Track Reconstruction of Cosmic Muons using Artificial Neural Networks

Karthik S

2016MPH09

Supervisor: Dr. Deepak Samuel

School of Physical Sciences

Department of Physics

Central University of Karnataka

This thesis is submitted in partial fulfillment for the requirements of the Degree of

Master of Science (Physics)

11-04-2018

Declaration

This thesis is a presentation of my original work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussion. This work was done under the guidance of Dr. Deepak Samuel, Assistant Professor Central University of Karnataka (CUK).

I further declare that this thesis has been written by me and has not been submitted for any previous degree.

Karthik S
Name of the Candidate

11/4/18
Signature with date

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

Dr. Deepak Samuel
Name of the Supervisor

Signature with date
11/4/18.

Acknowledgements

I would first like to thank my thesis advisor Dr. Deepak Samuel of the Department of Physics Central University of Karnataka (CUK). The door to Dr. Deepak Samuel's office was always open for any clarification and queries relating to my research or writing. His consistent motivation and support through out the project steered me in the right direction in fulfilling my endeavor.

I would also like to thank the faculty members of the Department of Physics at Central University of Karnataka with special mention to Dr. Bharat Kumar and Dr. Rajeev Sesha Joshi for their valuable guidance through out my course-work at Central University of Karnataka.

I finally thank all the members who were instrumental in helping me come this far.

Abstract

The India-based Neutrino Observatory (INO) is a multi-institutional effort to build a underground laboratory to explore neutrino physics in detail. The INO is planning to setup a magnetized 50 kton Iron-CALorimeter (ICAL) with Resistive Plate Chambers(RPC's) as active detectors to study neutrino interactions. A prototype stack (without magnet) comprising of 12 layers of RPC's of **1 m x 1 m** is setup to track the cosmic muon. Since there is no magnetic field the cosmic muons transverse the stack in a straight line, therefore a straight line track is fit for the incoming cosmic muon in the prototype stack. This conventional method for track fitting of the cosmic muon proves to be inefficient in a noisy environment. An alternative fitting algorithm using Artificial Neural Networks (ANN) is developed and studied in detail in this work.

The results of comparison of the performance of conventional fitting algorithm with the ANN algorithm are presented.

Keywords: cosmic muon tracking, Artificial Neural Networks (ANN), INO prototype stack

(Part of this work is being prepared for submission to Journal of The IEEE Transactions on Pattern Analysis and Machine Intelligence)

Table of contents

Chapter 1

Introduction

1.1 India-based Neutrino Observatory

India-based Neutrino Observatory (INO) is a multi-institutional effort in building one of the largest underground laboratory with a 1 *km* rock cover for non-accelerator based high energy and nuclear physics studies in India. Over 50 scientists from more than 15 institutions and universities have collaborated to take up this huge project. In this effort, the National Neutrino Collaboration Group (NNCG) was formed with the goal of creating an underground neutrino laboratory with long term goal of conducting decisive experiments in neutrino physics and other experiments which require a unique underground facility.

An Iron CALorimeter (ICAL) detector with Resistive Plate Chambers (RPCs) as the active detector element is to be setup at INO to study the neutrino events. The detector will be setup at Pottipuram village in Theni district, Tamilnadu, India.

The physics goals of INO when the ICAL is deployed are:

1. Precise determination of oscillation parameters in the neutrino oscillations using the atmospheric neutrinos.
2. Study *Matter effects* through electric charge identification that may lead to determination of the unknown sign of one of the mass difference
3. Study of charge conjugation and parity (CP) violation in the leptonic sector as well as possible charge-conjugation, parity time-reversal (CPT) violation studies.
4. Identify and study very high energy neutrinos and multi muon events.

The INO in its first phase of operation will carry out the above studies using the atmospheric neutrinos. However several other possible future studies include:

1. High-precision determination of oscillation parameters when ICAL is used a far-end detector for a long-baseline neutrino oscillation experiment.

2. Neutrinoless double beta decay to determine if neutrinos are dirac or majorana particle

3. Solar, Supernova and Geo-neutrino studies.

4. Tomography of earth using laboratory neutrino sources.

1.2 Iron CALorimeter (ICAL)

Neutrinos are chargeless, therefore they cannot be detected by any direct methods, however it can be detected by estimating the energy and the direction of the leptons produced by the neutrino (ν) in the Charged-Current (CC) interactions. To precisely measure the energy and momentum of the neutrino a highly reliable and a sensitive detector must be used. INO implements the Iron CALorimeter (ICAL) as the proposed detector to detect the neutrino events. By applying magnetic field to the detector one can estimate the energy of the neutrinos and one can distinguish between neutrino and anti-neutrino events.

The proposed Iron CALorimeter (ICAL) is a large detector consisting of 50 kton magnetized iron plates interleaved with layer of Resistive Plate Chambers (RPCs) detectors with the time resolution of the order of the 1 *ns* and a spatial resolution better than 1 *cm*.

The proposed ICAL detector will have modular structure of size **48 m x 16 m x 12 m**. It consists of 140 horizontal stacks of 6 *cm* thick magnetized iron plates interleaved with 2.5 *cm* gaps to accommodate the RPC detector layers. Furthermore, the ICAL is subdivided into three modules each of size **16 m x 16 m x 12 m**. The total active area of the fully constructed detector is **108,000** m^2. Figure 1.1 is a schematic of the proposed ICAL detector.

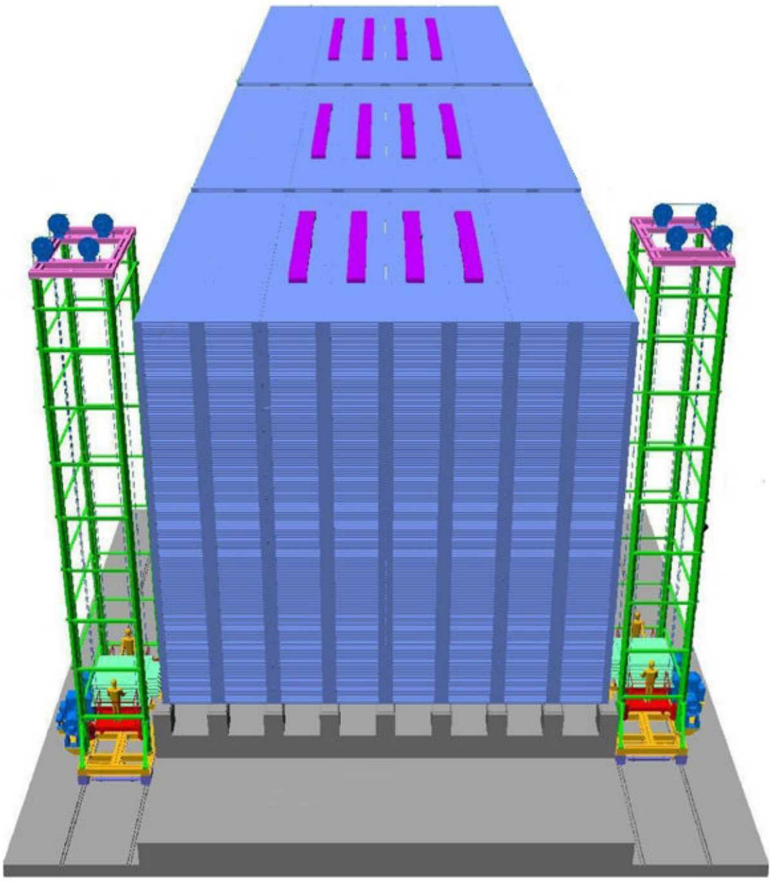


Fig. 1.1 Schematic of the proposed ICAL detector (INO/2006/01, Project report, Volume 1)

The entire Iron structure is magnetized with the field of about 1.3 Tesla using a giant coil (Violet in colour figure 1.1). Iron spacers are used to separate between the iron plates every 2 m along the X-direction thereby creating a road to insert the RPC in the Y-direction.

1.3 Detector development

In order to understand and estimate the performance of the final ICAL detector, many prototype detectors have been built to study the cosmic muons. From the prototype detectors many detector based studies and physics studies have been made in the past. These form a standard for the finally proposed ICAL. The detector parameters of the prototypes are closely monitored over a long period of time and are being improved in every phase of the up-gradation. Two major prototypes have been built in the recent years:

1. A 12-layered glass-RPC stack without a magnetic field. This was developed and is setup at the Tata Institute of Fundamental Research (TIFR) Mumbai India.

2. A 12-layered Bakelite RPC stack with a magnetic field. This was developed and is operational at the Variable Energy Cyclotron Centre (VECC) Kolkata India.

In this project we have used the TIFR prototype stack parameters to develop an alternate track fitting algorithm.
The TIFR prototype stack does not have a magnetic field. As a result, the cosmic muons passes through the detector in a straight line. A linear fit is made to the track of the muons after applying some logic cuts. Though this algorithm is very robust, it fails to reconstruct a noisy cosmic muon event.
This project aims to answer this following question: How effectively can one reconstruct the cosmic muon track in a noisy environment?
In this project, this question is addressed with the implementation of the Artificial Neural Networks (ANN) for the track fitting of the cosmic muons.
Chapter 2 discusses in detail about the construction and working of the TIFR prototype stack. It also deals with the physical parameters that the detector yields from a cosmic muon event. It also discusses in brief about the Data Acquisition System and the data storage in the TIFR prototype stack.
In chapter 3 the current track fitting algorithm is explained. The drawbacks and the shortcomings of the model are discussed in detail.
Chapter 4 gives a brief introduction to the Artificial Neural Networks and discusses the framework and implementation of the ANN to track fit for the cosmic muons.
Chapter 5 gives the results of ANN and conventional method for track-fitting. The chapter also compares the performance of both the algorithm for track fitting.
Finally, chapter 6 gives the conclusion about the results from the ANN and linear model. It also discusses the future scope of this work.

