International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:5, No:10, 2011

Prediction of Location of High Energy Shower Cores using Artificial Neural Networks

Gitanjali Devi, Kandarpa Kumar Sarma, Pranayee Datta, and Anjana Kakoti Mahanta

Abstract—Artificial Neural Network (ANN)s can be modeled for High Energy Particle analysis with special emphasis on shower core location. The work describes the use of an ANN based system which has been configured to predict locations of cores of showers in the range $10^{10.5}$ to $10^{20.5}$ eV. The system receives density values as inputs and generates coordinates of shower events recorded for values captured by 20 core positions and 80 detectors in an area of 100 meters. Twenty ANNs are trained for the purpose and the positions of shower events optimized by using cooperative ANN learning. The results derived with variations of input upto 50% show success rates in the range of 90s.

Keywords—EAS, Shower, Core, ANN, Location.

I. Introduction

Study of High Energy Particle Showers involve a plethora of theoretical and experimental works comprising of complex measurement and detection equipments. Cosmic showers are the generators of Extensive Air Showers. A primary cosmic ray produces many secondary particles called air showers. When millions or billions of these particles approach the surface of the earth or even a mountain than it is called an extensive air shower (EAS) [1]. Study of shower characteristics involve analysis of measuring the position, size and time extent of the events. Experimental density values maybe used to calculate the shower sizes and location of events which involves tedious theoretical work. Also there are several constraints related to experimental works related to the analysis of EAS. Some of them are due to inaccurate knowledge on interactions of shower particles and primary energies [2]. Therefore, there always exist a necessity to develop a readily available system which can be used to predict locations of shower events.

Several works exist which have used different approaches to analyze extensive air shower (EAS)s and thereby develop applications suitable for EAS shower size prediction and location. A work by D. Hanna [3] reports application of Artificial Neural Network (ANN)s for EAS. Another work by J C Perrett and J T P M van Stekelenborg [4] describes the implementation of an ANN to estimate the core position and energy of extensive air showers recorded by the South Pole air shower experiment (SPASE) [5]. Another work of similar nature is [8]. This work discusses the possibilities of using ANNs for individual EAS data evaluation. A work as cited in [6] uses ANN for providing a mass likelihood distribution for each measured shower, based on its multi-parameter training with simulated showers. Another work by A. Chilingarian . et. al [7] neural network models to recognize the experimental EAS without known primary and energy.

The present work discusses the formulation and working of an ANN based system for prediction of EAS event location in the range $10^{10.5}$ to $10^{20.5}$ eV. The system uses twenty multi layer perceptron (MLP) - a class of feed forward ANN trained with error back - propagation algorithm to determine shower event coordinates. The twenty MLP cluster is trained in a cooperative configuration to provide optimized results. The set up is trained extensively and tested with data samples simulated resembling experimental conditions.

Gitanjali Devi is with the Deptt. of Physics, Lalit Chandra Bharali College, Guwahati - 781011, Assam, India. e-mail: (gitanjalilcb@gmail.com); Kandarpa Kumar Sarma and Pranayee Datta are with Deptt. of Electronics and Communication Technology, Gauhati University, Guwahati - 781014, Assam, India e-mail: (kandarpaks@gmail.com)

Anjana Kakoti Mahanta is with Deptt. of Computer Science, Gauhati University, Guwahati -781014, Assam, India.

II. Basic considerations of the ANN

Artificial Neural Network (ANN)s are non- parametric prediction tools that can be used for a host of pattern classification problems [9] including face recognition. The application of the ANN considers two aspects. A MLP is constituted for this work with one hidden layer and input and output layers. The choice of the length of the hidden layers have been fixed by not following any definite reasoning but by using trial and error method. For this case several sizes of the hidden layer have been considered. Table I shows the performance obtained during training by varying the size of the hidden layer.

