Fermi National Accelerator Laboratory

FERMILAB-Conf-90/68

Tests of Track Segment and Vertex Finding with Neural Networks*

Bruce Denby and Eliane Lessner
Fermi National Accelerator Laboratory
P.O. Box 500
Batavia, Illinois 60510

Clark S. Lindsey
Iowa State University
Ames, Iowa 50011

April 1990

* Talk presented by C. S. Lindsey at the 1990 Conference on Computing in High Energy Physics, Santa Fe, New Mexico, April 9-13, 1990.

Operated by Universities Research Association Inc. under contract with the United States Department of Energy

TESTS OF TRACK SEGMENT AND VERTEX FINDING WITH NEURAL NETWORKS

Bruce Denby, Eliane Lessner*
Fermilab**, Batavia Il. 60510, USA

Clark S. Lindsey***
Iowa State University, Ames Ia. 50011, USA

ABSTRACT

Feed forward neural networks have been trained, using back-propagation, to find the slopes of simulated track segments in a straw chamber and to find the vertex of tracks from both simulated and real events in a more conventional drift chamber geometry. Network architectures, training, and performance are presented.

INTRODUCTION

Fast pattern recognition is desirable in high energy physics for two reasons. Firstly, in many high energy experiments, events are produced at very high rates (e.g. 100Mhz expected at the SSC), and the triggering systems, which accept or reject events upon the basis of characteristic patterns in the data, must operate at these rates. Secondly, even though trigger rejection factors are large, the final data samples can still be extremely large, and the more detailed 'offline' pattern recognition performed on these data can lead to processing times, using conventional techniques, of several years.

Neural networks have been proposed for a wide variety of pattern recognition applications, such as analysis of sonar returns [1] and automatic target recognition [2]. Neural networks are particularly suited to these applications due to their inherent parallelism and potential realization in hardware, which should allow solutions to pattern recognition problems 'in real time'.

A number of papers have suggested application of neural networks to specific problems in high energy physics, such as track reconstruction [3], electron identification [4], B-jet recognition [5], secondary vertex finding [5]. Here we discuss the application of feed-forward neural nets to the problems of quickly finding slopes of stiff track segments and locations of production vertices, starting from hits in drift chambers. The results presented here are encouraging but incomplete. More work will be necessary to realistically address the issues of noise and track crowding, to examine hardware implementations for specific cases, and to optimize the network architecture.

TRACK SEGMENT FINDING

'Track dictionary' techniques, in which hit patterns for all possible tracks are stored in a memory, have been used, both in software [6] and hardware [7] implementations, as a quick way of transforming from hits to tracks. A disadvantage of this method is that in large, high resolution systems, the number of possible tracks can be very large. A more compact representation can be had by representing each hit as an analog quantity, the drift time, associated with a wire number.

Feed-forward neural networks are able to perform arbitrary mappings of analog quantities from one representation to another [8], and so should be able to perform the mapping from the hits to the parameters of the tracks that produced them, e.g., the slope and/or intercept of track segments.

Talk presented by Clark S. Lindsey at the "Conference on Computing in High Energy Physics", Sante Fe, New Mexico, April 8-13, 1990.

TRACK SLOPES IN STRAW CHAMBERS

A section of a straw tube drift chamber is shown in fig. 1. It consists of a hexagonal close packed array of cylindrical straws, each enclosing an active drift volume with a sense wire at the center. A particle which traverses the drift volume of a particular straw will produce at its sense wire a signal delayed by a time proportional to the distance of closest approach of the particle to the wire. The data for a single track will be a set of times ranging from tmax, the maximum drift time, to tmin, for those wires touched by the track. These analog data are used as input to the neural network. Note that the net must solve the left-right ambiguity itself using the half cell shift of alternate layers.

