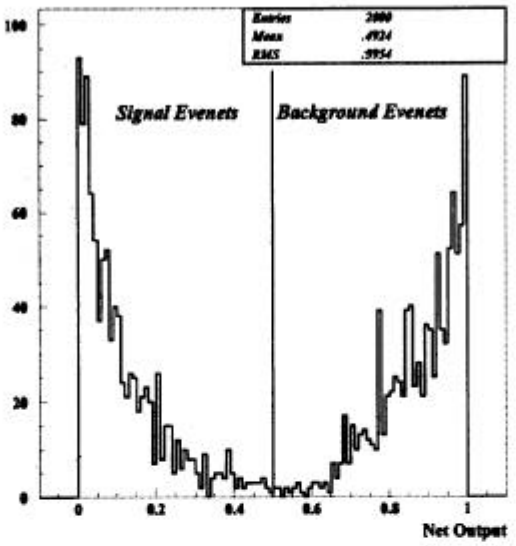

Figure 8: NN output distribution for the 400 MAGIC matrix (left - "signal" events, right -

The NN with 400 inputs provides 98% of correct classifications, as it is shown in Figure 8. proving adequateness of used net training methods for such huge amount of network weights. Of course more realistic simulations are necessary for definite conclusions and performance estimates. 108000 iterations were done within 6 hours. The Celeron 433 MHz CPU for this amount of iterations elapsed 6 days. The comparison of the timing of SAND and Celeron both using 400:64:1 network, is presented Figure 6. Table 2 displays information on the time elapsed for treating of the single event. The required level of the rejection have to be tested with MAGIC simulations, now under way.

References:

1. Fischer, T., Eppler, W., Gemmeke, H. , Kock, G., Becher, T., *The SAND Neurochip and its Embedding in the MiND System*
2. Fischer, T., Eppler, W., Gemmeke, H. , Menchikov, A., *Novel Digital Neural Hardware For Trigger Applications IN Particle Physics*, Proc. of 2nd Workshop on Electronics for LHC Experiments, Balatonfured, Hungary, (1996).
3. Gemmeke, H. , Eppler, W., Fischer, T., Menchikov, A., Neusser, S., *Neural Network Chip for Trigger Purposes in High Energy Physics*, NSS, (1996).
4. Martinez, M., for the MAGIC Collaboration, *The MAGIC Telescope Project*, Proc. 26th ICRC, Salt Lake City, 5, (1999), 219.

5. Bradbury, S., for the VERITAS Collaboration, *The Very Energetic Radiation Imaging Telescope Array System*, Proc. 26th ICRC, 5, Salt Lake City, (1999), 280 .
6. Kohnle, A., for the HESS Collaboration, *The High Energy Stereoscopic System*, Proc. 26th ICRC, Salt Lake City, 5, (1999), 239.
7. *The MAGIC Collaboration, The MAGIC Telescope*, Design study, MPI - PnE/98-5, (1998).
8. Chilingaryan, A., *ANI reference Manual*, version 98.5, unpublished.
9. Chilingaryan, A.A., *Neural classification technique for background rejection in high energy physics experiments*, Neuro-computing, vol. 6, (1994), 497.
10. Chilingaryan, A.A., *Detection of weak signals against background using neural network classifiers*, Pattern Recognition Letters, vol. 16, (1995), 333.
11. Bradbury, S.M., et al, *In Towards a Major Atmospheric Cherenkov Detector - 5* (Kruger Park), ed. O.C. de Jager, (1997), 345.
12. Gemmeke, H., Grinder, A., Keim, H., Kleifges, M., Kunka, N., Szadkowski, Z., Tscherniakowski, D., *Design of the Trigger System for the Fluorescence Detector of the Auger experiment*, In Proc. of Real Time Conference, May Santa Fe, Us, (1999).