TABLE I
Performance variation after 1000 epochs during training of a MLP with variation of size of the hidden layer

Case	Size of hidden layer (x input layer)	MSE Attained	Precision attained in %
1	0.75	1.2 x 10^{-3}	87.1
2	1.0	0.56 x 10^{-3}	87.8
3	1.25	0.8 x 10^{-4}	87.1
4	1.5	0.3 x 10^{-4}	90.1
5	1.75	0.6 x 10^{-4}	89.2
6	2	0.7 x 10^{-4}	89.8

The case where the size of the hidden layer taken to be 1.5 times to that of the input layer is found to be computationally efficient. Its MSE convergence rate and learning ability is found to be superior to the rest of the cases. Hence, the size of the hidden layer of the ANNs considered is 1.5 times to that of the input layer.

The selection of the activation functions of the input, hidden and output layers plays an important part in the performance of the system. A common practice can be to use a similar type of activation function in all layers. But certain combinations

and alterations of activation function types carried out during training provide a way to attain better performance. Two types of MLP configurations are considered- the first type constituted by a set of similar activation functions in all layers and the other with a varied combination of activation functions in different layers. Both these two configurations are trained with gradient descend with variable learning rate and momentum back propagation (GDMALBP) algorithm as a measure of training performance standardization. The outcome

TABLE II
EFFECT ON AVERAGE MSE CONVERGENCE AFTER 1000 EPOCHS WITH VARIATION OF ACTIVATION FUNCTIONS AT INPUT, HIDDEN AND OUTPUT LAYERS

Case	Input layer	Hidden Layer	Output Layer	MSE x 10^{-4}
1	log-sigmoid	log-sigmoid	log-sigmoid	1.45
2	tan-sigmoid	tan-sigmoid	tan-sigmoid	1.32
3	tan-sigmoid	log-sigmoid	tan-sigmoid	1.05
4	log-sigmoid	tan-sigmoid	log-sigmoid	1.02
5	log-sigmoid	log-sigmoid	tan-sigmoid	1.15
6	log-sigmoid	tan-sigmoid	log-sigmoid	1.19

of the MLP blocks vary depending upon the number of training sessions and the data used. Mean Square Error (MSE) convergence and prediction precision are used to ascertain the performance of the MLP blocks. Samples used for training includes images with several types of noise between 1 to 30 dB.

A. Multi Layered Perceptron Based Learning

The fundamental unit of the neural networks is the McCulloch-Pitts Neuron (1943). The MLP is the product of several researchers: Frank Rosenblatt (1958), H. D. Block (1962) and M. L. Minsky with S. A. Papart (1988). Backpropagation, the training algorithm , was discovered independently by several researchers (Rumelhart et. al.(1986) and also McClelland and Rumelhart (1988)).
A simple perceptron is a single McCulloch-Pitts neuron trained by the perceptron algorithm is given as:

$$O_x = g(([w].[x] + b) \tag{1}$$

where [x] is the input vector, [w] is the associated weight vector, b is a bias value and g(x) is the activation function. Such a setup, namely the perceptron will be able to classify only linearly separable data. A MLP, in contrast, consists of several layers of neurons. The equation for output in a MLP with one hidden layer is given as:

$$O_x = \sum_{i=1}^{N} \beta_i g[w]_i.[x] + b_i \tag{2}$$

where β_i is the weight value between the i^{th} hidden neuron. Such a set-up maybe depicted as in Figure 1. The process of adjusting the weights and biases of a perceptron or MLP is known as *training*. The perceptron algorithm (for training simple perceptrons consists of comparing the output of the perceptron with an associated target value. The most common training algorithm for MLPs is error backpropagation. This algorithm entails a back propagation of the error correction through each neuron in the network.