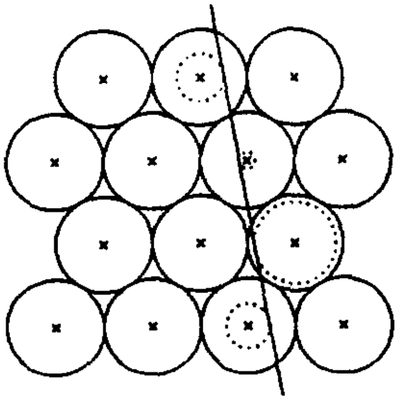

Fig.1. Section of straw tube drift chamber with track. Dots indicate drift distances from sense wires (x's).

The network architecture is shown in fig. 2. This is a feed-forward network architecture (see ref. [9] for a detailed discussion of this architecture). The input units correspond to the 14 straws of fig. 1. Twenty-five hidden units were used. The track angle was represented in the 14 output units by a Gaussian histogram of r.m.s. one bin (6 degrees) and mean equal to the angle.

At the input of each neuron, the sum is formed of the outputs of neurons in the preceding layer, each multiplied by a weight. The output of the neuron is a non-linear, sigmoidal function (e.g. tanh) of this sum.

Initially, a training set was made of 500 'events', each with a single straight track randomly distributed in angle between ±30 degrees from the vertical and with random horizontal offsets. Valid training events were required to pass through at least 4 straws to avoid edge effects. Each training event consisted of 14 input times and 14 target values which were zero except for three contiguous bins which formed a Gaussian histogram of r.m.s. one bin and with a mean equal to the angle of the training track (fig. 2). After a training session, the performance of the net was tested using a new independent set of 500 random tracks. A plot was made of the difference between the angle found by the network and the true angle of the track. After 2 million presentations, the r.m.s. deviation was about 1 degree but the distribution had large non-Gaussian tails. These were found to be pathological cases in which the track just grazed the edges of the straws and ambiguous cases due to the left-right ambiguities of the straws.

The training sample was increased from 500 to 5000 events to allow the network to see a wider variety of tracks. The result after an additional one million presentations on the larger training set (tested on an independent set of 5000 events) is shown in fig. 3. The central Gaussian has a sigma of 0.5 degrees. The net no longer makes very bad mistakes, but tails are still present.

No noise or drift smearing was used in this simulation. (These are introduced in the following sections on vertex finding). For a perfectly trained network, the network angle should agree perfectly with the true angle, except for those cases in which the left-right ambiguity causes the track angle to be ambiguous. The relative contributions to the observed width from ambiguities and inadequacies in training or network architecture are not known.

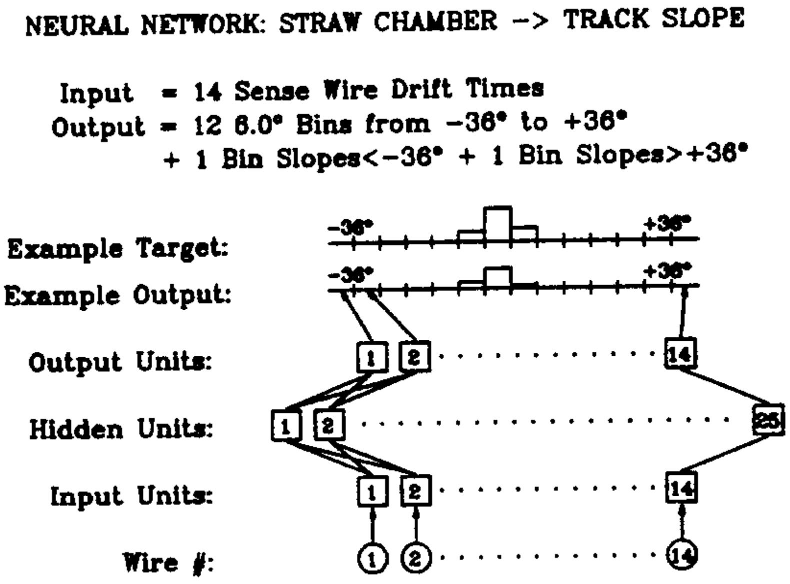
NEURAL NETWORK: STRAW CHAMBER -> TRACK SLOPE
Input = 14 Sense Wire Drift Times
Output = 12 6.0° Bins from -36° to +36°
+ 1 Bin Slopes<-36° + 1 Bin Slopes>+36°
Example Target:
-36°
+36°
Example Output:
-36°
+36°
Output Units:
1
2
14
Hidden Units:
1
2
25
Input Units:
1
2
14
Wire #:
1
2
14

Fig. 2. Neural network to find straw chamber track slopes with drift times as input. The output units (values 0.0-1.0) correspond to 6° bins in angle. All hidden units are connected to all input and output units. Only a few connections are shown for clarity. Also a bias unit (value fixed to 1.0) is connected to all hidden and output units but not shown.