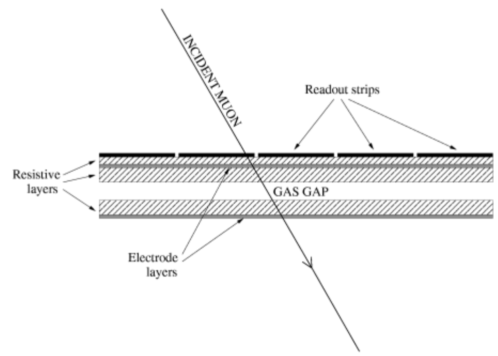
Chapter 2

TIFR prototype stack

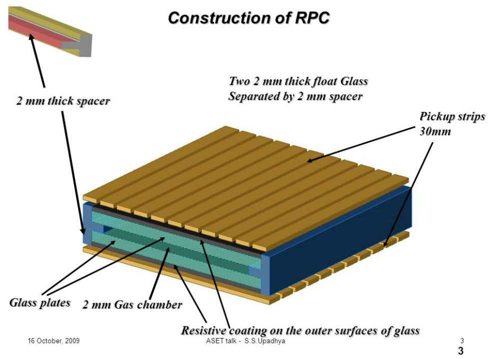
INO prototype stack is a 12 layered RPC stack each of **1 m x 1 m** with 32 pick up strips each on the X and Y plane to record the track timing signals of the cosmic muon. This section describes the construction and working of Resistive Plate Chamber (RPC) and the TIFR prototype stack.

2.1 Resistive Plate Chambers (RPCs)

The Resistive Plate Chamber (RPC) was developed by R. Santonico and R. Cardarelli in 1981. The typical time resolution of the RPC is about 1 *ns* and spatial resolution is better than 1 *cm*. RPCs are typically used as tracking detectors which require large sensitive area and in low counting rate experiments.



(a) Muon passing through the gas chamber (http://cms.web.cern.ch/news)



(b) Cross-sectional view (INO/2006/01, Project report, Volume 1)

Fig. 2.1 (a) shows the working of RPC when a charged particle enters the chamber and the (b) shows different components of a RPC.

Resistive Plate Chambers (RPCs) consist of two parallel plates, a positively-charged anode and a negatively-charged cathode, both made of a very high resistivity plastic material and separated by a gas volume. A high electric field is maintained between the plates. A suitable ionizing gas usually inert gases or a mixture of gases is passed through the gas gaps and pickup strips are grooved to the pickup panels at its positions as shown in the figure 2.1b. When a charged particle with minimum ionizing energy (E > 2 MeV) pass through the gas chambers (2.1a) it ionizes the gas mixture filled in the chamber. Due to the presence of the electric field, the ions and the electrons drift to the respective electrodes, during the drift the electrons ionizes the gas molecules in its drift path thereby producing more electrons, by the time the electron reaches the anode a avalanche of electrons is produced which in turn generates a signal.The signal is then picked up the the pick-up strips that are grooved to the pick-up panels. High resistivity material (glass for INO prototype stack) confines the charge and localizes them in space and makes it remain active for the time. This working of the RPC is called the avalanche mode of operation. The Resistive Plate Chambers (RPC) can be operated in (i) Avalanche mode and (ii) Streamer mode.The INO RPCs are operated in the avalanche mode.

2.2 INO RPC stack - Construction and working

2.2.1 INO prototype construction

The RPCs at the INO stack works in the avalanche mode.The INO prototype stack is made of 12 layers of RPCs with a gap of 16 *cm* totalling the stack height to 176 *cm*. Following are the properties of RPC layers of the TIFR prototype stack.

1. Two, **100 cm x 100 cm** glass plates ($\rho = 10^{10} - 10^{12}\Omega$) sandwiched between the electrodes (2.1b).

2. The RPCs are operated at a high electric field of about 10 kV, the glasses are coated with graphite to increase the resistivity which helps in setting a uniform electric field.

3. The two plates are separated by polycarbonate button spacers placed at equal intervals. The side and the corner spacers are used to completely seal the RPC layers airtight.

4. An ionizing gas like Argon (Ar) is filled in the gas gap between the glass plates. Freon (R134a) is used to absorb the free charged particles in the gas to prevent undesirable avalanche. Iso-Butane is added to absorb the photons produced during the recombination process, restricting the occurrence of secondary avalanche. The ionizing gas may attain saturation when quenching gases are not mixed in right proportion therefore small amount of SF_6 is added as quenching gas. For the functioning of RPC in avalanche mode the gas mixture Freon:Iso-Butane:SF_6 are added in the ratio 95.5:3.9:0.6.

5. The pickup panel is made from the plastic honeycomb core 5 *mm* thick, 50 micron Aluminium sheet on one side for grounding and copper pickup strips on the other side. Two pickup panel is kept on over the RPC, these pickup panels are placed orthogonal to each other each having 32 pickup strips totaling to 64 strips, the pickup strips give the position of the hit in the X-Y plane.

6. The pickup panel and the RPC are separated by a thin insulating mylar sheath.

7. The entire setup is packed in a Aluminium casing.
   A schematic of the TIFR prototype stack is shown in the figure 2.2 below.

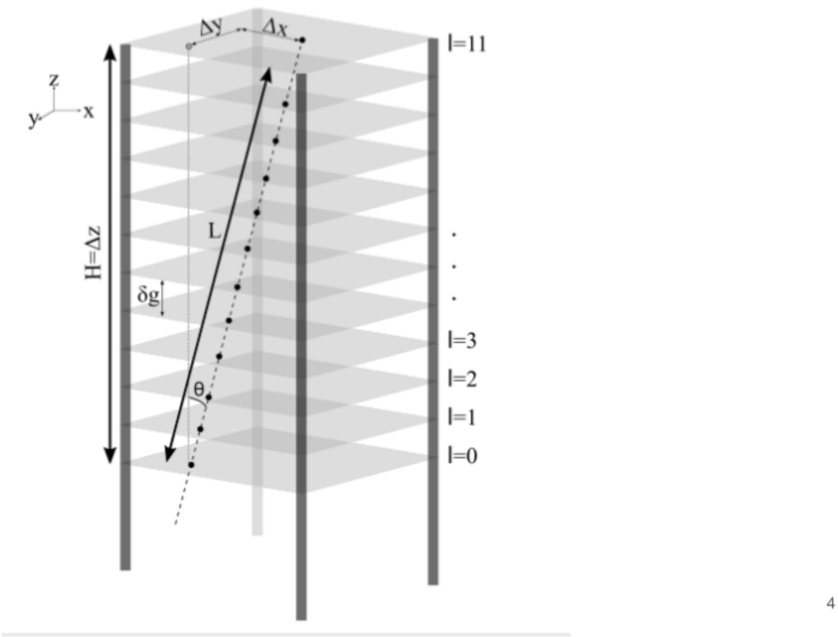


Fig. 2.2 Schematic of the TIFR prototype stack illustrating the physical parameters ([1]).

2.2.2 INO prototype Data Acquisition System

The INO prototype uses the standard VME-based data acquisition system for the recording of the data from the stack. When a muon passes through the stack a signal is generated in the RPCs in each layer, these signals are picked up by the pickup strips on both the X and Y pick-up panels.This gives the X and Y coordinates of the muon. The layer number gives the Z coordinate. These signals are transmitted via the strips acting as the transmission lines to the electronics. The signal from the strips are first pre-amplified and then fed into the front-end electronics where Analog Front End modules (AFE), Digital Front End modules (DFE) and signal router are present for data acquisition. If a trigger signal is received confirming the passage of the muon then the processed signal are latched to the VME-based module that records the events. Finally the VME controller is interfaced with the PC for data analysis. The following parameters are stored as a ROOT data file in the ROOT framework:

1. X position in TBits format - a sequence of 32 binaries representing hits in a layer

2. Y position in TBits format - a sequence of 32 binaries representing hits in a layer

3. Timing information as measure from the TDC (V1190B,CAEN) with 100 *ps* LSB.

4. Noise rate of the strips at regular intervals.

2.2.3 Trigger logic for TIFR prototype stack

The trigger system for the INO prototype stack is a simple algorithm. There must be a signal from the top(0^{th} layer), middle (6^{th} layer) and the bottom (11^{th} layer). This 3 fold coincidence generates the trigger for the data acquisition. If triggered, the processed signal gets latched to the VME-based data acquisition system.
The trigger logic can be expressed as follows.

$$Y = T * M * B$$

where Y = output from the trigger
T = Top layer hit (0^{th} layer logical)
M = Middle layer(6^{th} layer) hit (logical)
B = Bottom layer(11^{th} layer) hit (logical)

Chapter 3

Linear Track Fit

As mentioned in chapter 1, before deployment of the main ICAL detector many performance tests are carried out on the prototype stack to study the long term efficiency and performance of the stack. As a part of the detector based studies, parameters like detector efficiency, strip multiplicity, noise rates and time resolution are constantly monitored. Also, many physics based studies have been performed continuously to build a calibration database for the main ICAL. Physics based studies include the zenith angle distribution of the cosmic muons, velocity calculation of the cosmic muons which are performed by fitting a straight line to the incoming cosmic muon.

There are electronic noises and some detectors might give out false signals thereby biasing the track of the actual muon. On other hand, the avalanche created during the passage of the muon may fire more than one pickup strip resulting in a cluster of strips getting fired. For example, when a cosmic muon enters the gas chamber of a layer there is a finite probability that random strips in the layer may get fired due to electronics and also there is a finite probability that the muon may pass through the center of two strips which may fire adjacent strips. Therefore, when the trigger records an event it records the track with its noises. Hence, an appropriate logic must be applied to filter out the noises and the bias hits and fit a track that nearly follows the actual one. A linear fit is performed after applying some logic cuts. This algorithm is used for a very long time to reconstruct the track of the cosmic muons.