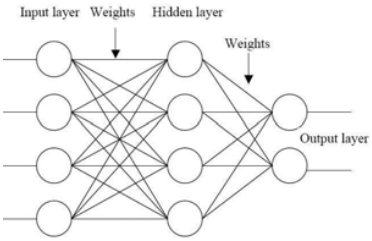

Fig. 1. Multi Layer Perceptron

B. Application of Error Back Propagation for MLP training

The MLP is trained using (error) Back Propagation (BP) depending upon which the connecting weights between the layers are updated. This adaptive updating of the MLP is continued till the performance goal is met.Training the MLP is done in two broad passes -one a forward pass and the other a backward calculation with error determination and connecting weight updating in between. Batch training method is adopted as it accelerates the speed of training and the rate of convergence of the MSE to the desired value [9]. The steps are as below:

- **Initialization**: Initialize weight matrix **W** with random values between [-1,1] if a tan-sigmoid function is used as an activation function and between [0, 1] if log-sigmoid function is used as activation function. **W** is a matrix of CxP where P is the length of the feature vector used for each of the C classes.
- **Presentation of training samples**: Input is $p_m = [p_{m1}, p_{m2}.....p_{mL}]$.
  The desired output is d_m=$[d_{m1}, d_{m2}.....d_{mL}]$.
  - Compute the values of the hidden nodes as:

$$net^h_{mj} = \sum_{i=1}^{L} w^h_{ji} p^{mi} + \emptyset^h_j \tag{3}$$

  - Calculate the output from the hidden layer as

$$o^h_{mj} = f^h_j(net^h_{mj}) \tag{4}$$

    where f(x)=$\frac{1}{e^x}$
    or f(x)=$\frac{e^x - e^{-x}}{e^x + e^{-x}}$
    depending upon the choice of the activation function.
  - Calculate the values of the output node as:

$$o^o_{mk} = f^o_k(net^o_{mj}) \tag{5}$$

- **Forward Computation**: Compute the errors:

$$e_{jn} = d_{jn} - o_{jn} \tag{6}$$

  Calculate the mean square error(MSE) as :

$$MSE = \frac{\sum_{j=1}^{M} \sum_{n=1}^{L} e^2_{jn}}{2M} \tag{7}$$

  Error terms for the output layer is:

$$\delta^o_{mk} = o^o_{mk}(1 - o^o_{mk}) e_{mn} \tag{8}$$

Error terms for the hidden layer:

$$\delta^h_{mk} = o^h_{mk}(1 - o^h_{mk}) \sum_j \delta^o_{mj} w^o_{jk} \quad (9)$$

- **Weight Update**:
  - Between the output and hidden layers

$$w^o_{kj}(t+1) = w^o_{kj}(t) + \eta \delta^o_{mk} o_{mj} \quad (10)$$

where η is the learning rate($0 ¡ \eta ¡ 1$). For faster convergence a momentum term(α)maybe added as:

$$w^o_{kj}(t+1) = w^o_{kj}(t) + \eta \delta^o_{mk} o_{mj} + \alpha(w^o_{kj}(t+1) - w_{kj}) \quad (11)$$

  - Between the hidden layer and input layer:

$$w^h_{ji}(t+1) = w^h_{ji}(t) + \eta \delta^h_{mj} p_i \quad (12)$$

A momentum term maybe added as:

$$w^h_{ji}(t+1) = w^h_{ji}(t) + \eta \delta^h_{mj} p_i + \alpha(w^o_{ji}(t+1) - w_{ji} \quad (13)$$

One cycle through the complete training set forms one epoch. The above is repeated till MSE meets the performance criteria. While repeating the above the number of epoch elapsed is counted. A few methods used for MLP training includes:

- Gradient Descent(GDBP)
- Gradient Descent with Momentum BP (GDMBP)
- Gradient Descent with Adaptive Learning Rate BP (GDALRBP) and
- Gradient Descent with Adaptive Learning Rate and Momentum BP (GDALMBP).