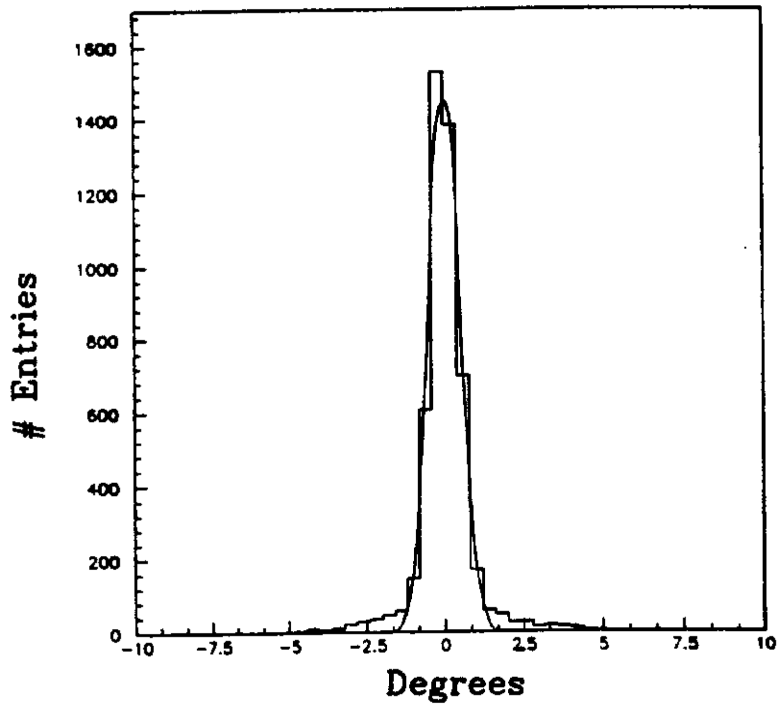

Fig. 3. Differences between target slope and net slope for 5000 independent tracks. Fit gives r.m.s of 0.5 degrees.

PRIMARY VERTEX FROM A 4-LAYER DRIFT CHAMBER

A network can also be trained to find the intercept of a track on a particular axis. Fig. 4 shows simulated events in a more conventional drift chamber design having 4 layers of 5 wires each. The sense wires are spaced 1.1cm apart and are separated by field wires which define the sensitive area for each cell. The drift distance is defined as the distance from the point where a track crosses the sense wire plane to the nearest sense wire. The layers are spaced by 1.1cm, with the bottom layer being 13.1cm above a "beamline" from which simulated tracks originate. Two of the layers are shifted by one-half cell relative to the other two. The goal is to have the net determine, using only the drift distances, the primary vertex on the beamline from which the tracks in a given "event" originate.

