# Artificial Neural Network (ANN) Based Machine Learning Model for Accelerator Systems


***By***

***Surendra Yadav***

***BDS, RRCAT, Indore***


***School on "Scientific Computing, Artificial Intelligence, and Machine Learning" , IUAC New Delhi on 24th to 27th July, 2023***

# Outline

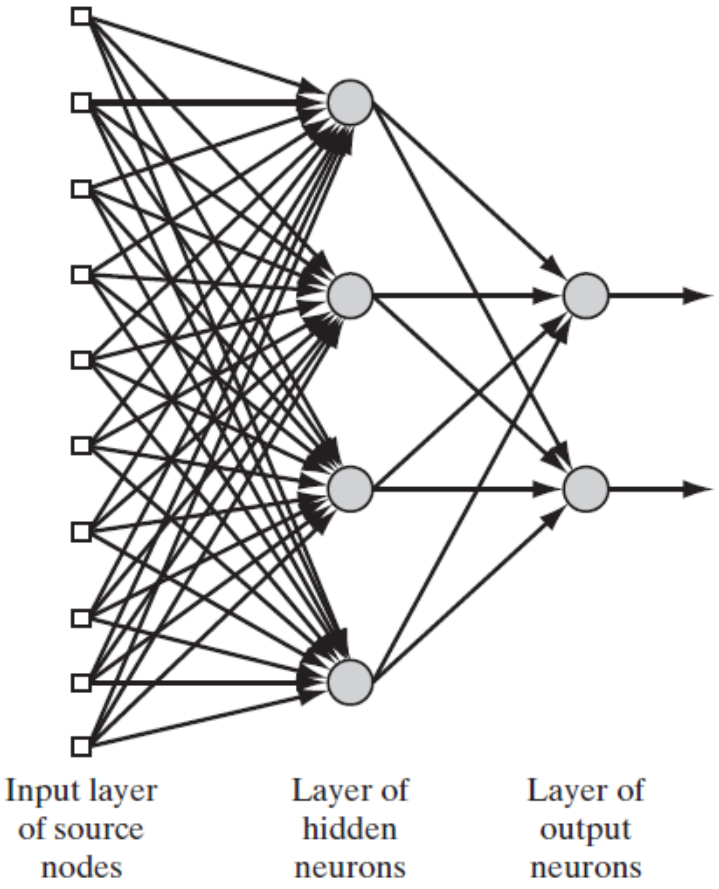
- ***Introduction***
- ***Artificial neural network (ANN)***
- ***Applications of AI for accelerator***
- ***ANN based applications in Indus-2 SRS***
  - ***Fault identification of BPM sensor and measurement of beam position with three electrode's signals***
  - ***Minimization of transverse beam instabilities in Indus-2 with betatron tune and CBM measurement system***
- ***Summary***

# Introduction

- **The operation of the complex machine like accelerators is challenging due to handling of the following,**
  - **Nonlinearities**
  - **Uncertainties**
  - **Multiple coupled parameters etc.**
- **Achieving the real machine model is deviated from mathematical model or some time model is also not known.**
- **Hence, data driven modeling is required which uses the AI techniques.**
- **ANN is the most suitable technique for this purpose.**

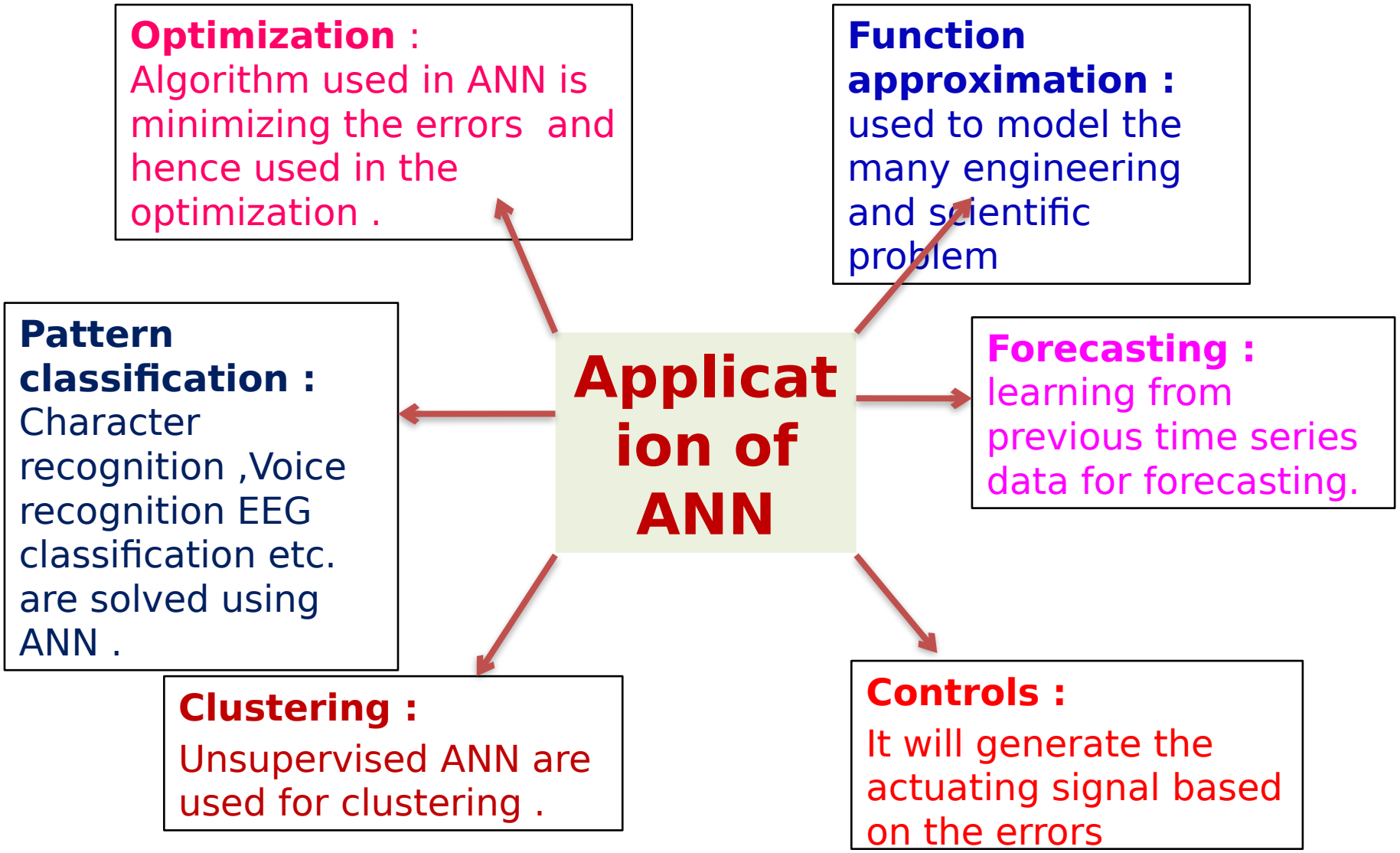
# Introduction to ANN

- ANN is inspired by biological neural network.
- ANN are massively parallel computing systems consisting of an extremely large number of simple processors with many interconnection.[1]
- A general structure of ANN are shown below