3.1 Current linear fit algorithm

As discussed in the previous section many noise hits are recorded along with the track hits in an event. The following logic cuts are applied to reconstruct a valid cosmic muon track at the INO prototype detector. The logic cuts are applied independently to the X and Y plane hits since they are uncorrelated.

1. If in a layer, there are more than 2 hits being fired then the layer is rejected and will not be considered for reconstruction.

2. If in the layer, the number of hits is less than or equal to 2 but they are away by 2 strip-widths then that layer is rejected and will not be considered for reconstruction.

3. If the layer is accepted, then the average of the strip hit is taken as the hit point for reconstruction.

4. After rejection of the layers, if total number of accepted layer is less than 4 then the entire event is rejected and will not be reconstructed.

5. If the event is accepted, a linear fit is made using the points obtained.

   Sometimes there might be some outliers that might bias the linear fit parameters. To reduce these bias effects caused by the outliers, the strips that are away by 2 strip-width from the fitted line are removed and after removing if the total number of layers is greater than 2 then a second track fit is made.

For instance, the above discussed algorithm is applied for the sample track image shown in the following figure.

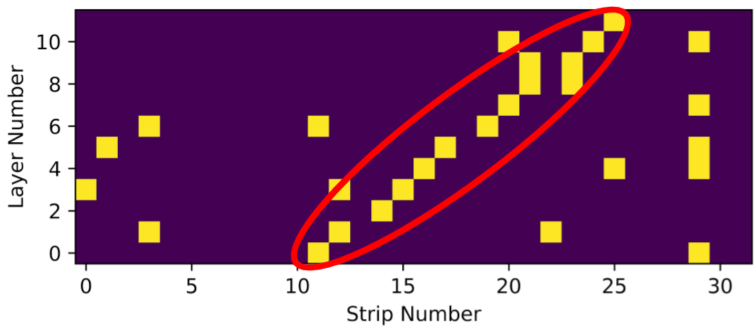


Fig. 3.1 Event display showing possible cosmic muon track encircled in red along with noise hits.

Figure 3.1 shows an event display from which the track of the cosmic muon is deduced. There are 32 strips on the X axis and 12 layers on the Y axis. The muon when passing through the layers will fire a particular sequence of strips corresponding to the track, however there may be electronic noises and other bias effects which gives extra hits other than the actual track. When the linear track fit algorithm is applied to the event shown in the figure 3.1 we infer the following:

1. The 11^{th} layer is most probably accepted since there is only one hit in the layer. The 11^{th} layer is accepted for the reconstruction of the event.

2. The 10^{th} layer is rejected since there are more than 2 hits in the layer (3 hits). Therefore it is rejected for the reconstruction of the event

3. The 8^{th} and the 9^{th} layers are accepted since they are within the 2 strip-widths.

4. The 7^{th} layer however will be rejected, since there are 2 hits which are away by more than 2 strip-widths.

5. We find now that the total number of accepted layer after first fit is 1, therefore, the entire event will not be reconstructed.

The current algorithm fails in reconstructing the event due to the noise hits. However, one can visually deduce the track of the muon to be the one encircled in red as shown in the figure 3.1. Therefore an alternative algorithm must be developed which can reconstruct the event even in an noisy environment. This project intends in developing such an algorithm with the help of Artificial Neural Network (ANN).

Chapter 4

Track Fitting using Neural Networks

4.1 Introduction to Artificial Neural Networks (ANN)

The idea of Artificial Neural Network (ANN) emerged way back in the 1940s, however the extensive use of the ANN was recently in the field of image and pattern recognition. The ANN performs a task given, by learning through its experience or training. ANNs have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis. Recently, high energy physicists have started to explore the use of ANN and deep learning algorithms in their field. This project intends in exploring the strength of ANN in identifying the track of the cosmic muons in a noisy environment. The ANN is inspired by the way the brain recognizes the object. Whenever a object is seen a set of neurons in the visual cortex get fired, these neurons are finally connected to a set of neurons present in the part of the brain that is responsible for image recognition. Every object has at least some uniqueness in it and that will be identified by the brain, for example when a dog's image is seen it has a unique outline or a feature which differentiates it from the other object. This is picked up by the neurons and a particular set of neurons fires, the same set of neurons fire when another dog's image is seen, in this way the brain recognizes an image. It picks out the most prominent feature from the given image. The Artificial Neural Network (ANN) tries to replicate this functionality of the brain. As an illustration, one can represent the ANN in the following way (Fig:4.1).

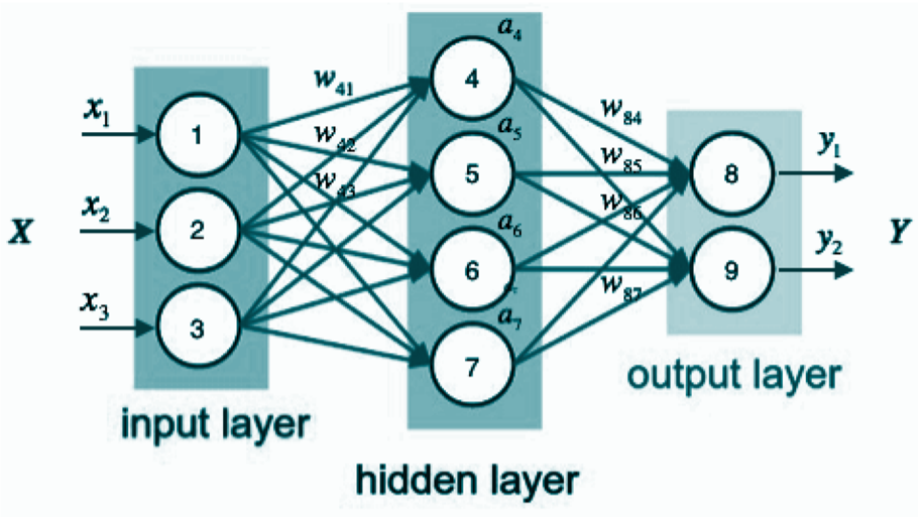


Fig. 4.1 Schematic of an Artificial Neural Network with input layer, hidden layer and output layer with its weight function to each neuron-neuron pair (https://www.digitaltrends.com).

In figure 4.1 the image will be sensed by the visual cortex which is a set of neurons these neurons are represented here as the input layer(x_i's fig:4.1). The input layer is finally connected to a set of neurons in the brain which is responsible for image recognition these neurons are represented here as the output layer. The input neurons are connected to a set of output neurons via a sequence of neurons called as the hidden layer. As in figure 4.1, connection between each neuron neuron pair is characterized by its weight function (w_{41})[4.1]. The weight function can be interpreted as the Bayesian probability function, i.e, how much can one neuron influence its connected neuron. Each neuron carries an activation in it which is a number between 0 and 1 (δ is the activation). A neuron is called activated when its activation is close to 1. For instance the weight w_{41} is the probability of x_1 activating 4th neuron in the hidden layer. Similarly the weight w_{84} is the probability of the 4^{th} neuron in hidden layer will activate the output layer neuron.

As a summary, whenever an image is fed into an ANN, a set of input neurons get fired and the weights are a measure of connection between the neurons of different layers, depending on the input and the weights a set of neurons in different layers fire finally pointing to a single output which corresponds an image. The weights are determined by training the network for sufficient number of times. Different inputs with known outputs are given to the neural network and initially random weights are assigned to the neuron-neuron pair and through successive iteration the the weights are optimized by the method of error back-propagation to achieve the best weights. Once when the neural network model is sufficiently trained new image is fed as input and the output is predicted with a probability.

Components of a ANN

1. **Neuron**
   A artificial neuron resembles a biological neuron which takes in an input and gives out an output depending on the input.

2. **Activation**
   Every neuron is associated with an input activation function (δ) which is computed by taking the inputs from the neurons from previous layer. The output of the activation function is called as activation. After the activation the output of neurons form the input to the neurons in the next layer.

3. **Output function (a_{ij})**
   The output function takes the activation of the neurons and squeezes them into a number within a given range usually [0,1] or [-1,1].

4. **Weight function (w_{ij})**
   The network consists of connections, each connection transferring the output of a neuron i to the input of a neuron j. In this sense i is the predecessor of j and j is the successor of i. Each connection is assigned a weight w_{ij} it is the Bayesian probability of j being fired due to i.
   Given an activated neuron which is connected to many other neuron the probability that the connected neuron gets activated is called as its weight.

5. **Learning rule**
   The learning rule is a rule or an algorithm which modifies the parameters of the neural network, in order for a given input to the network to produce a favored output. This learning process typically amounts to modifying the weights and thresholds of the variables within the network.

6. **Back propagation algorithm**
   Back propagation is a method to calculate the gradient of the loss function (produces the cost associated with a given state) with respect to the weights in an ANN.

A detailed documentation of the ANN using scikit-learn is given in reference [4]. The first successful application of the ANN was in the identification of the handwritten digits.

4.2 Sample application of ANN : Handwritten digit recognition

In this section, a detailed explanation about working of ANN in recognition of a handwritten digit is given. This is a very successful application in the field of pattern recognition. This example is discussed since it has a close resemblance to track fitting using ANN for cosmic muons.