Here, while MSE approaches the convergence goal, training of the MLPs suffer if:
$Corr(p_{m,i}(j), p_{m,i}(j+1))$ = high and
$Corr(p_{m,i}(j), p_{m,i+1}(j))$= high. This is due to the requirements of the (error) back propagation algorithm as suggested by equations 8 and 9.

III. Application of ANN for Core Location Prediction

Showers were generated according to a modified NKG function [3] with particle content between $10^{10.5}$ to $10^{20.5}$ eV with Moliere radius of 70 m. Their cores were evenly distributed within a circle of radius 50 m centered on the middle of the array and the detectors distributed within a 100 meter arc. This restriction was adopted to avoid edge effect. A conceptual model of the core and detector locations used for the work are depicted in Figure 2. The high energy showers between $10^{10.5}$ to $10^{20.5}$ eV are simulated and density values calculated. While calculating density values core positions and locations of detectors are important. These coordinates are used to simulate the density values. The work considers twenty shower events taking place within a radius of fifty meters. The ANN is designed to accept density values obtained for twenty showers and provide coordinates of the EAS events of which the measurements are made. The density values captured by the detectors and coordinates of the shower events are unique hence require separate ANN predictors for each of the

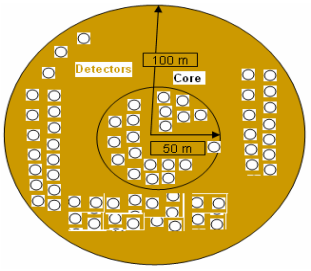

Fig. 2. Conceptual set-up used for simulation of density functions of EAS

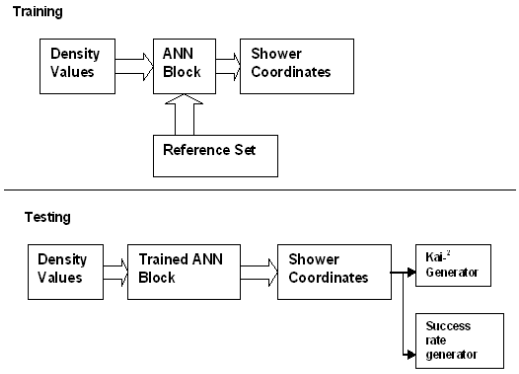

Fig. 3. Experimental Set-up

positions. For twenty shower events similar number of ANN are formed and a lay-out akin to the committee machine [9] is formulated. The set-up is essential for accurate prediction and optimization of the results. The training is carried out in a cooperative environment as to minimize the predicted error. The fundamental considerations governing the working and parameter selection of the cooperative ANNs or committee machines can be explained using the following analysis [10] [9]:
Let a training set of m input - output pairs be $(x^1, t_1), (x^2, t_2), ...(x^m, t_m)$ be given and N networks are trained using this set of data. For simplicity, let for n-dimensional input there be a single output. Let for network functions f_i for a number of networks represented by indices $i = 1, 2, ...N$, the cooperative or committee network formed generates as output given as

$$f = \frac{1}{N} \sum_{i=1}^{N} f_i \quad (14)$$

The rationale behind the use of the averaging in the output of the cooperative or committee network as given by eq. 14 is the fact that if one of the constituent networks in the ensemble is biased to some part of the input samples, the ensemble average can scale down the prediction error considerably [10].

A quadratic error function can be computed from each of the error vectors e_i using the ensemble function f as

$$Q = \sum_{i=1}^{m}[t_i - \frac{1}{N}\sum_{i=1}^{N} f_i]^2 \quad (15)$$

Using matrix notation, the quadratic error can be expressed as

$$Q = |\frac{1}{N}(1,1,....)E|^2 = \frac{1}{N^2}(1,1,....1)EE^T(1,1,.....1)^T \quad (16)$$