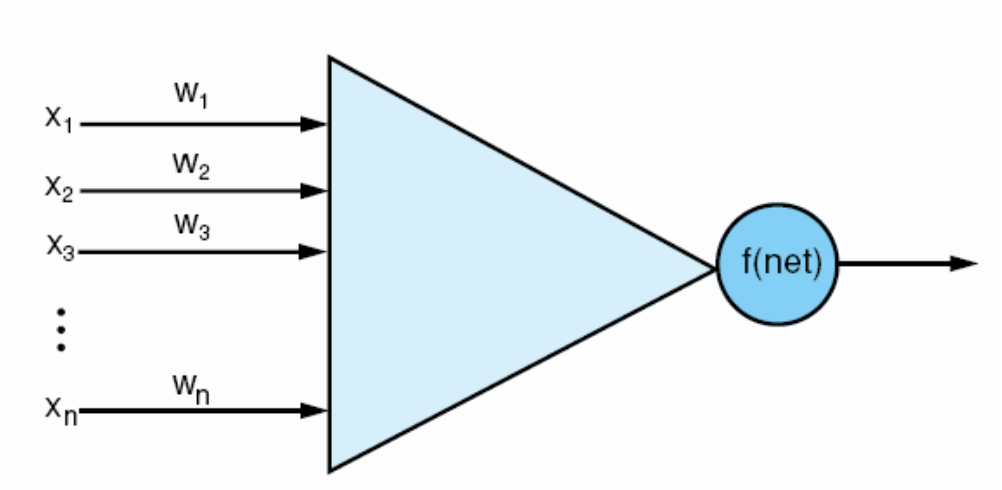
# Artificial Neural Networks

- 1943: McCulloch and Pitts ✉model of a neuron --> Perceptron
- 1960s: Widrow and Hoff ✉ explored Perceptron networks (which they called “Adelines”) and the delta rule.
- 1962: Rosenblatt ✉ Convergence of the perceptron training rule.
- 1969: Minsky and Papert showed that the Perceptron cannot deal with nonlinearly-separable data sets---even those that represent simple function such as X-OR.
- 1970-1985: Very little research on Neural Nets.
- 1986: Invention of Backpropagation [Rumelhart and McClelland, but also Parker and earlier on: Werbos] which can learn from nonlinearly-separable data sets.


Optimization :
Algorithm used in ANN is minimizing the errors and hence used in the optimization .
Function approximation :
used to model the many engineering and scientific problem
Pattern classification :
Character recognition ,Voice recognition EEG classification etc. are solved using ANN .
Application of ANN
Forecasting :
learning from previous time series data for forecasting.
Clustering :
Unsupervised ANN are used for clustering .
Controls :
It will generate the actuating signal based on the errors

# An Artificial Neuron

- A neural network is a collection of artificial neurons.[3]
- The neuron responds to input coming from x1, x2, …, xn.
- The neuron computes its output value, denoted here as f(net).




•The computation for f(net) takes the values of the inputs and multiplies each input by its corresponding weight

x1*w1 + x2*w2 + … + xn*wn

Different types of neurons will use different activation functions with the simplest being

- if x1*w1 + x2*w2 + … + xn*wn >= t then f(net) = 1 else f(net) = -1

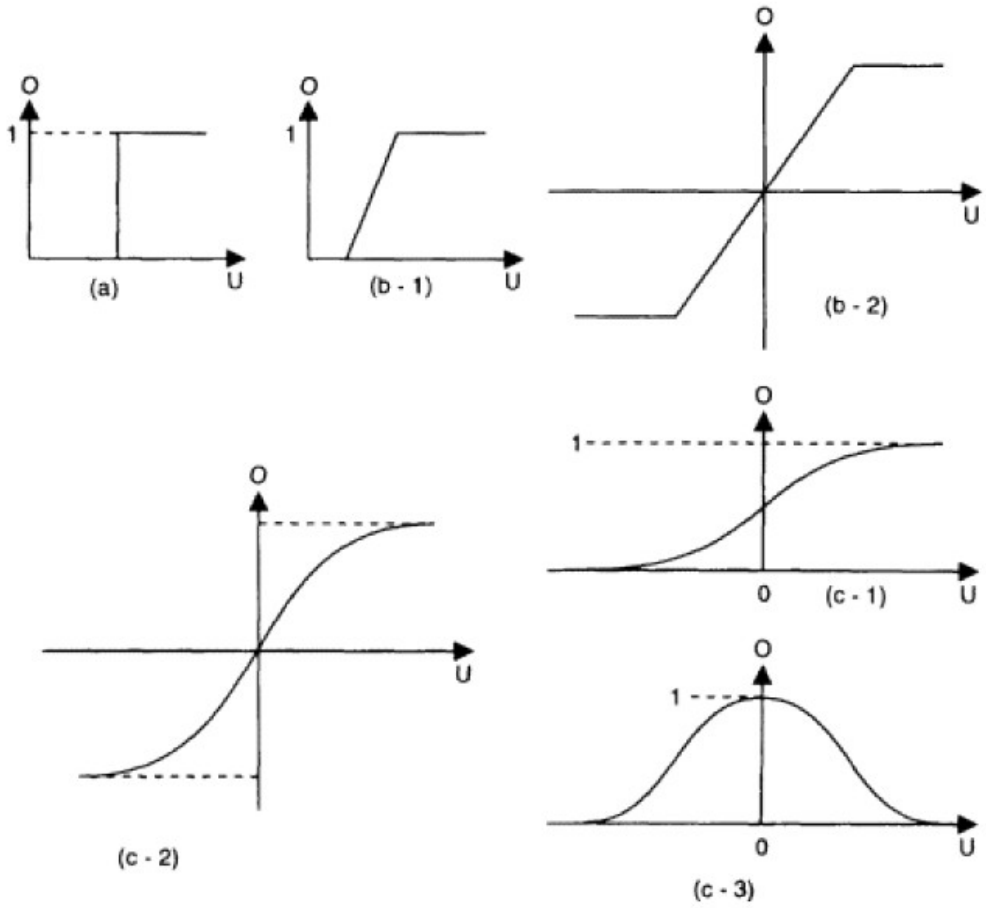
# Activation Functions

The most used activation functions shown in right side :

(a) Hard-limited threshold

(b) linear threshold: if the input is above a certain threshold, the output becomes saturated (to a value of 1); there are different variants of this function depending on the range of the neuronal output values shown in (b-1) and(b-2);

(c) sigmoid function: logistic function (c-1); bipolar logistic function (c-2); (c-3) Gaussian (bell shape) function.[5]



# Type of neural networks

- **Feed-forward networks**:
  - In this N/w the data flow from input to output units is strictly feed-forward.
  - The data processing can extend over multiple layers of units, but no feedback connections or connections between units of the same layer are present.
  - Example: Single layer perceptron ,multilayer perceptron, radial basis function nets .

- **Recurrent networks:**
  - It contain feedback connections.
  - In some cases, the activation values of the units undergo a relaxation process such that the network will evolve to a stable state in which these activations do not change anymore.
  - Example : Competitive networks, Kohonen's SOM, Hopfield network, ART model .

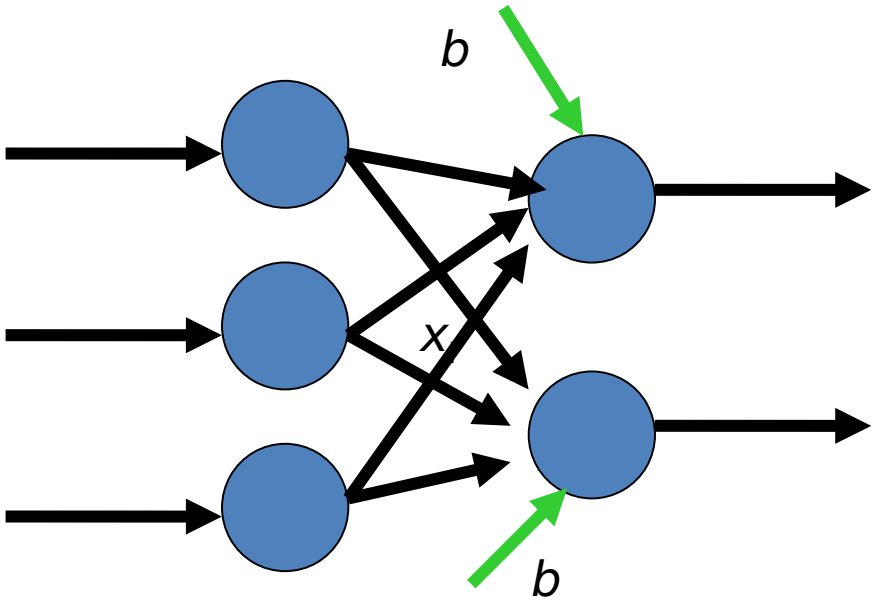
# Continued….

- **Supervised learning**:
  - In which the network is trained by providing it with input and matching output patterns.
  - These input-output pairs are usually provided by an external dataset.

- **Unsupervised learning** :
  - In which an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population.
  - Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.