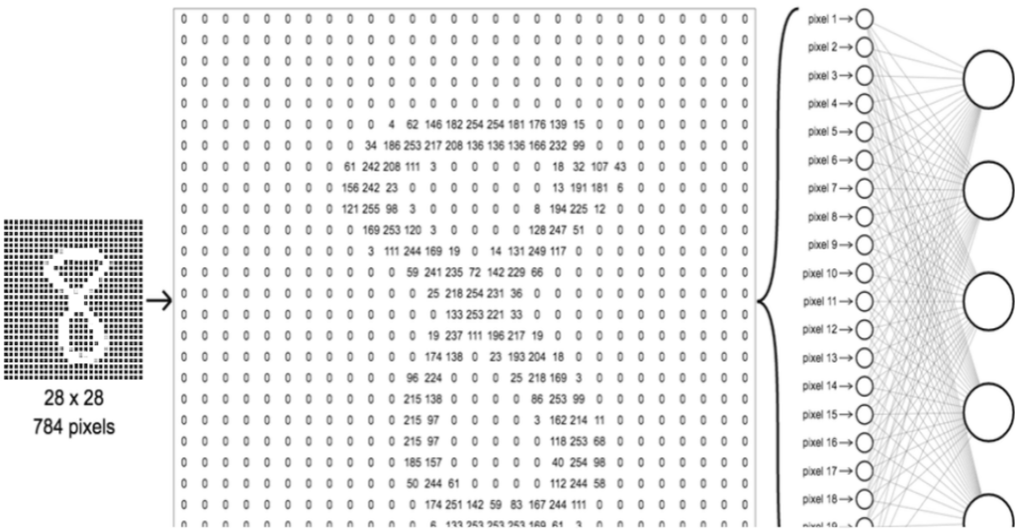


Fig. 4.2 Handwritten digit 8 seen as an array of numbers where black represents 0 and white represents 1 by the ANN model ($https://ml4a.github.io/ml4a$).

Figure 4.2 shows an image of handwritten digit 8 along with the colour code representation. It is a **28 x 28** pixel image and the neural network takes this image as a 2D array of size **28 x 28** with a total of 784 input neurons. The image is flattened and then fed into the neural network. As a consequence, as discussed in previous section, a particular sequence of neurons corresponding to the digit 8 will be fired.

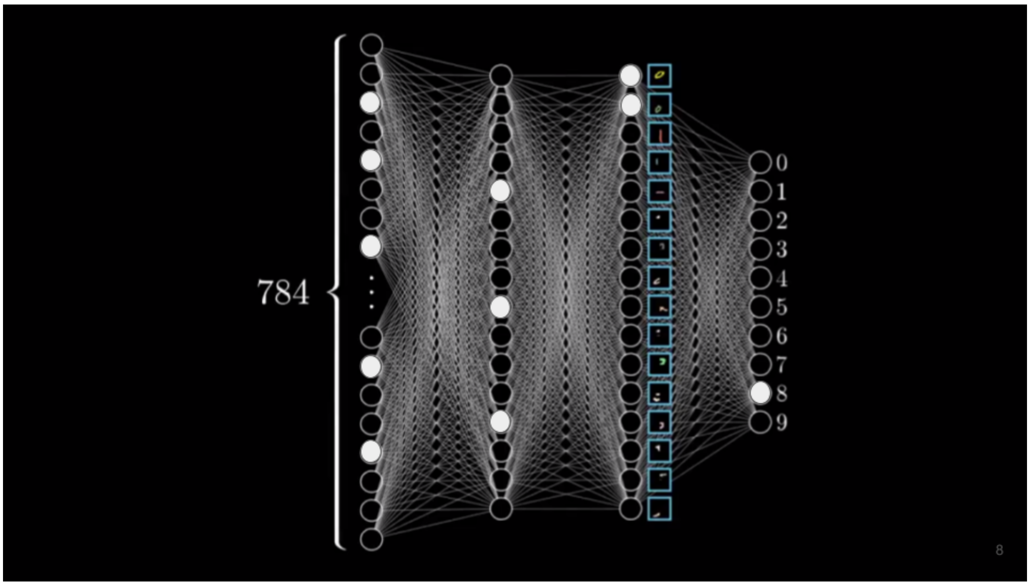


Fig. 4.3 Illustration of a ANN with its hidden layers for digit recognition (*https://ml4a.github.io/ml4a*).

In this example there are a total of 784 input neurons out of which a sequence of them fire for a digit 8, these neurons are finally connected to a set of 10 output neurons which corresponds to the number from 0-9. Here, the input and the output layer of neurons are mediated by two hidden layers and each neuron-neuron pair is assigned a weight. To optimize the weight one has to train the model sufficiently. In this context different images of the same digit 8 written by different people are given as input and the output is set to the neuron corresponding to 8. The neural network over every iteration adjusts the weight to optimize them to finally fire the neuron corresponding to 8. The same procedure is then repeated for the other numbers. Once when the neural network is sufficiently trained a new digit image can be fed and it would give out the most probable digit.
In this project a similar algorithm is used to retrieve the cosmic muon track from a noisy event. The input to the ANN is a noisy track and the output is the slope and the intercept of the cosmic muon track. We train the model sufficiently with different noisy events as input with known slope and intercept as output. Once sufficiently trained the ANN model predicts the slope and intercept for any inputed noisy event. A simulation framework to generate these noisy tracks was developed to study the efficacy of the ANN in the context of our project. This simulation framework is discussed in detail in the next section.

4.3 Simulation framework

A noisy track event is generated and is fed into the ANN as its input. Physical parameters were taken into account to create a noisy track event that closely resembles the tracks generated in the TIFR prototype stack. To create such a noisy event the following procedure is followed:

1. Generation of a clean track
2. Fold with detector efficiency (η)
3. Addition of strip multiplicity (M_S)
4. Addition of random noise (M_N)

The simulation framework is built using ***Python 3.6***.

4.3.1 Generation of a clean track

The first step of the simulation framework is to generate a clean track. A random slope and an intercept are generated for a straight track. In this project we focus on only fully contained events, i.e, cosmic muons tracks that traverse through all the layers. Therefore, the generated slope is constrained within a range [-3,3]. An intercept is generated within the range of [0,31] so as to constrain the strip hit in the 0^{th} layer within the detector volume. Now, a null matrix of size **12 x 32** is created which corresponds to 12 layers with their 32 strips in the TIFR prototype (figure 2.2). The matrix elements closest to the line generated from the slope and intercept are set to 1. The matrix representing this data of a clean track is shown in figure 4.4 with yellow points representing the elements set to 1.

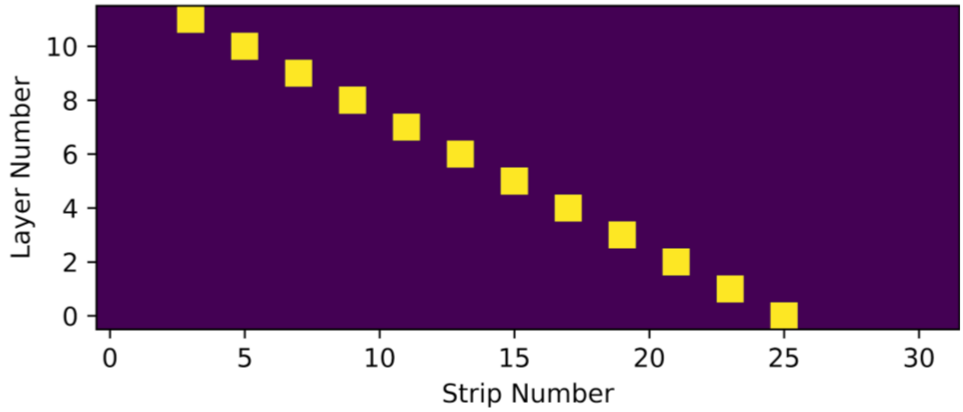


Fig. 4.4 An event showing clean track of the cosmic muon with no noise added to it.

4.3.2 Detector Efficiency (η)

To account for the detector efficiency (η), a random number between 0 and 100 is generated for every layer. If the generated random number is less than η then the matrix element corresponding to hit is set high, else if the generated random number generated is greater than the detector efficiency then the matrix element corresponding to hit is set to 0. For INO prototype RPC detector, the efficiency ranges between 95% to 99%. Figure 4.5 shows the matrix after folding with the efficiency of 80% for figure 4.4.

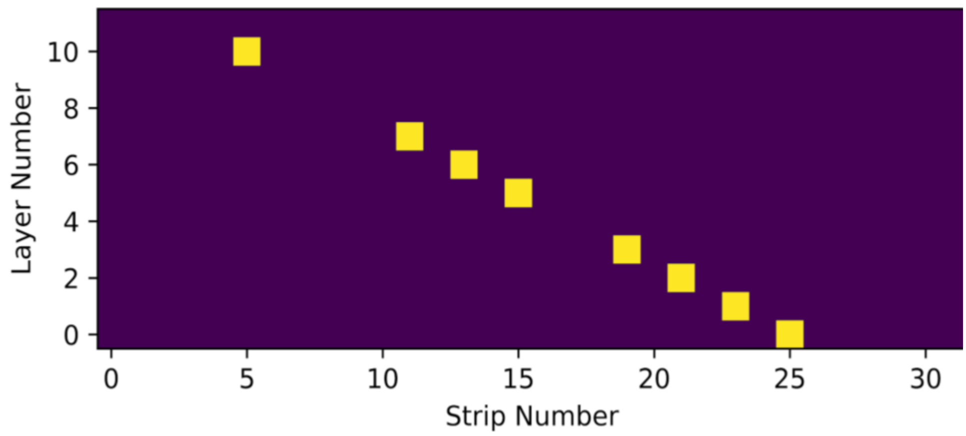


Fig. 4.5 Track after convolution of detector efficiency ($\eta = 80\%$)

4.3.3 Strip Multiplicity (M_S)

The simulation framework also accounts for the strip multiplicity of the detector. When a muon passes through the gas chamber in an RPC there is finite probability that more than one pickup strip in a layer gets fired. The average number of strips fired due a muon per layer is defined as the strip multiplicity (M_S). Previous experimental data from the TIFR prototype stack have shown the strip multiplicity to be slightly greater than 1.5. Therefore, we have assumed the strip multiplicity (M_S) to be 2. To account for the strip multiplicity, a random number is generated between 0 and 150 for each layer. If the random number generated is between 0 and 50, the matrix element corresponding to left of the actual hit is set to 1, else if the random number generated is between 100 and 150, the matrix element corresponding to right of the actual hit is set to 1. If the generated random number is between 100 and 150, no extra hits are added. The track, after the addition of effect of the strip multiplicity, is shown in the figure 4.6.