EE^T is the correlation matrix representing the error residuals. If each function approximation produces uncorrelated error vectors, the matrix EE^T is diagonal and the i^{th} diagonal element Q_i is the sum of quadratic deviations for each functional approximation, i.e $Q_i = \|e^i\|^2$. Thus,

$$Q = \frac{1}{N}(\frac{1}{N}(Q_1 + Q_2 + ... + Q_N)) \quad (17)$$

It implies that the total quadratic error of the ensemble is less by a factor $\frac{1}{N}$ than the average of the quadratic errors of the total computed approximations. This holds only if N is not very large. If the quadratic errors are not uncorrelated, i.e if EE^T is not symmetric, a weighted combination of N functions f_i can be approximated as

$$f = \sum_{i=1}^{N} w_i f_i \quad (18)$$

The weights w_i must be computed in such a way as to minimize the expected quadratic deviation of the function f for the given training set. With the constraint with the constraint $w_1 + + w_N = 1$, eq. 16 transforms to

$$Q = \frac{1}{N^2}(w_1, w_2, ...w_N)EE^T(w_1, w_2, ...w_N)^T \quad (19)$$

Differentiating the above eq. 19 w. r. t $w_1, , w_N$, and using a Lagrangian multiplier λ for the constraint $w_1 + + w_N = 1$, the above functional modifies to

$$Q' = \frac{1}{N^2}wEE^T + \lambda(1,1,.....1)w^T \quad (20)$$

$$= \frac{1}{N^2}wEE^T + \lambda 1 w^T \quad (21)$$

where 1 is a row vector with all its N components equal to 1. Setting the partial derivative of Q' with respect to w to zero this leads to

$$\frac{1}{N^2}wEE^T + \lambda 1 = 0 \quad (22)$$

With simplification,

$$\lambda = \frac{1}{N^2 1(EE^T)^{-1}1^T} \quad (23)$$

The optimal weight set can be calculated as

$$w = \frac{1(EE^T)^{-1}}{1(EE^T)^{-1}1^T} \quad (24)$$

assuming that the denominator does not vanish. This method, however, is dependent on the constraint that EE^T is not ill-conditioned.

IV. Experiential Results and Discussion

Each of the twenty units of the ANN cluster is formed by cascade feed-forward networks - a variation of the MLP trained with back-propagation. The average data size for each of the block is fifty sets of 20×100 where 20 represents the number of shower cores and 100 denotes the density values recorded by the detectors. Noise between -3 dB and 3 dB are mixed to make the ANN cluster robust enough to deal with variations found from experiential works.

Figure 4 shows the location of shower events as predicted by the ANN blocks with detector positions and core positions shown. The plot is for one event of which the density values are fed to the trained ANN set-up. After training with fifty sets of data the plotted values are generated as the average of twenty sets of inputs of which half are with noise variation in the mentioned range. The results show a success rate of around 95%. The above is repeated for another event and a similar success rate is obtained. A set of results are shown in Table III summarizing the training and testing processing. The location of all the twenty showers has also been generated

TABLE III
Average performance derived during training of the ANN cluster for one shower event

SL Num	Epochs	Success rate in %	Time in sec.s
1	5000	94.1	98.3
2	10000	95.3	186.3
3	15000	96.2	318.2
4	20000	96.3	473.5

using the ANN cluster as a whole. Initially as the training is limited to a few thousand session, the event cluster is spread insider and outside the fifty meter radius. The expected results are a grouping inside the fifty meter arc. As training sessions are increased with more number of samples, the predicted results start to cluster inside the intended circle. Figure 6 show a grouping generated by the ANN - cluster after 5000 sessions of training. The grouping clearly shows the location of shower events generated using density values placed inside the circle. A better clustering of the events recorded after 10000 iterations is shown by Figure 7. The number of training sessions have been extended to 20000 also but the best results are obtained around the 10,000 to 12,000 mark. hence, testing results are derived from the ANN cluster trained upto this limit. A unitary ANN block instead of a cluster can also be sued for the purpose but prediction results are atleast 5% below than that generated by the cluster. Moreover, the cluster with its optimization capacity provides the best result out of a sample set of twenty applied to it. This is an advantage provided by the cluster. Moreover, as the shower core has its own unique location and density values, a unitary block shows best discrimination capacity only when applied for a single shower core location prediction. A plot of the success rates of the core location prediction is depicted by Figure 8. The training doesn't improve much after 12000 sessions but training time increases.