# Perceptron

- A single layer feed-forward network consists of one or more output neurons, each of which is connected with a weighting factor $w_{ij}$ to all of the inputs $x_i$.




$$s_j = \sum_{i=0}^{n} w_{ij} x_i + b_j$$

# Learning in Perceptron

- The weights of the neural networks are modified during the learning phase

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}$$

$$b_{ij}(t+1) = b_{ij}(t) + \Delta b_{ij}$$

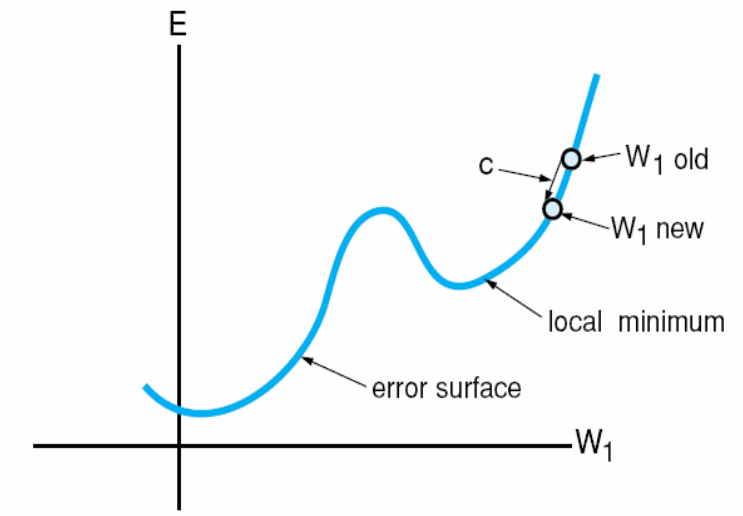
Process:

- Start with random weights
- Select an input couple *(**x**, f(**net**))*
- if $y \neq f(\mathbf{net})$ then modify the weight according with $\Delta w_{ij} = f(net)x_i$

- Note: The weights are not modified if the network gives the correct answer

# Gradient Descent Learning

- Gradient  decent is a learning algorithm that will move the edge weights closer and closer to that optimal location. The idea is to minimize the error by correcting the edge weight with help of the error in output.
- For a perceptron, the best set of values for the edge weights are identified after enough training iterations
- For other forms of neural networks, training might cause the edge weights to descend to a *local* minima



# Delta Rule

- Many of the training algorithms will take a partial differential of the summation value used to compute activation.

- It requires a sigmoid function for activation because the linear threshold function is not a continuous function.

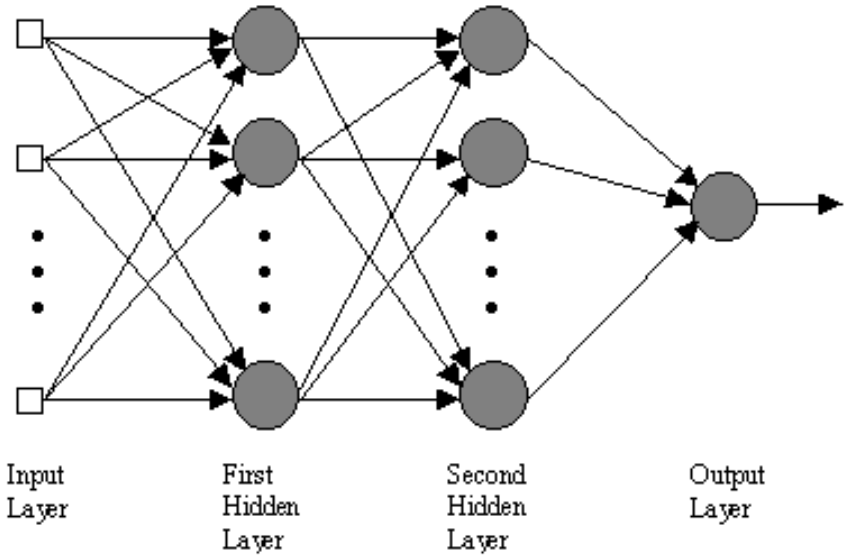
- weight adjustment for the edges into perceptron node i is

$$c * (d_i - O_i) * f'(net_i)\ x_j$$

  - c is the constant training rate of adjustment
  - $d_i$ is the value we expect out of the perceptron
  - $O_i$ is the actual output of the perceptron
  - f is the threshold function so f' is its partial derivative
  - $x_j$ is the jth input into the perceptron

# Feed-Forward Back-Propagation Network

- The layers of perceptron is completely connected to the next layer and the preceding layer.
- The training of network is performed using an algorithm called back-propagation.
- Unlike a perceptron network, all of the edge weights in this network can be trained because the back-prop algorithm is more powerful.

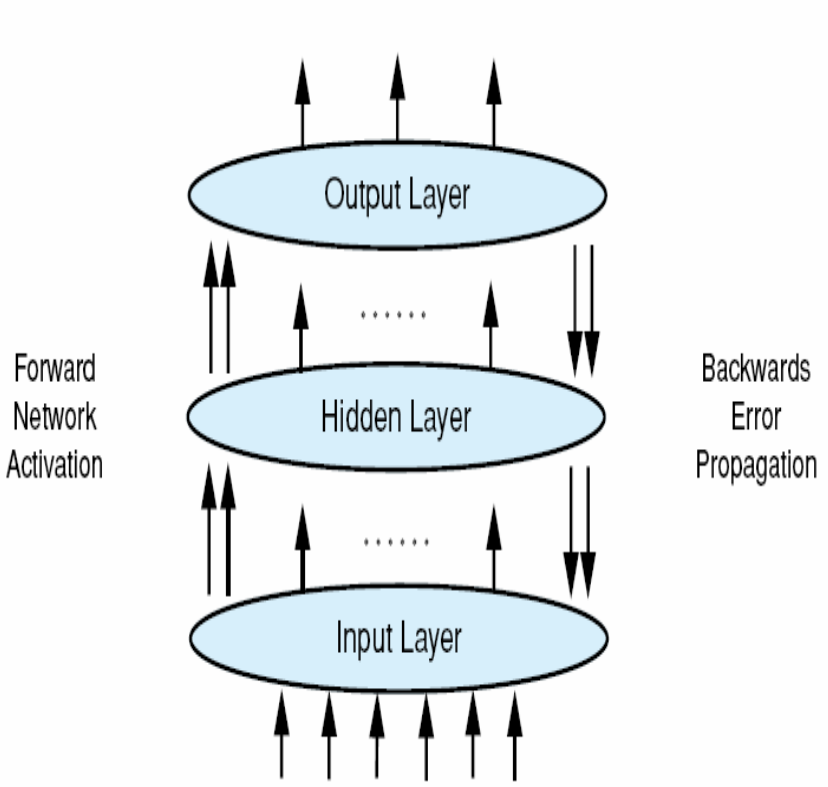



***The output node(s) will deliver a real value between 0 and 1 but not exactly 0 or 1. Hence multiclass problem may be solved with ANN structure.***

# Training

For each item in the training set

- Compute the activation value for each node in the first hidden layer
- Pass those values forward until you reach the output layer
- compute what the errors
- back propagate the error to the previous level, adjusting weights
- continue to back propagate




One iteration through the entire training set is called an epoch.

The training time is deeply affected by initial conditions, Neurons and

# Machine Learning using ANN

- Machine learning is a subset of AI that focuses on algorithms and statistical models that enable computers to learn from and make predictions or decisions based on data without being explicitly programmed.

- The model can be different type depends on the requirements.

- Most common models are used for Classification and regression purposes.

- ANN based regression model is also known as universal approximate.

# Continued….

- Deep learning is a specialized subfield of machine learning that uses artificial neural networks with multiple layers (deep architectures) to learn and represent complex patterns and features from data.
- However, it deals with the raw data rather than features as required in ML.
- It requires large number of data for

# Continued….

- Machine learning is used in applications, including spam detection, recommendation systems, credit scoring, and regression problems.

- Deep learning has shown remarkable performance in image and speech recognition, natural language processing, autonomous vehicles, and playing complex games like Chess.

- Machine learning algorithms rely on gradient-based optimization techniques to update model parameters during training.