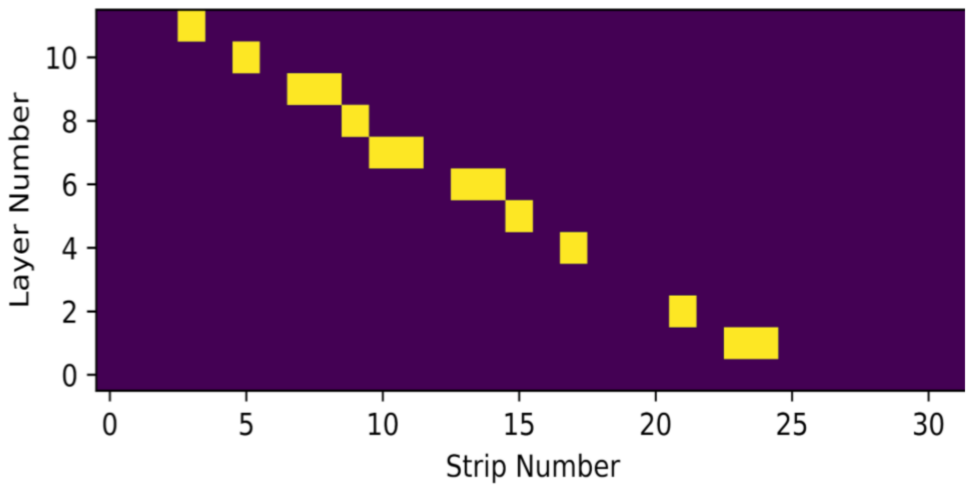


Fig. 4.6 Track after accounting for Strip multiplicity ($M_S = 2$)

4.3.4 Noise Multiplicity (M_N)

There are noise hits along with the cosmic muon hits which may arise due to detector electronics or any other external noises present. An additional parameter called as Noise Multiplicity (M_N) is defined to account for this noise. Noise multiplicity (M_N) is defined as the maximum noise hits per layer per event. A random number (integer) R is generated between 0 and M_n for each layer. Then, R random numbers between 0 and 32 are generated and the corresponding strip number is set to 1. Figure 4.7 shows the matrix after including the noise multiplicity (M_N) of 4 for figure 4.6.

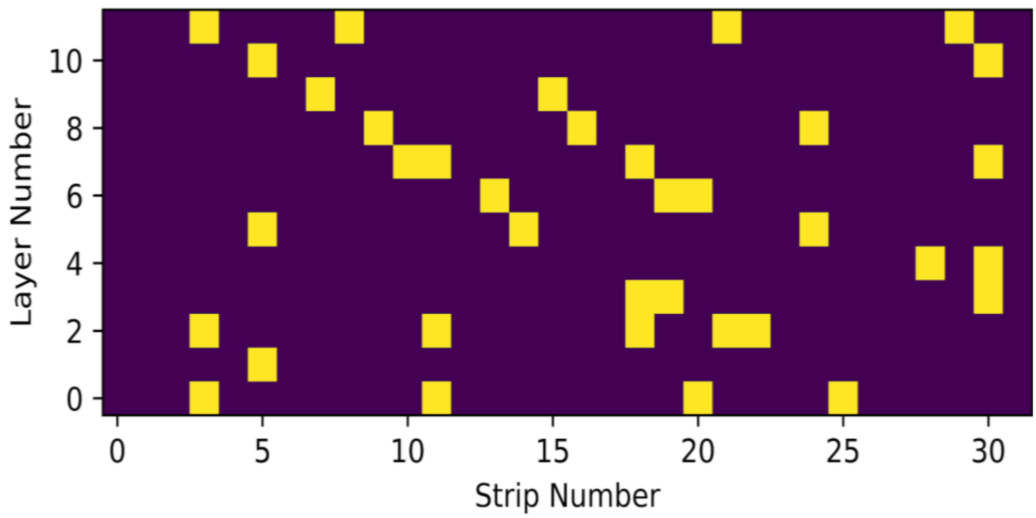


Fig. 4.7 Track after accounting for Noise multiplicity($M_N = 4$)

This generated noisy event will serve as the input to the neural network model. The ANN in return will output a slope and an intercept from which the clean track can be reconstructed.

The entire simulation framework is summarized in the figure 4.8.

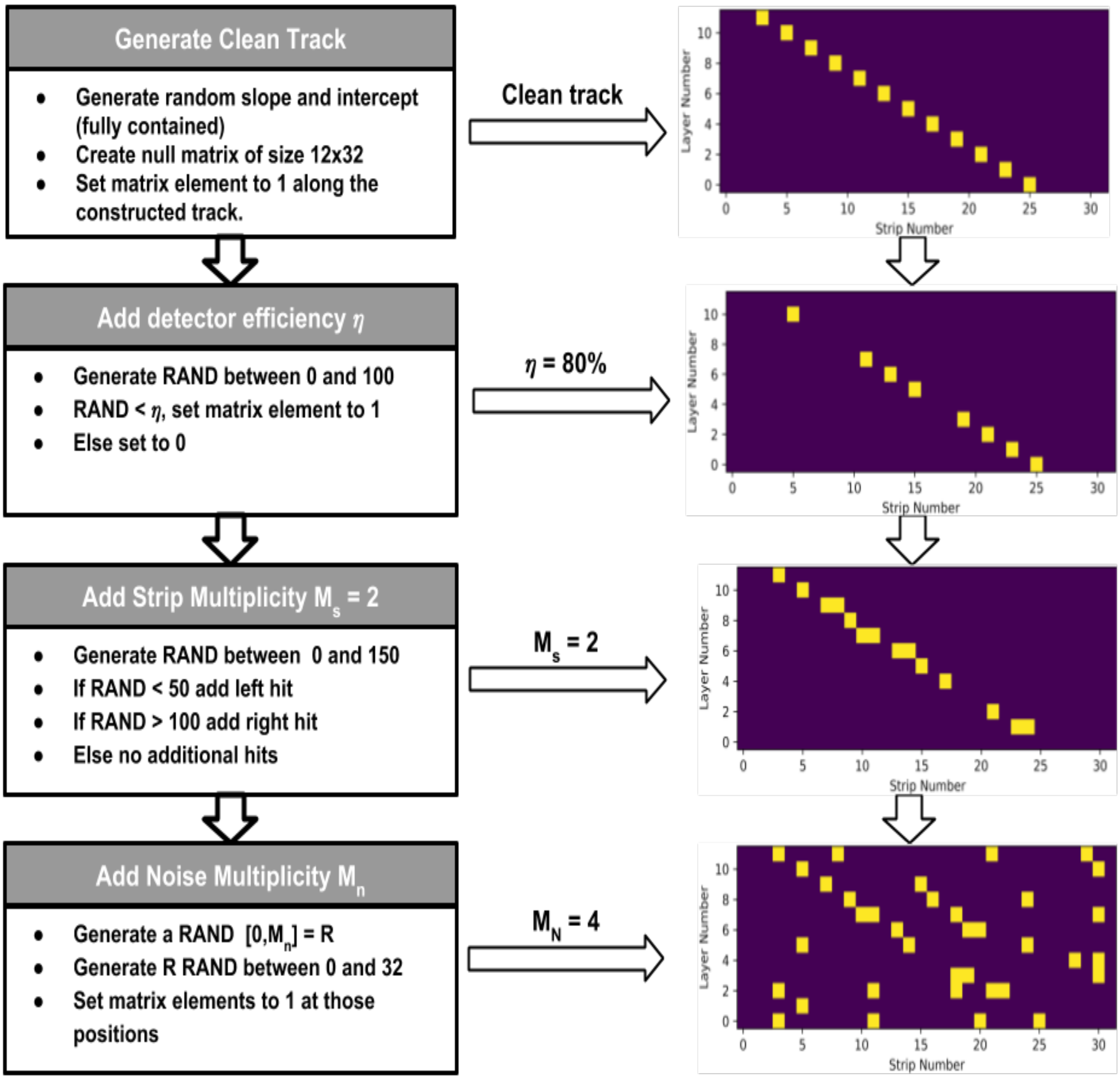


Fig. 4.8 Simulation framework to simulate noisy events of the cosmic muons. This serves as input to the ANN to predict the track of the cosmic muon

4.4 Implementation of ANN algorithm

The noisy track event generated from the above simulation framework is fed as input to the neural network model. To implement the ANN, the scikit-learn module was used. The scikit-learn Multi-Layer Perceptron (MLP)-Regressor module was used to train and test the events. Following are the steps involved in implementation of the algorithm:

1. A total of 6 million train events are generated using the developed framework (Noisy image with slope and intercept) with the following parameters:

   (a) Noise multiplicity (M_N) equally distributed from 0 to 5.

   (b) Detector efficiency(η) = 100%.

   (c) Strip multiplicity(M_S) = 2.

   There are a total of 6 set of events with each set containing 1 million events. Each set has a different noise multiplicity (from 0 to 5).