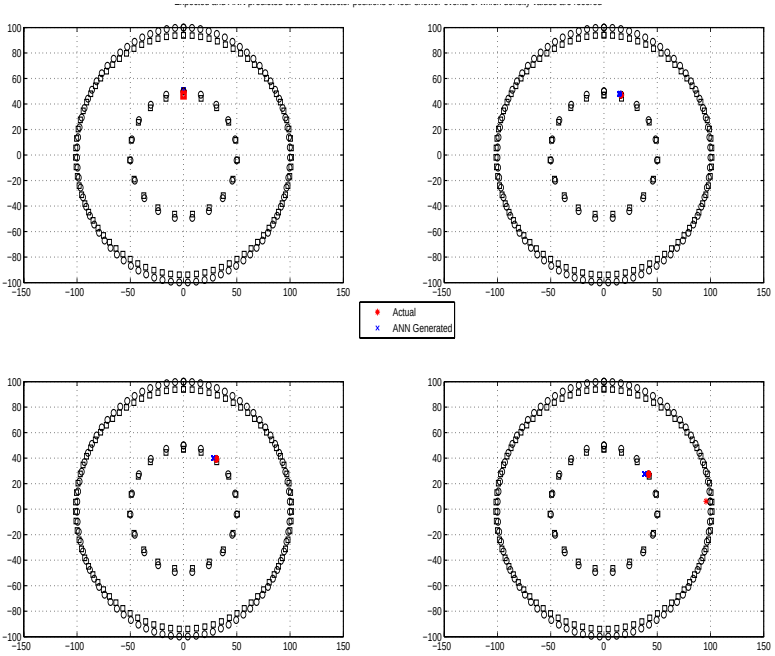

Fig. 4. Location of shower events with detector and core positions generated by the ANN set-up for one event

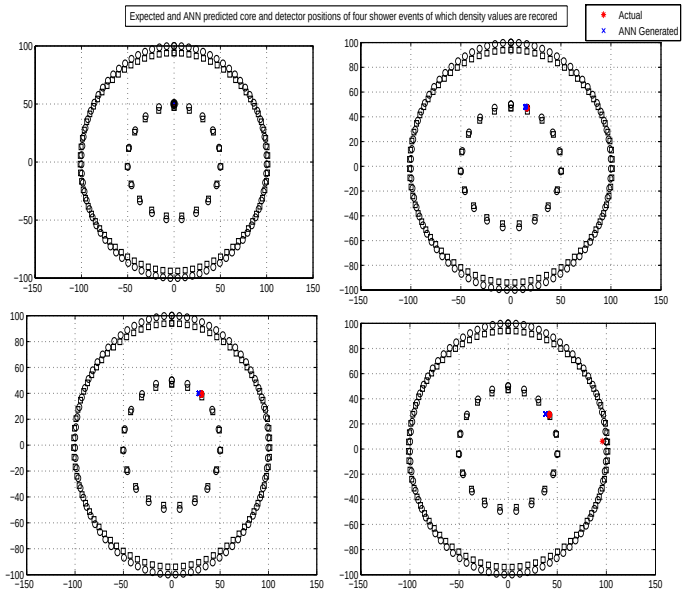

Fig. 5. Location of shower events with detector and core positions generated by the ANN set-up for another event

V. Conclusion

The work is an attempt to use ANN based techniques to predict core locations. The experiential work carried out with a ANN cluster is found to be better suited for handling core location prediction and optimization. Shower event data are simulated and the extracted data are used for ANN training which is tested further to verify the extent of training. The system thus developed is a readily available tool which can provide nearly precise location details of shower in the range $10^{10.5}$ to $10^{20.5}$ eV. The system has the potential to become a part of an experiential set-up for which it needs to be extended and modified further for real time applications.