- Deep learning models often use more advanced optimization algorithms, such as stochastic gradient

# AI APPLICATIONs IN ACCELERATOR

- Accelerator researches are frontier in the field of scientific research.
- Many scientific computation, modeling, control, optimization are required in this field.
- Many times, we have only inputs, outputs and some in between states.
- With this, regression model between input and output cannot be modeled using traditional Methods.

# Use of AI for Accelerator

**Accelerator components design for magnets, radio frequency (RF) , Beam diagnostics Systems .**

**Lattice design like optimization searches for optimal set points, optimum field amplitudes, optimum element placement etc.**

**AI Techniques**

(Artificial Neural Network (ANN), Genetic Algorithm (GA) , Swarm intelligence , fuzzy logic etc.)

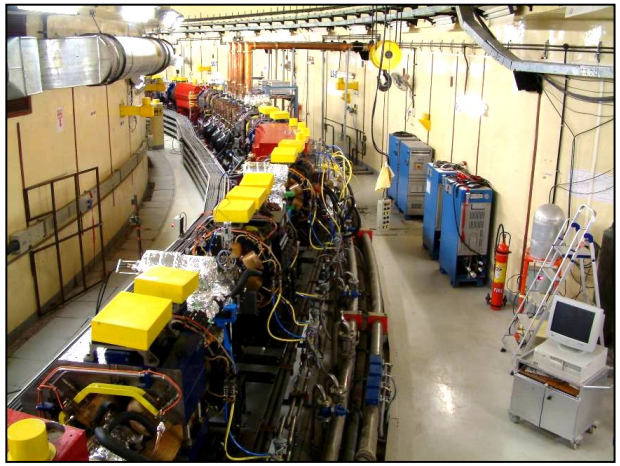
**Operator support system for tuning and optimization of machine**

# AI applications at Indus Accelerator

- Modified Genetic Algorithm based online optimization of beam transmission for Indus-2 beam injection.
- Fuzzy Control System for Controlled beam excitation during beam parameter measurement.
- ANN based fault identification of BPM sensor.
- ANN based regression model for Beam instability and control scheme.

# Indus-2 Synchrotron Radiation Source

- Indus-2, an indigenously designed and built synchrotron radiation source, is presently operating in round the clock mode at beam current up to 200 mA at 2.5 GeV energy.

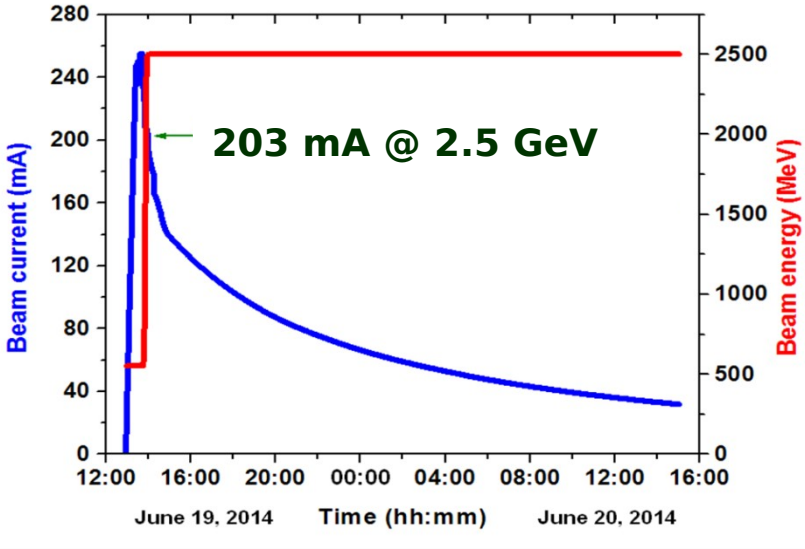


- Important milestones

**2.5 GeV, 100 mA : Dec. 2011**

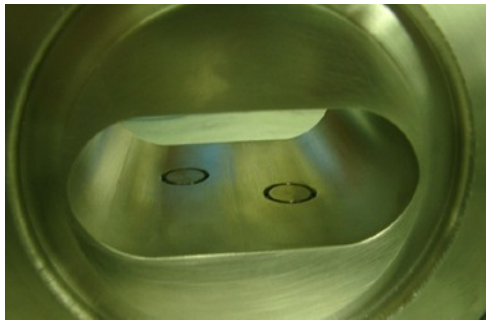
2.5 GeV, 150 mA : Jan. 2013

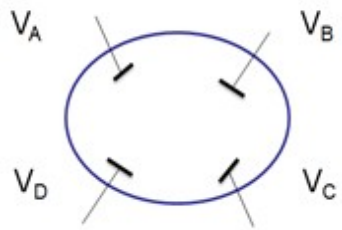
**2.5 GeV, 200 mA : June 2014**



# Fault Identification of BPM Sensor using ANN

**In the conventional method, signals of all the four electrodes are used in the position calculation algorithm for beam position measurement, instability measurement and betatron tune measurement .**






$$x = k_x \frac{\Delta x}{\Sigma} = \frac{(V_A + V_D - V_B - V_C)}{(V_A + V_B + V_C + V_D)}$$

$$z = k_z \frac{\Delta z}{\Sigma} = \frac{(V_A + V_B - V_C - V_D)}{(V_A + V_B + V_C + V_D)}$$

**Aim is to measure the beam position using three electrodes signals and also monitor consistency of beam position monitors .**



**Surendra Yadav et al., Intelligent approach for three electrode based beam position calculation in Indus-2 , National Conference on Reliability and Safety Engineering (NCRS-2016), December, 2016**

# Fault Detection of BPM

**The set of four beam position data using four combinations of three electrode signal are compared for consistency check.**

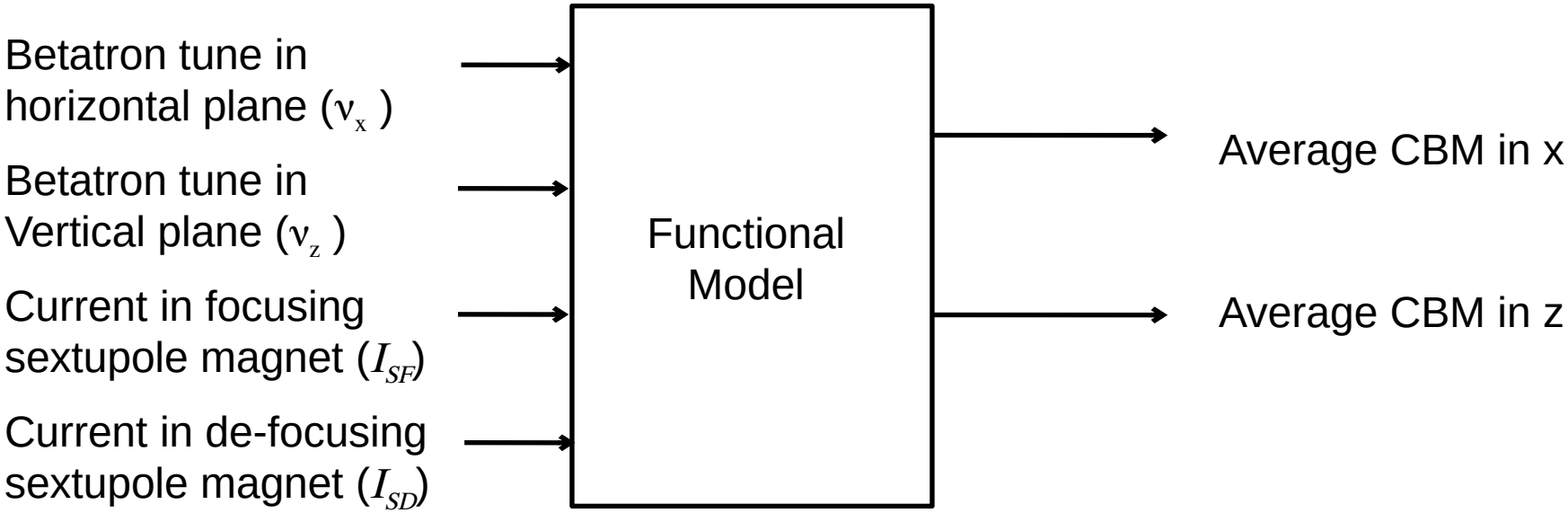
**Beam positions [x1, x2, x3, x4 ] and [z1, z2, z3, z4] are obtained from the electrode voltages [$V_A$,$V_B$,$V_C$,$V_D$]using neural network. The consistency of the signal will be identified as follows**