2. The ANN had 2 hidden layers each of dimensions **6 x 32** with an activation function as ***tanh***.
   The ANN model for predicting the track of the cosmic muons has 2 hidden layers. The activation function is a mathematical function which transforms the total activation of the neuron (refer Activation) to a number between 0 and 1. If the number is close to 1 then the neuron is said to be activated.

3. ***Adam*** solver was used as the optimizer. It uses the method of stochastic gradient optimization to predict the output of the neural network. The solver is easy to implement, highly efficient for large number of data, and it consumes less memory when compared to other solver algorithms. It also proves highly reliable and efficient for noisy data.

4. The ANN outputs a single value of slope and the intercept. Random weights for every neuron-neuron pairs are seeded into the network. Finally the neural network returns a slope and an intercept depending on the initialized weights. The entire process is iterated over fixed number of iterations till desired performance is achieved.

5. Metric is a tool to measure the performance of the neural network. In this model the Mean Square Error(MSE) loss is used as metric. The MSE is the

Mean Square Error between the ANN prediction and the true value. MSE back propagates through the network and adjusts the weights according to the solver to optimize the results in every iteration. This iteration optimizes the weights and the prediction becomes closer and closer to the actual value in every iteration. The loss for every iteration decreases since the weights are optimized over every iteration. In this ANN model, a total of 500 iterations are made to reduce the loss to about 0.00001.

6. A total of 0.6 million test events were generated and the results from the ANN were compared to that of the actual slope and the intercept.

7. A total of 0.6 million test events with detector efficiency 95% and 90% were also studied in detail and the results are compared and analyzed.

The entire implementation of the ANN algorithm is pictorially represented in the figure 4.9

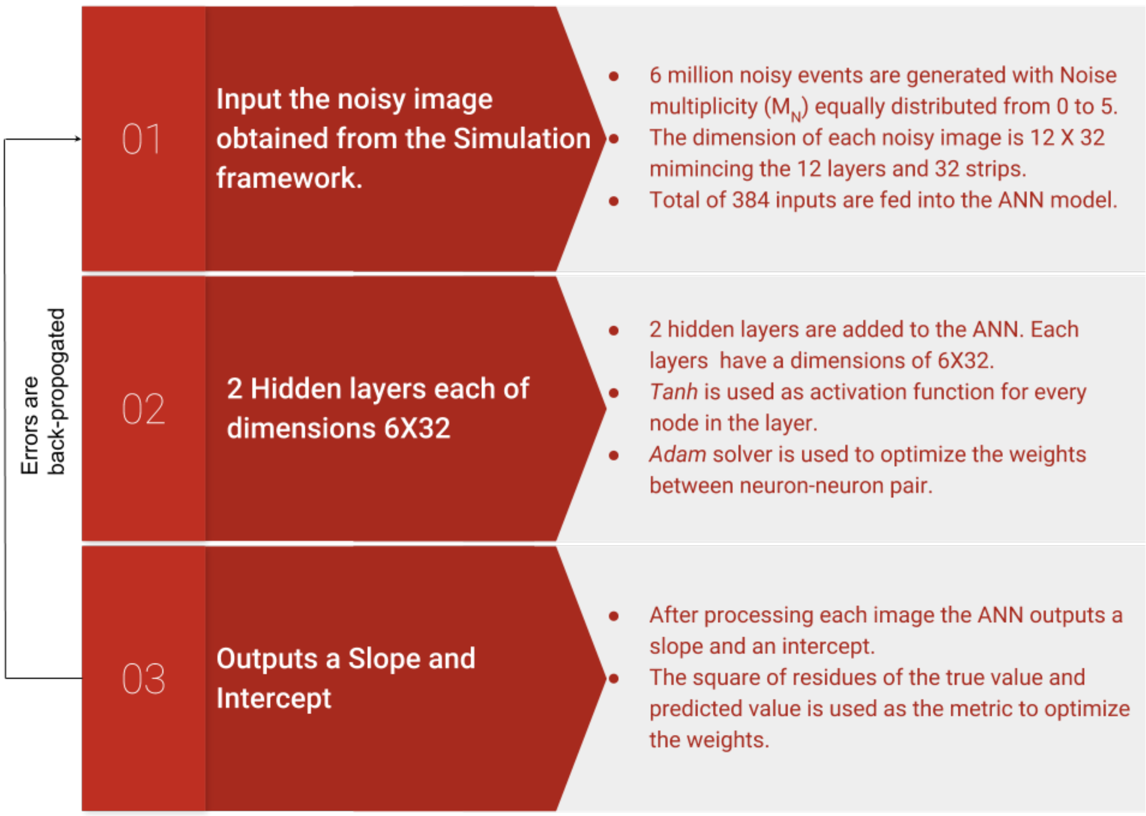


Fig. 4.9 Steps involved in the ANN implementation for predicting the track of the cosmic muon from a noisy event.

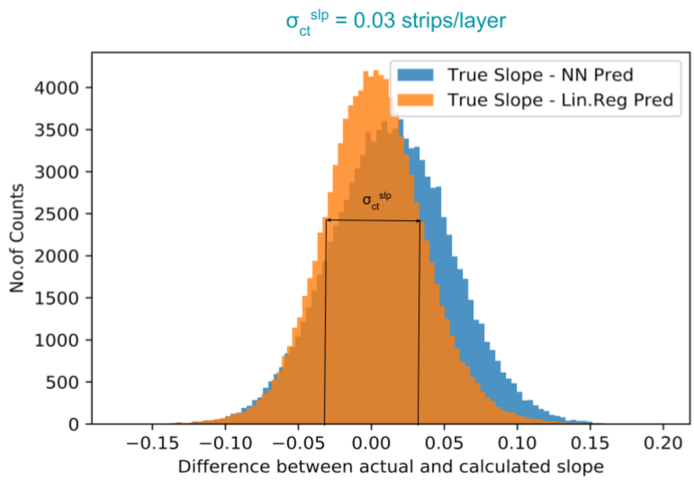
Chapter 5

Results and Analysis

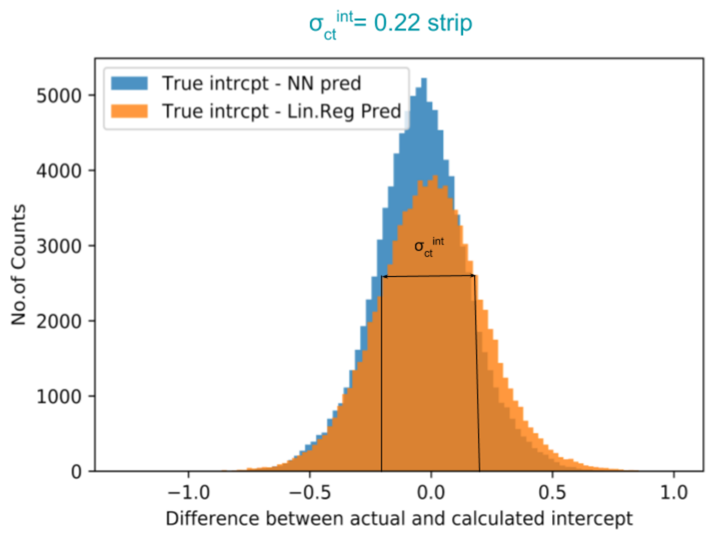
In this section, the trained model of the ANN was tested with 0.6 million test events with varying noise multiplicities (M_N). The results of the model are tabulated and discussed in detail. A slope and an intercept is returned by the ANN model. The true value of the slope and the intercept are compared and if the predicted slope and intercept are within allowed uncertainties then the fit parameters are used to reconstruct the track. The results of the ANN model along with the conventional model are compared in detail in this section.

5.1 Residual distributions of slope and intercept

A histogram of residues of the slope and residues of the intercept, i.e, the difference between the true value and the predicted value from the ANN is plotted and standard deviation (σ) gives the accuracy of the predicted values.
For benchmarking, 0.1 million clean tracks were generated and were given as input to the ANN. Similarly, a linear fit was also made for the clean track (events with $\eta =$ 100%, $M_N = 0$, $M_S = 1$). The best fit parameters are obtained by fitting a straight line to the clean track. Therefore, the results from the linear fit and the ANN predictions are compared and the results are studied.
Figure 5.1 is the histogram of the residual distribution of the slope and the intercept.



(a) Residual distribution of slope



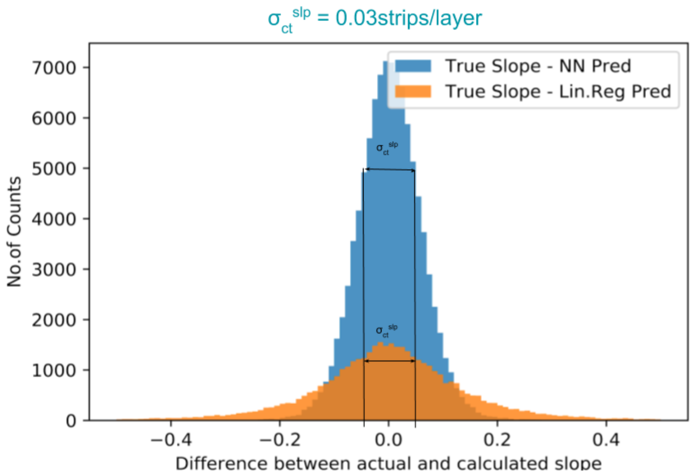
(b) Residual distribution of intercept

Fig. 5.1 The residual distribution for the prediction by linear and ANN model for a total of 0.6 million clean tracks i.e events with $M_N = 0$, $M_S = 1$, $\eta = 100\%$.