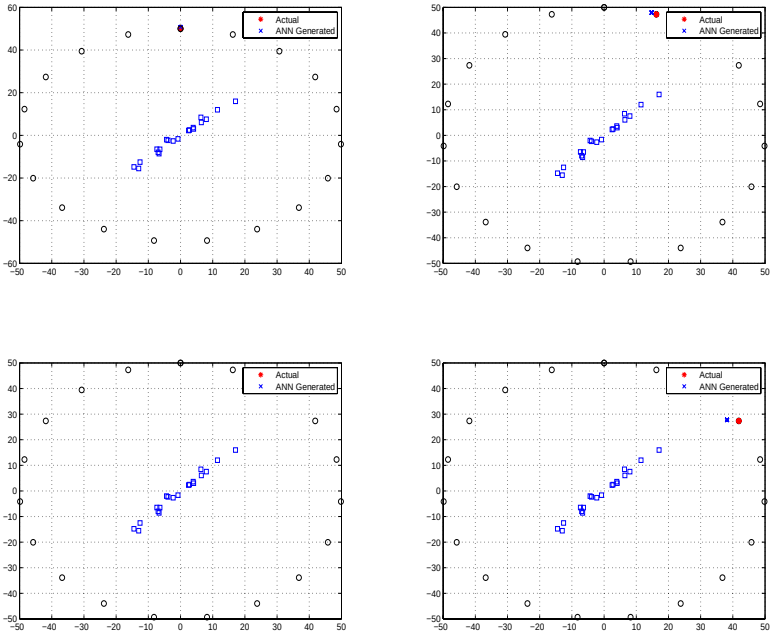

Fig. 6. Shower events of four cases predicted by ANN after 5000 sessions taking density values from 100 detectors

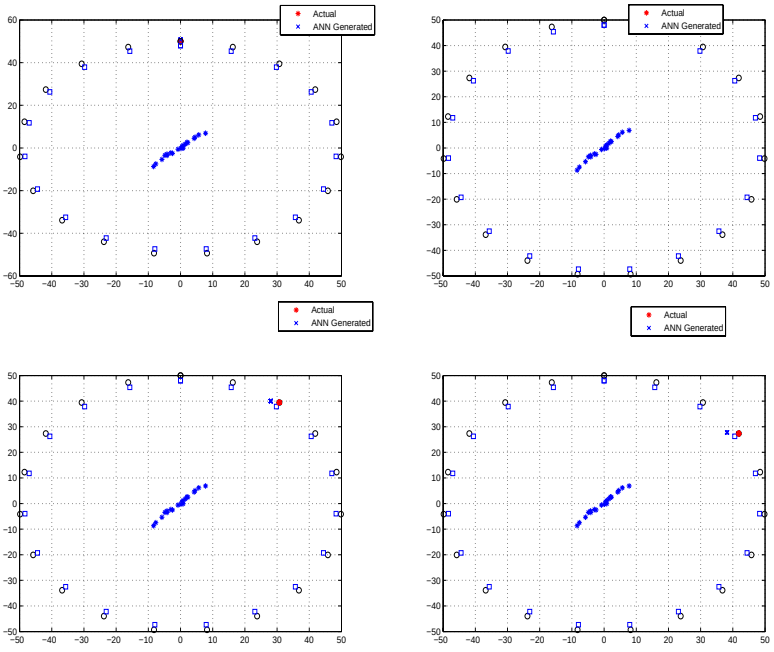

Fig. 7. Shower events of four cases predicted by ANN after 10000 sessions taking density values from 100 detectors

REFERENCES

[1] R. A. Mewaldt: "Cosmic Rays", Macmillan Encyclopedia of Physics- available at http://www.srl.caltech.edu/personnel/dick/cos-encyc.html, 1996.