**The maximum error signal is calculated as**

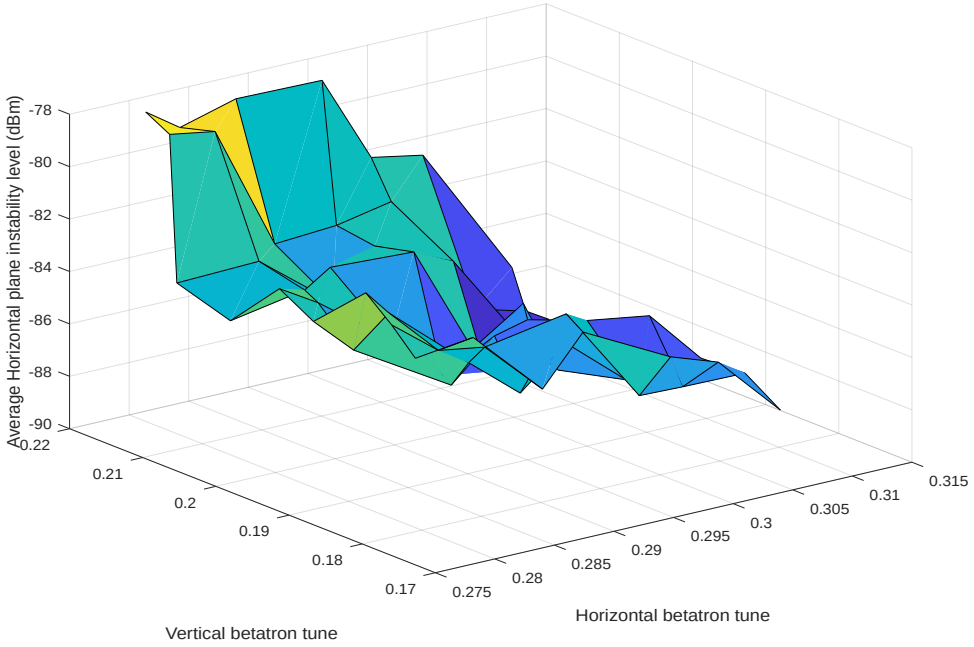
**The consistency level$I_{cns}$is defined as**

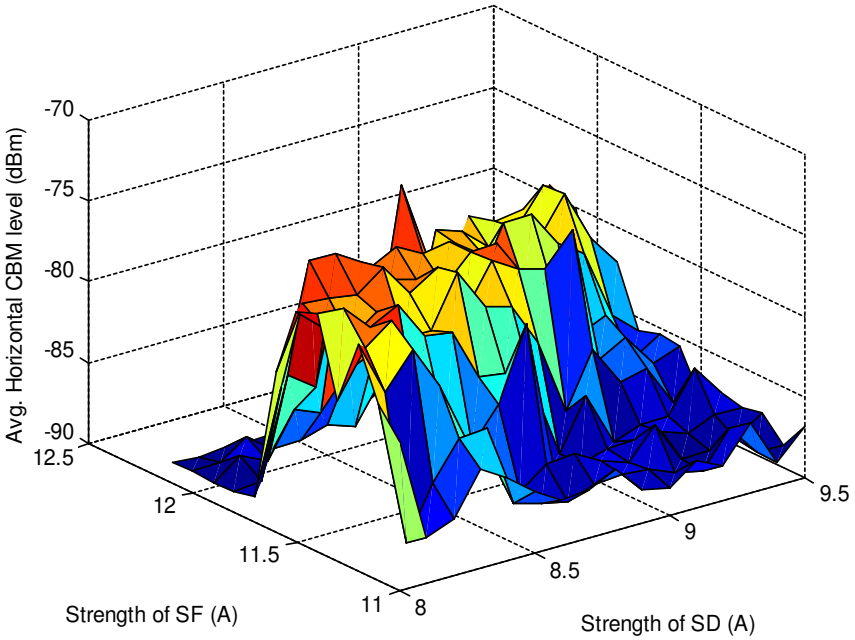
**Where *Pth* is the position threshold beyond that the beam position monitor is considered as malfunctioning. In Indus-2, it is considered as 300 microns.**