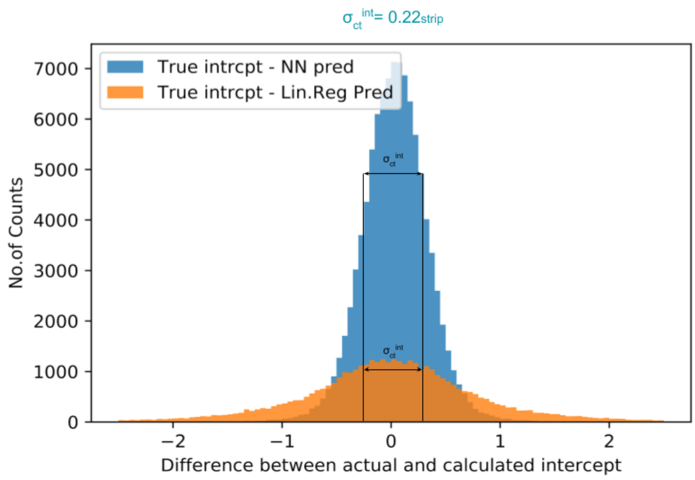
Figure 5.1 reveals that both the linear and ANN predictions are very close and the standard deviation of the linear model distribution for both slope and the intercept is the uncertainty in the measurement of the results. Figure 5.1, as expected, has a gaussian profile and its standard deviation is comparable to that of theoretically calculated values ($\sigma_{CT}^{slp} = 0.024$ strips/layer, $\sigma_{CT}^{int} = 0.19$ strips). There is, however, a slight deviation between the linear and the ANN model which is yet to be studied in detail. The σ_{CT} of the slope and intercept will serve as the bench mark to evaluate the efficacy of the model.

A noisy event with strip multiplicity (M_N) is given to the ANN and if the predicted slope and intercept are within the uncertainty of the $3\sigma_{CT}$ for both slope and intercept, the reconstruction is counted to be a valid reconstruction.

Following the same logic, a study from the residual distribution of the slope and the intercept was made on the variation of the reconstruction efficiency by varying the noise multiplicity from 1 to 25. As an example, figure 5.8 shows the residual distribution of slope and intercept from a linear and ANN model for noisy events i.e events with strip multiplicity(M_S) = 2, noise multiplicity (M_N) = 3, detector efficiency (η) = 100%.



(a) Residual distribution of slope



(b) Residual distribution of intercept

Fig. 5.2 The residual distributions for the prediction by linear and ANN model for a total of 0.6 million noisy events with $M_N = 3$, $M_S = 2$, $\eta = 100\%$ generated for 0.6 million test events.

Figure 5.2 clearly shows the inefficiency of the current linear fit model. In the data sample studied, there are a maximum of 3 noise hits per layer and the strip multiplicity is set to 2. Therefore, a total of 4 hit per layer on top of the actual hit is possible. With the linear fitting algorithm discussed in the section (3.1) most of the events will not be reconstructed which is the reason for the drastic fall in the total number of counts in the case of linear model distribution. As discussed previously, a cut is made using the standard deviation (σ_{CT}) within which the events are reconstructed. Due to less number of counts, the spread (σ) in the case of the linear model is large when compared to the ANN distribution. Qualitatively, it is evident that the ANN reconstructs more number of events compared to linear model within the allowed uncertainties.

5.2 Reconstruction Efficiency (η_{rec})

To quantify the reconstruction ability of the ANN and the linear model a parameter called as reconstruction efficiency (η_{rec}) is defined. The reconstruction efficiency is given by:

$$\eta_{rec} = \frac{Total\ numberof\ events\ reconstructed\ within\ 3\sigma_{CT}}{Total\ numberof\ Events} \tag{5.1}$$

The reconstruction efficiency (η_{rec}) gives the total number of valid reconstructions (within both σ_{CT}^{slp} and σ_{CT}^{int}) out of the total number of events given in as the input to the model.
The reconstruction efficiency is studied as a function of the noise multiplicity for both the linear model and ANN the results are given in figure 5.3.

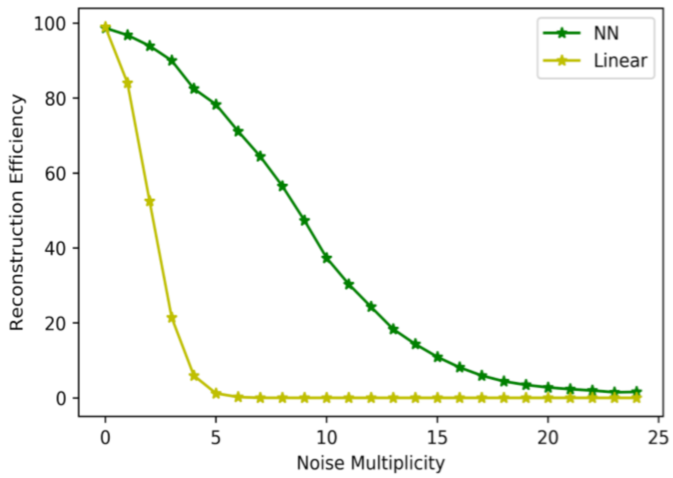


Fig. 5.3 Reconstruction efficiency(η_{rec}) for ANN and linear model as a function of noise multiplicity (M_N) for a 100% efficient detector generated for a total of 2.5 million events with equally distributed noises.

For noise multiplicity (M_N) = 0, both the linear fit model and the ANN model reconstructs almost all the events, both perform with the same efficiency. However when the noise multiplicity is increased to 1 the reconstruction efficiency (η_{rec}) drastically falls in the linear model to about 80% but the ANN reconstructs about 96% of the total events.

If the noise multiplicity is increased to 5, i.e when events are 20% noisy, the linear algorithm rejects almost all the events and none gets reconstructed, however the ANN on the other hand, reconstructs almost 80% of the events. The efficiency of the model is so high that the ANN 50% of the events are reconstructed even for a noise multiplicity (M_N) of 10.

The study is then made for 0.6 million events with detector efficiency(η) = 95%. The results are summarized in figure 5.4.

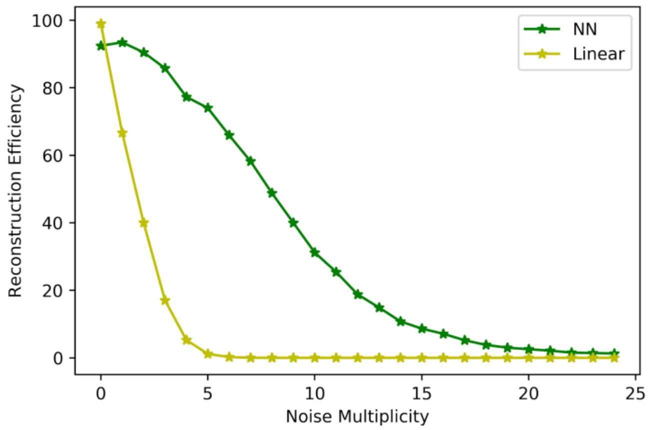


Fig. 5.4 Reconstruction efficiency(η_{rec}) as a function of noise multiplicity (M_N) for a 95% efficient detector generated for a total of 2.5 million events with equally distributed noises.

The plot closely follows the trend of the 100% efficient detector except for the small deviation at $M_N = 0$. This could be because of the way ANN has been trained. The ANN was trained only for 100% efficient detector, i.e, all the strips gets fired along the track, hence for a clean track the ANN looks for at-least 1 hit in each layer. For a detector efficiency of 95% not all the layers for a clean track will be fired (4.4). Therefore, some hits will be missing and ANN's prediction accuracy decreases. To validate the proposal on the deviation same study was carried out for a 90% efficient detector. The results are given in figure 5.5.

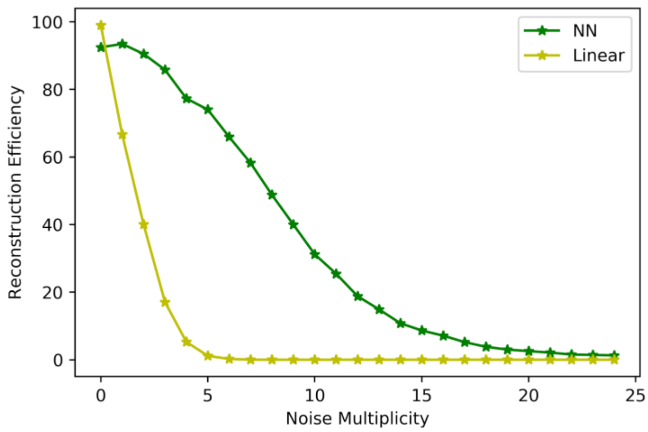
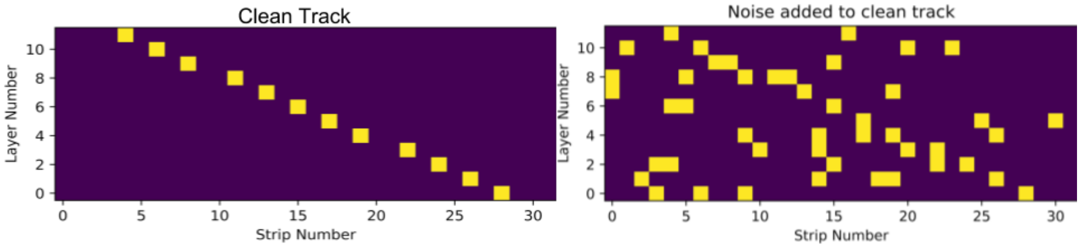
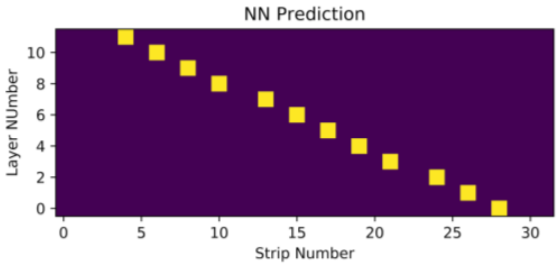


Fig. 5.5 Reconstruction efficiency(η_{rec}) as a function of noise multiplicity (M_N) for a 90% efficient detector generated for a total of 2.5 million events with equally distributed noises.