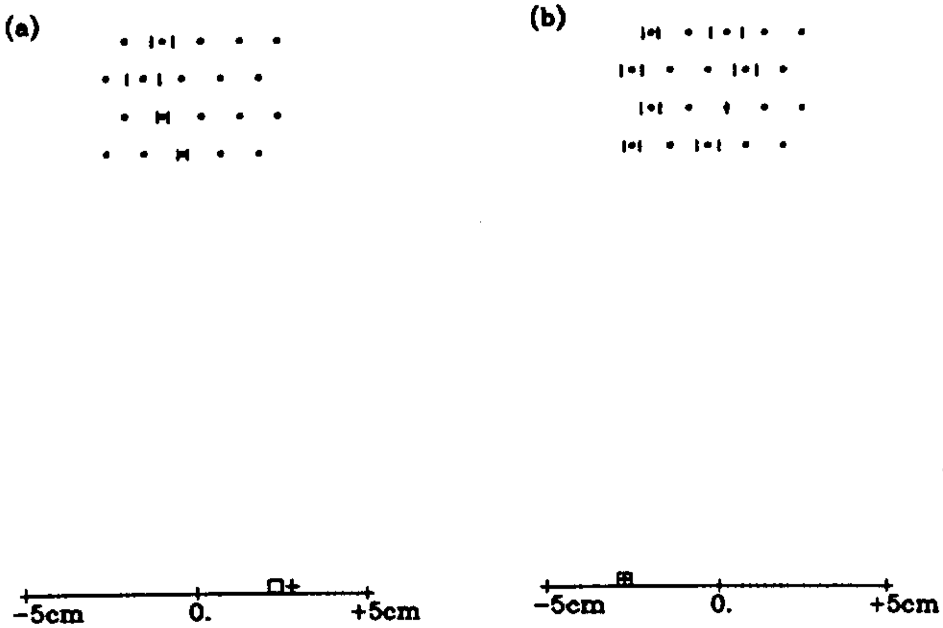

Fig. 4. Sample events in 4-layer drift chamber simulation: (a) 1 track and (b) two track events. Points indicate sense wires, and vertical lines are drift distances to tracks (with left-right ambiguity). Tracks originate from ±5cm range of "beamline". Box is the target vertex and plus sign is the neural net vertex (here the vertex is given as the position of the output bin with largest value, see fig. 2).

The net architecture here is similar to that in fig. 2, except 20 input units represent 20 sense wire cells, with the inputs proportional to the drift distances, and 20 output units correspond to 0.5cm bins along the 10cm range of the beamline where vertices are distributed. Instead of 25 hidden units, 50 were used.

Like the straw chamber case, the net is trained by comparing the net's 20 output values to 20 "target" values for each event. Back-propagation minimizes these differences over a set of 5000 one track plus 5000 two track events (two tracks were not allowed to cross the same cell). A target distribution indicated the vertex positions with Gaussian distributions whose means were the vertex positions and r.m.s. of 3mm. In fig. 4 the target and net vertices are shown as the positions of the bins with the largest values (see also example distributions in fig.2).

After about a million iterations through these events, the net was tested on independent events. The difference between the average of the target distribution and the average of the net's output distribution is shown in fig. 5. A Gaussian fit shows an r.m.s. of about 0.2cm. As with the straw chamber track slopes, there are non-Gaussian tails. These are again believed due to the "resolution" of the net with this number of hidden units and amount of training, as well as ambiguities.

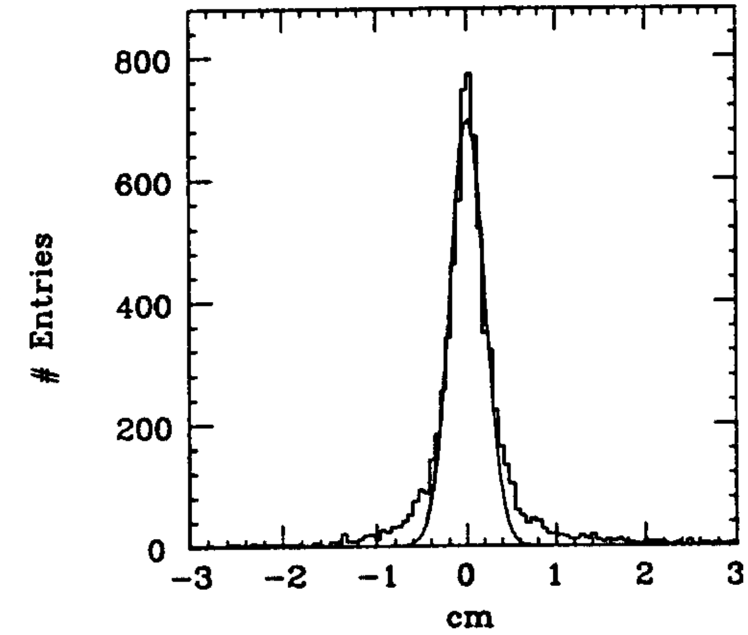

Fig. 5. Differences between target vertex and neural network vertex for an independent set of events. The fit gives r.m.s of 0.2cm.