# Minimization of Transverse Beam Instabilities



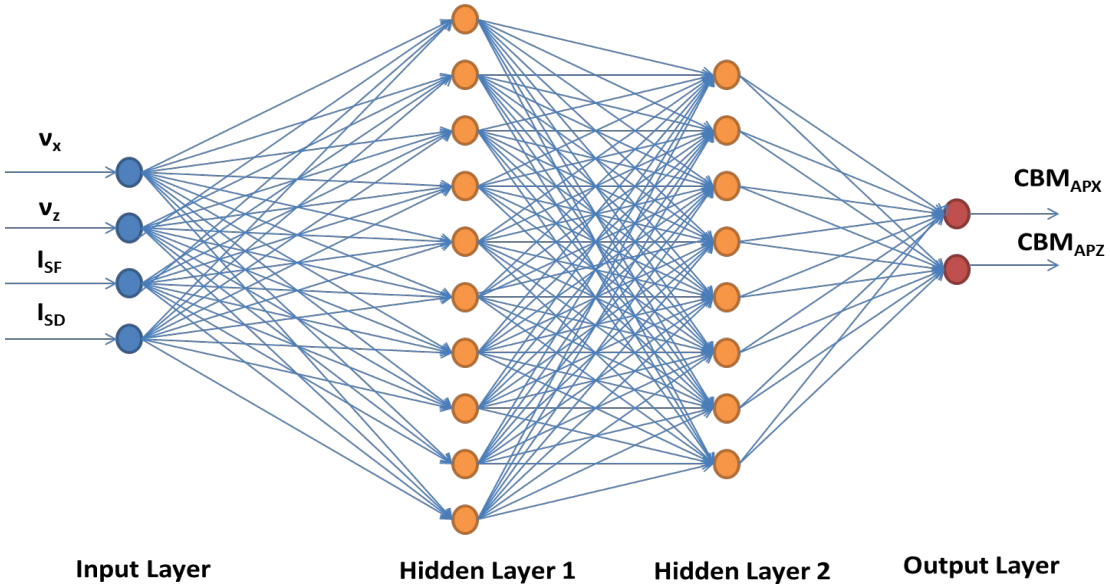
# Datasets Generation of TCBM Levels




- Nonlinear dependency
- Multiple minima and maxima
- Time variant nature

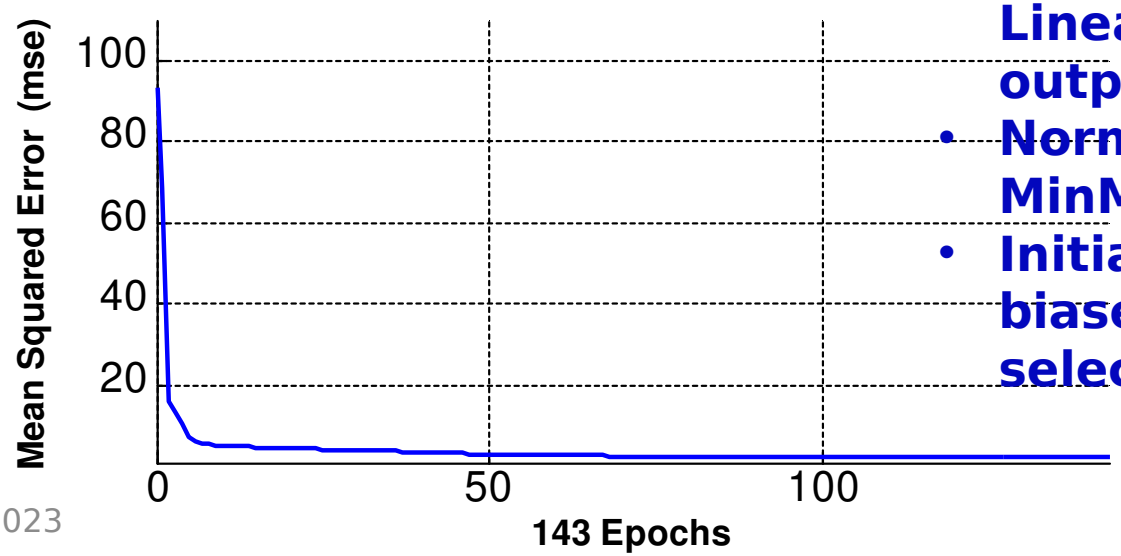
# ANN Based Model of TCBM level



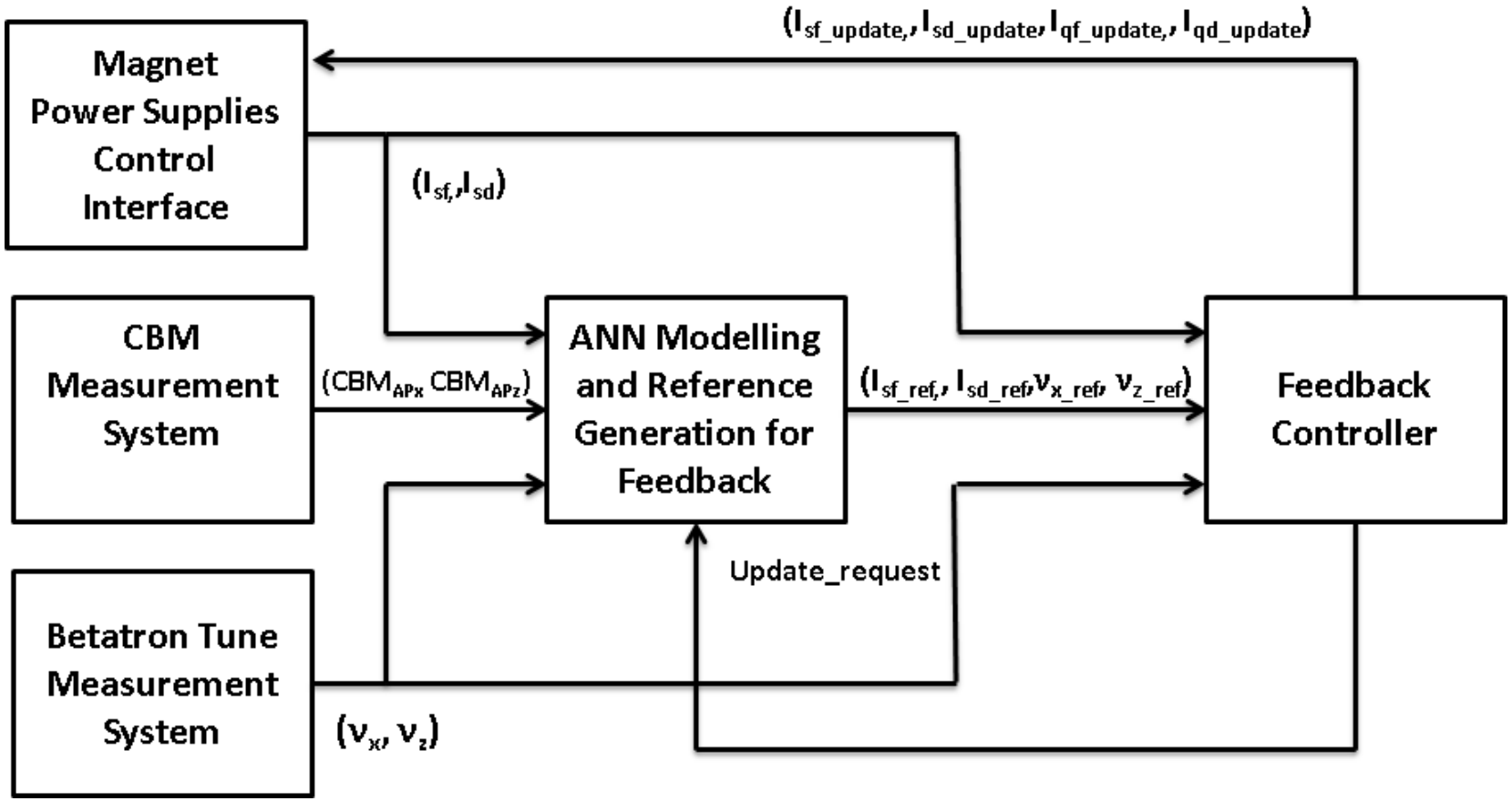

**ANN model has following features:**

- **No. of inputs: 4,**
- **No. of outputs: 2,**
- **No. of hidden layers: 2,**
- **Training algorithm: Mean Squared Error Minimization Algorithm,**
- **Activation function: Sigmoidal function in hidden layers and Linear function in output layer,**
- **Normalization : MinMaxScaler,**
- **Initial weight and biases: Randomly selected.**

Training Curve :



# Implementation of ANN Guided Feedback System




**($\nu_x$, $\nu_z$) : Operating betatron tune value**

**($CBM_{APx}$, $CBM_{APz}$): Average TCBM levels**

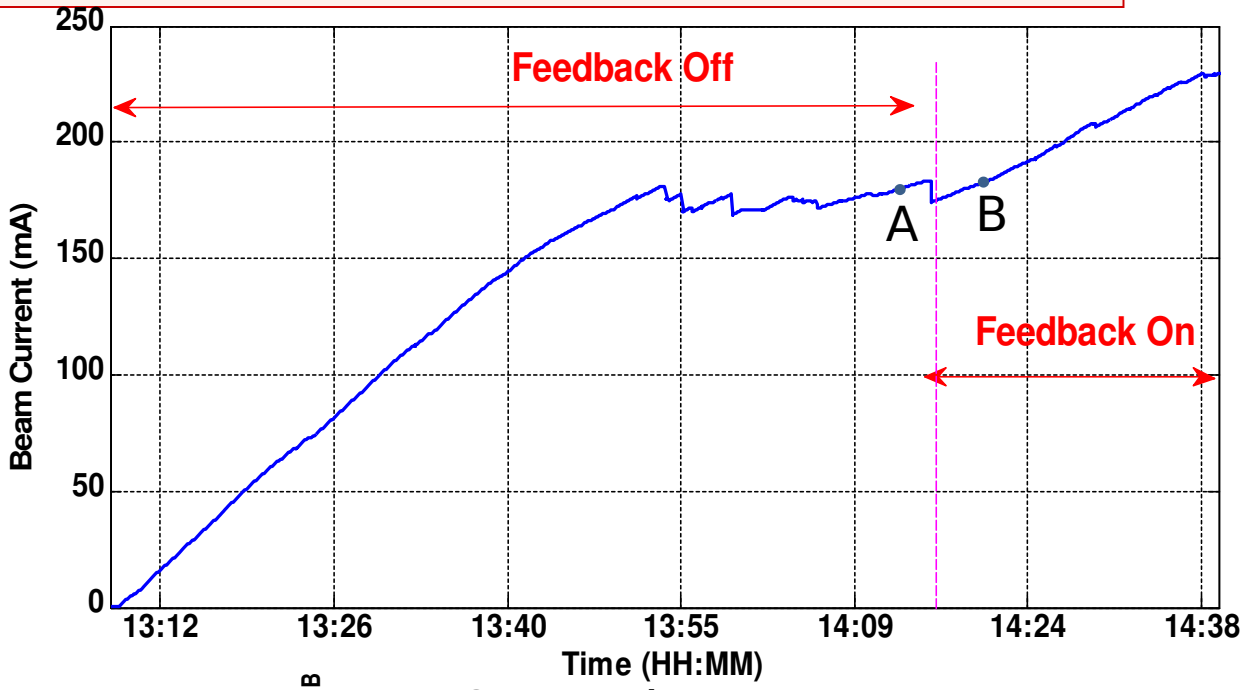
**($I_{sf\_ref}$, $I_{sd\_ref}$, $\nu_{x\_ref}$, $\nu_{z\_ref}$) : Reference settings for the feedback controller**