[2] R. Engel: "Theory and Phenomenology of Extensive Air Showers", Forschungszentrum Karlsruhe, Germany- available at http://moriond.in2p3.fr/J05/trans/sunday/engel1.pdf, 2005.

[3] D. Hanna, "Application of Neural Nets to Extensive Air-Showers", Proceedings of the 22nd International Cosmic Ray Conference, IUPAP, Volume 4, p: 500, 1991.

[4] J. C. Perrett and J T P M van Stekelenborg, "The applications of neural networks in the core location analysis of extensive air showers", Bartol Research Institute, University of Delaware, 217 Sharp Laboratory, Newark, DE 19716, USA J. Phys. G: Nucl. Pert. Phys. 17, pp:1291-1302, 1991.

[5] H. J. Mayer, "A neural network algorithm for core location analysis at large extended air shower arrays", Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and

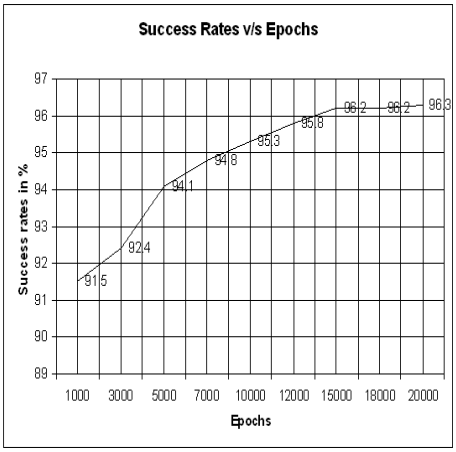

Fig. 8. Success rates generated by the ANN cluster predictor during 1000 to 20000 training sessions

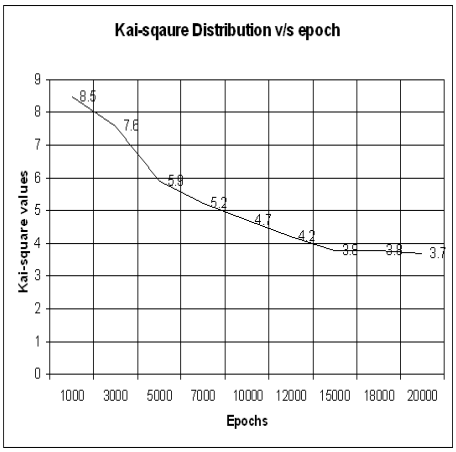

Fig. 9. Kai^2 distribution generated by the ANN cluster predictor during 1000 to 20000 training sessions

Associated Equipment, Volume 317, Issues 1-2, pp. 339-345, June, 1992.
[6] P. Sommers: "Extensive Air Showers and Measurement Techniques", Physics - Astrophysics, C. R. Acad. Sci. Paris, t. 4, Serie IV, pp. 1 - 12, 2004.
[7] A. Chilingarian, G. Gharagyozyan, S. Ghazaryan, G. Hovsepyan, E. Mamidjanyan, L. Melkumyan, V. Romakhin, A. Vardanyan and S. Sokhoyan: "Study of extensive air showers and primary energy spectra by MAKET-ANI detector on mountain Aragats", Elsavier Journal of Astroparticle Physics, vol. 28, pp. 5871, 2007.
[8] T. Wibig, "The Artificial Neural Networks as a tool for analysis of the individual Extensive Air Showers data", Preprint submitted to Elsevier Preprint (1 February 2008), available at http://arxiv.org/abs/hep-ph/9608227v1.
[9] S. Haykin: *Neural Networks A Comprehensive Foundation*, 2^{nd} ed., Pearson Education, New Delhi, India, 2003.
[10] R. Rojas, *Neural Networks A Systematic Introduction*, Springer, Berlin, 1996.