In the examples shown so far, there was no smearing of the drift distances to simulate a finite resolution. In a test, the vertex resolution was found to worsen roughly linearly with the drift resolution, giving a resolution of 0.3cm with a smearing of 200μm. Even though trained only on non-smeared cases, the net's performance degrades smoothly rather failing dramatically when presented with this new situation.

The effect of noise was also studied by superimposing a random hit on normal 1 and 2 track events in a cell not crossed by a track. After another million iterations of training the accuracy for events without noise remained 0.2cm and with noise decreased to 0.6cm due to the increased ambiguities, especially with 2 tracks.

TRACK VERTEX FROM 3-LAYER DRIFT CHAMBER DATA

To test these techniques with events from an actual drift chamber, data from a chamber used in experiment E-735 [10] at the Tevatron proton-antiproton Collider was obtained. The "Z-Chamber" is a small 3 layer planar chamber with 96 wires per layer, 10cm long wires, and about 1.0m wide [11]. The chamber (see schematic in fig. 6a) sat next to the beampipe with the wires vertical and normal to the beam-line. The cell geometry is similar to that used in the 4-layer simulation above - the wire spacing and layer spacing is 1.1cm and the 1st layer is 13.1cm from the beamline (however, ambiguous cases increase with 3 rather than 4 layers). The chamber was used both to assist in tracking particles entering an adjacent spectrometer and to determine the primary event vertex.

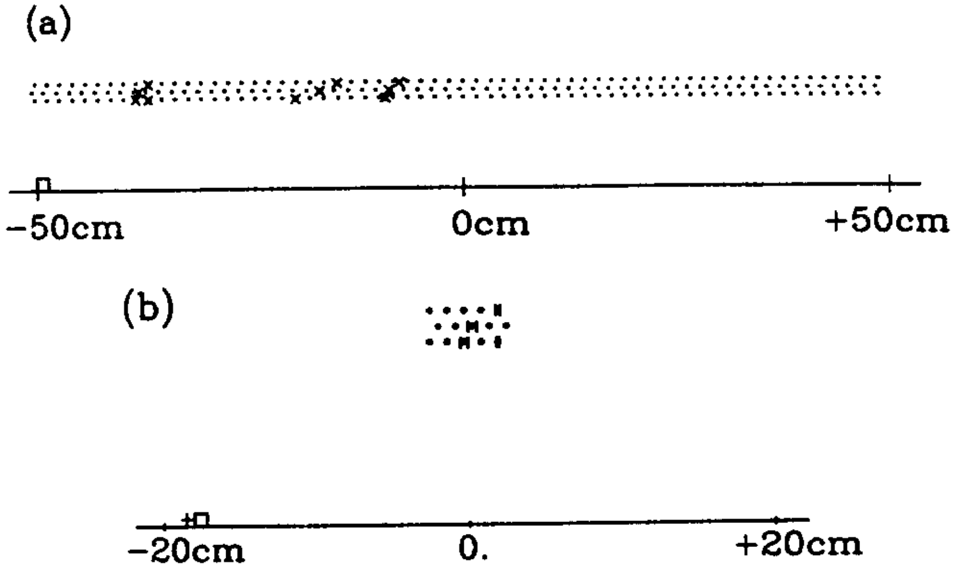

Fig. 6 Tracks in the Z-Chamber from proton-antiproton collision. (a) hits in the 3 layers of the chamber are shown as X's and the points are sense wires. Box indicates the vertex as found by a conventional pattern recognition and fitting routine. (b) The first cluster of hits to the left in (a) have been fed into a net which looks at 15 wire sections and finds the track vertex within ±20cm range from the center of that section. The plus sign is the net vertex.