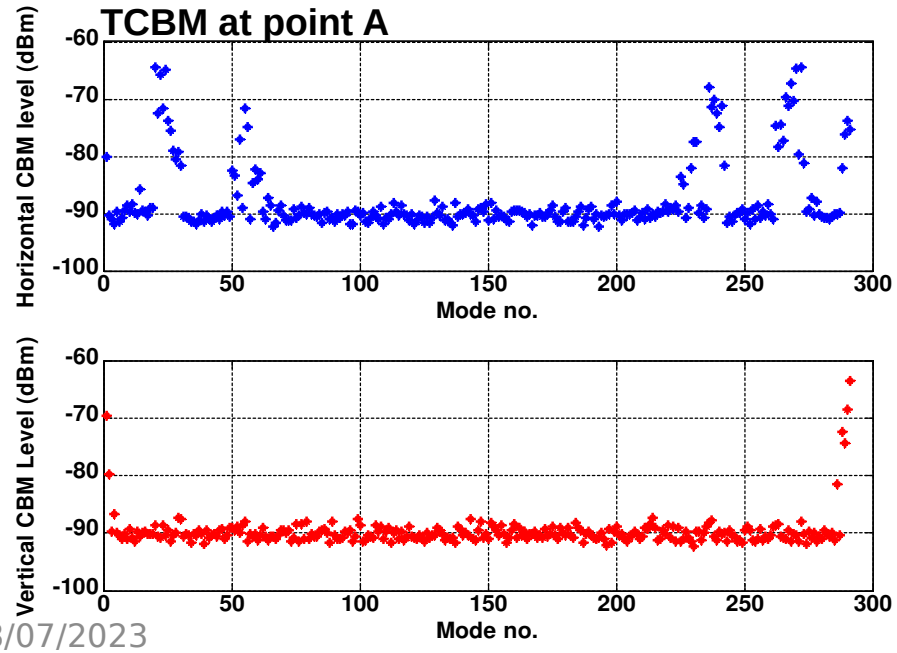
**Surendra Yadav et al., Review of Scientific Instruments, 2021, 92, 033304.**

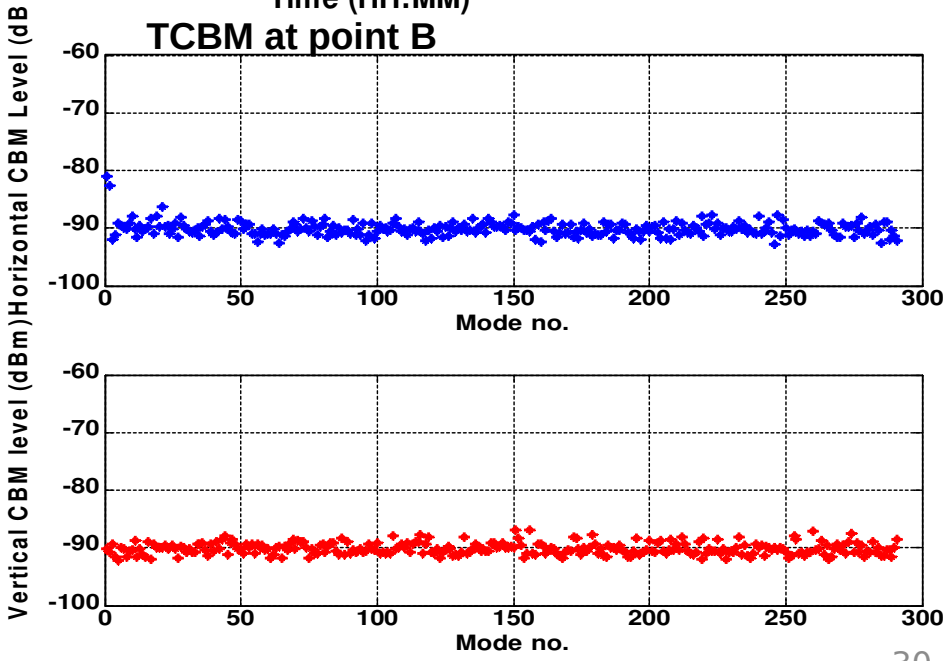
# Beam Experiments and Results

- Beam energy : 546 MeV,
- Operating Betatron tune : [0.303 0.193]

- Chromaticity: [0.4 0.9] with $I_{sf}$=11.7 A and $I_{sd}$=8.24 A.



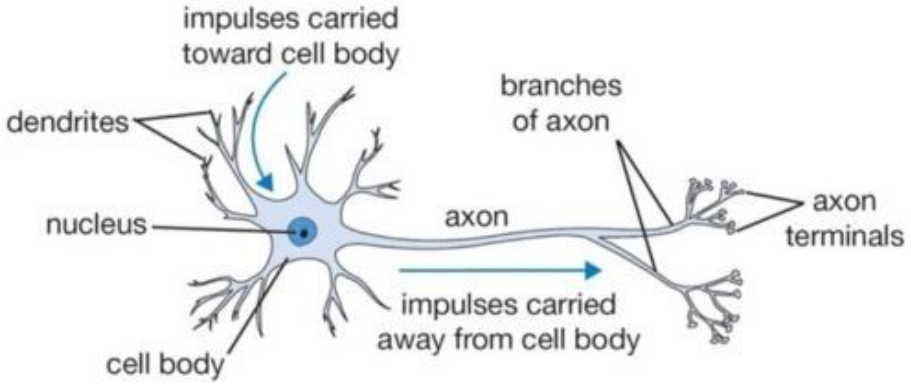
# Summary

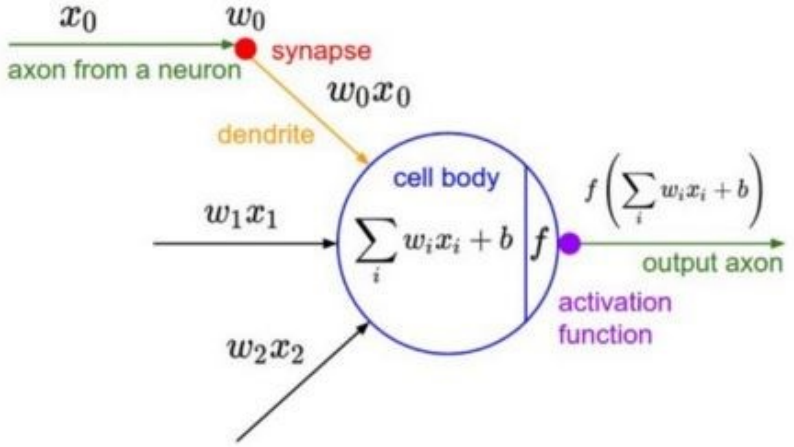
- Performance of Indus Accelerators has increased using AI technique.
- Operator support system has been developed with AI for auto optimization the routine operation.
- CNN based fault diagnostics system is being deployed.