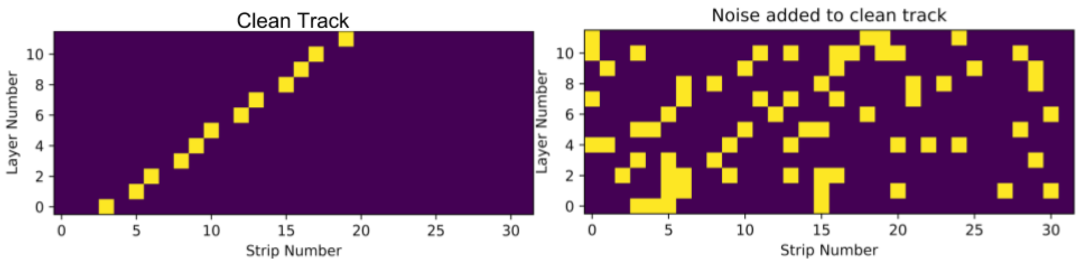
The plot of the reconstruction efficiency (η_{rec}) vs the noise multiplicity (M_N) for a 90% efficient detector also behaves the same way as that of 95% efficient detector (Fig:5.4). This is a conclusive evidence that the training must be improved by including various detector efficiency to address the problem.
The plots however reveal that the ANN algorithm is far more efficient than the conventional linear model. This serves as an evidence to implement ANN algorithm in the field of particle tracking.
Figures 5.6 to 5.9 are some of the sample reconstructions from the ANN model which are rejected by the linear model which are presented to illustrate the power of ANN in track reconstruction.

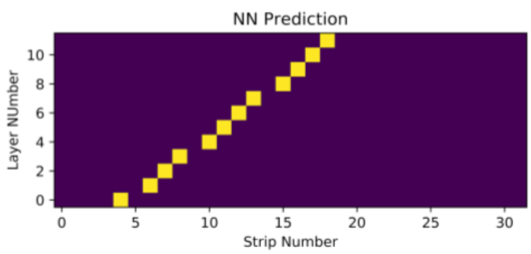




Parameter	Actual	NN prediction
Slope (strips/layer)	-2.2153	-2.1967
Intercept (layer)	28	28

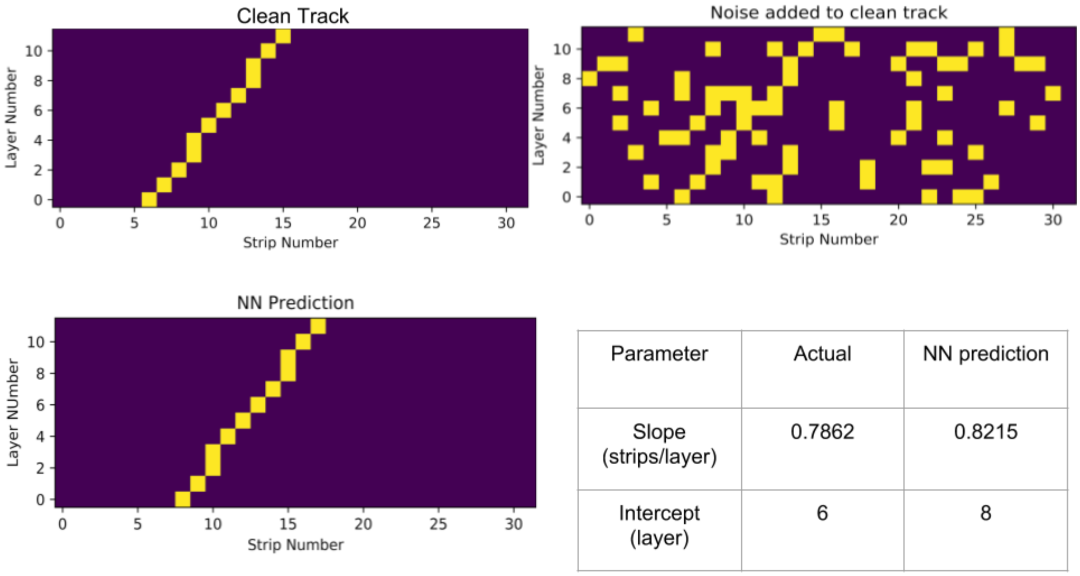
Fig. 5.6 Sample reconstruction of an event with noise multiplicity $M_N = 5$, $M_S = 2$, $\eta = 100\%$





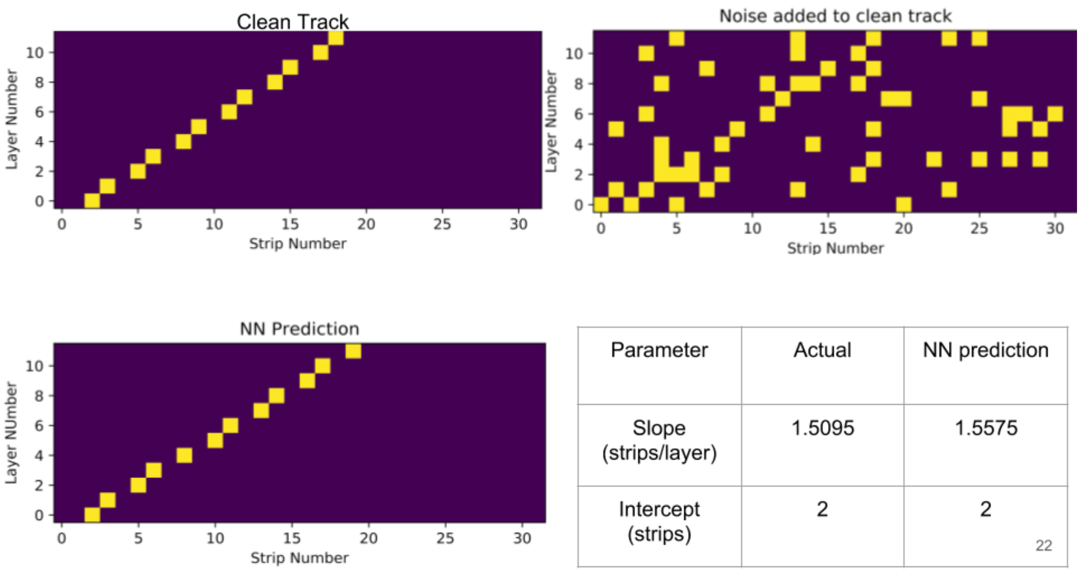
Parameter	Actual	NN prediction
Slope (strips/layer)	1.38203	1.2679
Intercept (layer)	3	4

Fig. 5.7 Sample reconstruction of an event with noise multiplicity $M_N = 9$, $M_S = 2$, $\eta = 100\%$



Parameter	Actual	NN prediction
Slope (strips/layer)	0.7862	0.8215
Intercept (layer)	6	8

Fig. 5.8 Sample reconstruction of an event with noise multiplicity $M_N = 7$, $M_S = 2$, $\eta = 100\%$



Parameter	Actual	NN prediction
Slope (strips/layer)	1.5095	1.5575
Intercept (strips)	2	2

Fig. 5.9 Sample reconstruction of an event with noise multiplicity $M_N = 6$, $M_S = 2$, $\eta = 100\%$

Chapter 6

Conclusion and future scope of the work

6.1 Conclusion

The conventional algorithm for the tracking of cosmic muon in the TIFR prototype stack was studied in detail and the efficiency of tracking was found to be poor in the present algorithm. ANN is a very successful algorithm in the field of pattern recognition and image analysis which was tried out in this project work and the results have been discussed in this project. ANN is a potential alternative to the conventional linear fit model. The reconstruction efficiency of the ANN is substantially higher and can still be improved by increased statistics. Few sample reconstructed events were shown for illustrating the potential of the ANN.
Some of the results are summarized here:

6.2 Future scope

Further intended works and future scope of the work include the following:

1. To address the problem of the small deviation in prediction of tracks for the zero multiplicity case in 95% and 90% efficient detectors by training the ANN with events having various detector efficiency (η).

2. To develop an algorithm to predict the track of the muons in partially contained events.

3. To develop an algorithm to predict curved tracks, because the main ICAL to be set-up at the INO will operate with magnets which is used to estimate the energy and to distinguish between the particle, anti-particle interactions.

4. This model could be used in the track identification at the INO-ICAL detector for the tracking of the neutrino induced events.

5. The potential application of the model is in surface neutrino detection, where the problem is to identify the track induced by a neutrino event in a sea of cosmic muonic tracks.

   We hope that this work will be a stepping stone for the simulation team of the INO collaborating in seeking alternate emerging technologies for efficient track reconstruction.

References

[1] Samuel, D., Onikeri, P.B. and Murgod, L.P., 2017. Angular resolution of stacked resistive plate chambers. Journal of Cosmology and Astroparticle Physics, 2017(01), p.058.

[2] Majumder, G., Mondal, N.K., Pal, S., Samuel, D. and Satyanarayana, B., 2014. Study of the directionality of cosmic muons using the INO-ICAL prototype detector. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 735, pp.88-93.

[3] Majumder, G., Mohammed, S., Mondal, N.K., Pal, S., Samuel, D. and Satyanarayana, B., 2012. Velocity measurement of cosmic muons using the India-based Neutrino Observatory prototype detector. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 661, pp.S77-S81.

[4] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.