A net with 3 x 96 input units and enough output units to cover about a 1.5m vertex range is too large for rapid simulation so here we only dealt with a small section of the chamber. Fig. 6b shows a section of the Z-chamber with 5 wires in each layer for a total of 15 wires. (The whole chamber can be dealt with by partitioning it into small subunits, each with its own neural network, whose outputs are in turn superimposed such that the primary event vertex(ices) appear as peaks in a global output histogram. Studies are in progress on primary event vertex finding using this technique).

The network was again similar to fig. 2, except with 15 input units for 15 sense wires, 100 hidden units, and 42 output units, where units 2-41 represent vertices in forty 1cm bins along the beamline. Bin 1 represents all vertices to the left of this range and bin 42 represents all vertices to the right of this range.

Using conventional tracking and pattern recognition methods as discussed in [11], tracks and vertices were found for each event. Although the drift resolution was 100μm for optimum running conditions, it was run at a lower than optimal voltage, so as to extend chamber lifetime, which gave 500μm resolution. This results in about 1cm vertex resolution. Note also that the drift times are for the distance of closest approach to the wire rather than the distance from crossing the wire plane as was done for the 4-layer simulation.

To train the net to find the vertex of a track, a selection was made of events with only one track found by the tracking program. For such events, beginning from one end of the chamber, 5 wire wide sections were examined to find if the 3 hits used in the track fit were within the section. If this wasn't the case then the next section was examined, stepping in 1 wire increments. This stepping through the chamber continued until all 3 track hits were inside the 5 wire wide section.

Fig. 6a shows such an event. Here a track with a background hit is seen in the first cluster of hits on the left. The 5-wire wide section of the chamber containing the cluster is shown in fig. 6b. The target vertex (as found by the tracking program) for this case is translated into the +/-20cm range of the net for the five wire section. The 15 drift distances (from calibrated drift times) and the target position along the 40cm wide range for this event and others like it were saved in a training set of 10000 events. In about 15% of the events there were background noise hits (i.e. hits not used by the fitting program) in the 5-wire wide section.

After a million iterations through the training set, the net was tested on an independent data set. The resulting distribution is shown in fig. 7. The difference between the average target and net output values gives an r.m.s. of about 0.6cm. Here the tails are quite wide and are due to (1) the background noise events where the resolution degrades to 1.5cm; (2) the greater chance of ambiguities with only 3 layers; (3) the 500µm drift resolution; (4) possible inadequacies in the training or number of hidden units.

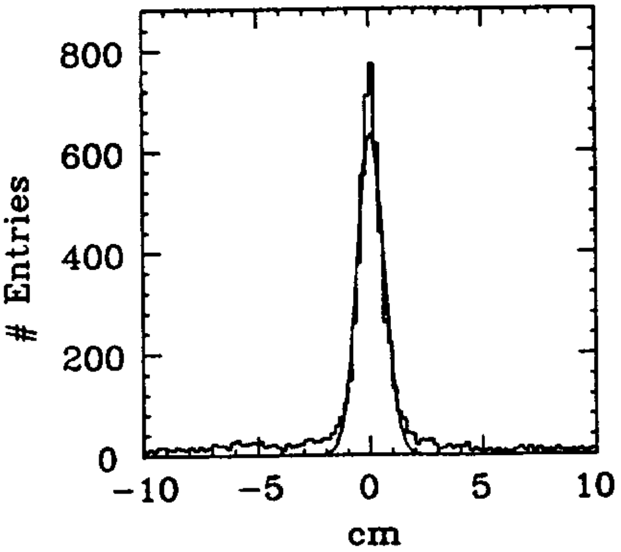

Fig. 7. Difference in fitted track vertex and neural net track vertex for hits in 15 wire sections of the Z-chamber. Fit gives r.m.s. of 0.6cm.

DISCUSSION

We have shown that using drift chamber information, a neural net can: (1) be trained to find the slopes of single tracks; (2) find the vertices of 1 and 2 track events; (3) work with typical drift chamber resolutions; (4) work in presence of noise; (5) be trained to handle new types of input, showing adaptability. The technique was found to work for both simulated and real data from a collider experiment. The latter included such complications as 500µm drift resolution, drift times which were for the distance of closest approach rather than from the horizontal crossing point, presence of background hits, and training sets made from data events so that the "targets" were smeared by the resolution of conventional tracking methods (i.e. the "true" vertex wasn't known but was given by the extrapolated track fit).