# Thank You

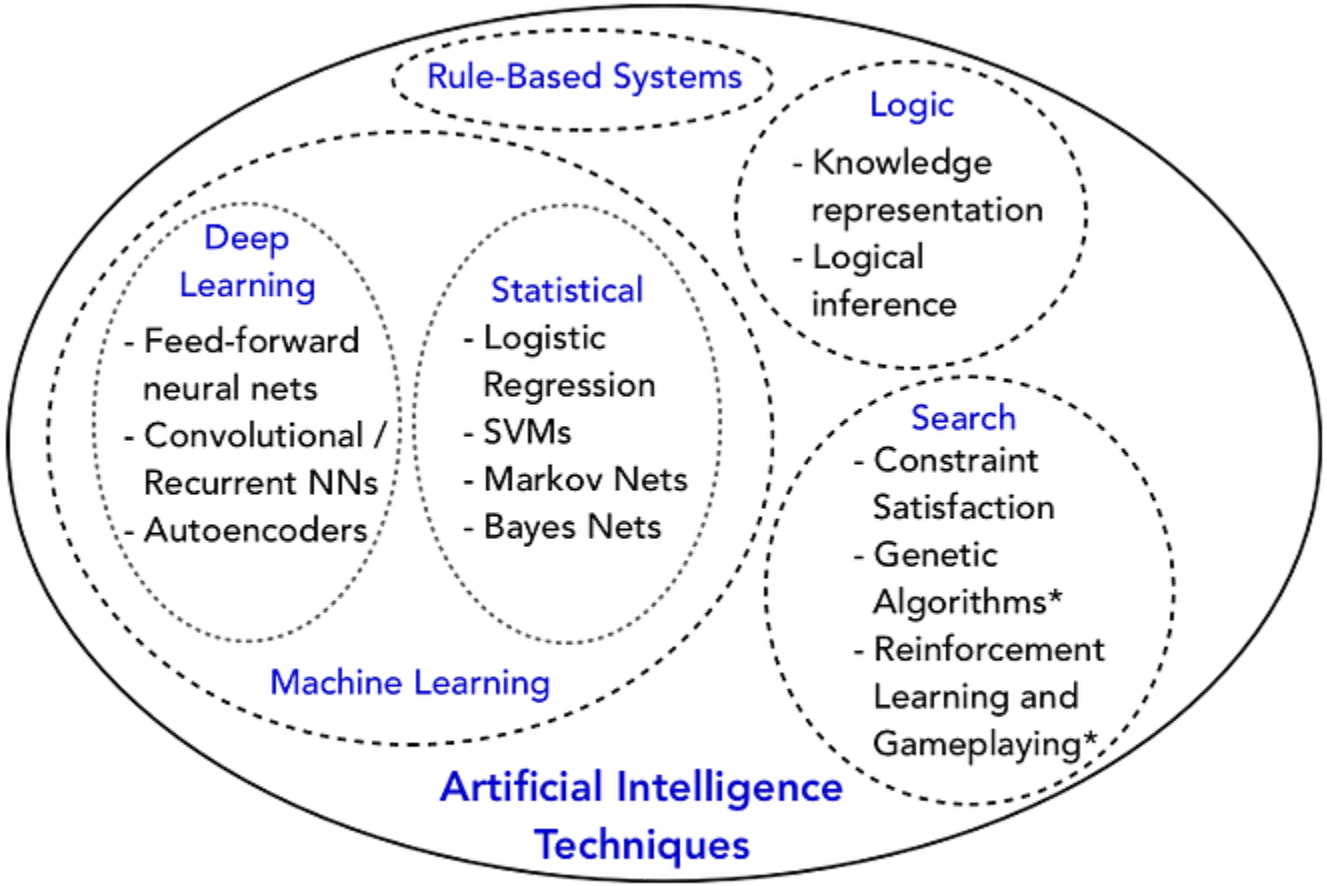
# Inspiration from the Brain

- ANN are *inspired* by the structure of neurons in the brain.




Credits : Stanford Course




Ref: https://medium.com/@EloquentLabs/mapping-the-ai-landscape-21d7673522c5

# Competitive Learning

- A “winner-take-all” competitive form of learning can be applied to FF networks without using the reinforcement step of backprops.

- when an example is first introduced, the output node with the highest value is selected as a “winner”

- edge weights from node i to this output node are adjusted by $c*(x_i - w_i)$
  - c is our training constant
  - $x_i$ is the value of input node i
  - $w_i$ is the previous edge weight from node i to this node

- the common application for this learning algorithm is to build self-organizing networks (or maps), often called Kohonen networks.

# Recurrent Networks

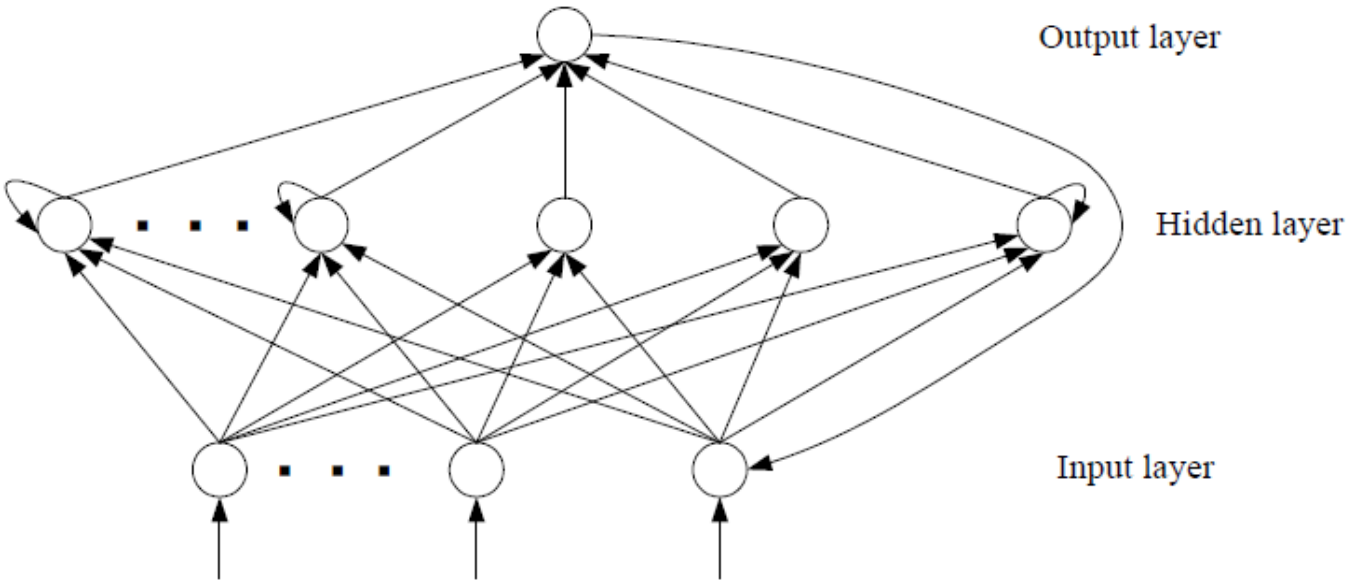
- One problem with NNs as presented so far is that the input represents a “snapshot” of a situation. Dynamic condition is difficult for ANN structure so far.

- In a recurrent network, Ordinary multi-layered FF/BP network is used and wrap the output nodes into some of (or all of) the input nodes.

- In this way, some of the input nodes represent “the last state” and other input nodes represent “the input for the new state”.

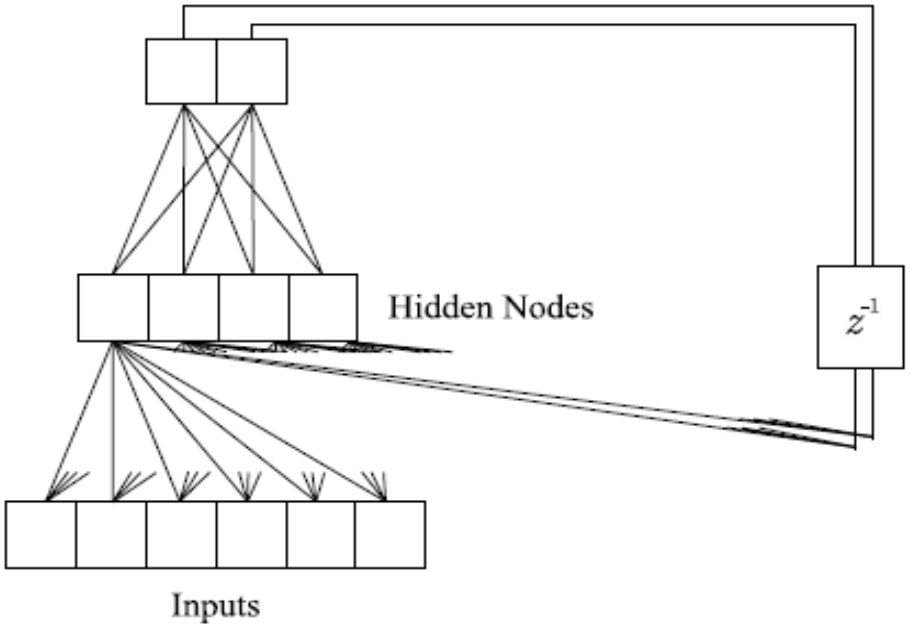
- Recurrent networks are a good deal more complex than ordinary multi-layered networks and so training them is more challenging.

# Examples




Above, the recurrence takes the single output value and feed it into a single input node

To the right, the outputs are fed into hidden layer nodes instead of input nodes



# Advantage of NNs

- Through training, the NN learns to solve a problem without the need for a lot of programming.
- Capable of solving low level recognition problems where knowledge is not readily available
- Able to handle fuzziness and ambiguity
- Uses distributed representations for graceful degradation
- Capable of supervised & unsupervised learning