The results may already be accurate enough for some applications, especially in triggering. For more demanding applications, such as secondary vertex finding, however, further study must be done to improve the accuracy of the nets. Topics of interest include: (1) understanding non-Gaussian tails to the resolution; (2) improving the performance in presence of noise; (3) studying effects of overlapping tracks; (4) determining the optimum level of hidden units and training. Although back-propagation training has been an area of intense study by the neural networks community, there is still no way known to determine, a priori, the number of hidden units needed for a given application. In the cases here, we chose the number of hidden units to be roughly similar to the total number of input and output units. We found for the vertex finding tests that the accuracy improved by increasing the number of hidden units However, the amount of training time needed to reach a given level of accuracy increased rapidly as well.

The neural network chip ETANN [15] developed by Intel is also being examined. It allows configurations of 64 inputs, 64 hidden and 64 output units. A setup is being prepared to find if this chip can reproduce the results found here with simulations.

ACKNOWLEDGEMENTS

We thank Theo Alexopoulos for Z-Chamber information and also thank Chiho Wang for programming assistance. Thanks also to E. W. Anderson. This research supported in part by DOE Contract DE-AC02-85ER40193 and Fermi National Accelerator Laboratory.

* Present address: Advanced Photon Source, Argonne-Bld 360, 9700 S.Cass Ave., Argonne Il. 60439.

** Operated by the Universities Research Association, Inc under contract with the U.S. Dept. of Energy.

*** Talk presented by C. S. Lindsey. Present address: E-735, MS-219, Fermilab, P.O. Box 500, Batavia Il. 60510.

REFERENCES

1. R. Gorman & T.Sejnowski,"Neural Networks",vol.I,pp.75-89 (1988).
2. M. Oyster, Hughes Aircraft, in "The DARPA Neural Network Study", AFCEA International Press, Fairfax, VA (1988), p.451.
3. B. Denby, "Computer Physics Communications" 49(1988) pp.429-448.
C. Peterson, Nucl. Inst. Meth., A279(1989) pp. 537-545.
4. D. Cutts et al., "The Use of Neural Networks in the D0 Data Acquisition System", presented at the conference REAL-TIME '89, Williamsburg, VA (May 1989), to be published.
5. B. Denby et al., FERMILAB-CONF 90/20, Neural Networks for Triggering, presented at 1989 IEEE Nuclear Science Symposium, San Francisco, CA, to be published in IEEE Trans. Nucl. Science.
6. J.J. Becker et al., Nucl. Inst. Meth. A235 (1985) 502.
7. M. Dell'Orso and L. Ristori, VLSI Structures for Track Finding, proceedings of Int. Conference on the Impact of Digital Microelectronics and Microprocessors on Particle Physics, Trieste, Italy, 28-30 March, 1988, World Sci. Pub. Co.
8. Robert Hecht-Nielsen, Theory of Backpropagation Neural Networks, proc. of the International Joint Conference on Neural Networks, vol I, pp. 593-605, Washington, D.C., 18-22 June, 1989,IEEE Catalog no. 89CCH2765-6.
9. D. Rumelhart et al., Parallel Distributed Processing, Explorations in the Microstructure of Cognition, vol. I, ch. 8, MIT Press, Cambridge, Mass.
10. Bannerjee et al., Nuc. Instr. Meth., Phys. Res. A26 1211(1988).
11. T. Alexopoulos, A. R. Erwin, C. Findeisen, K. Nelson, M. Thompson, "A One Meter Long Low-mass Mini-Drift Vertex Chamber used at the Tevatron Collider", in preparation.
12. M. Holler, S. Tam, H. Castro, R. Benson, "An Electronically Trainable Analog Neural Network (ETANN) with 10240 'Floating Gate' Synapses", proc. Int. Joint Conf. on Neural Networks, vol. II, pp.191-196, Wash., D.C., 22 June,1989, IEEE Catalog 89CH2756-6.