Application of Machine Learning Techniques to Separate Resonance Particles From Background Processes

Shivam Raj

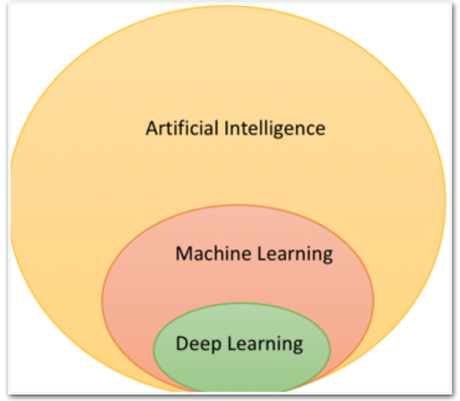
9th Semester Presentation

Supervised by: - Dr. Prolay Kr. Mal

Outline

- What is Machine Learning?
- What is Deep Neural Network (DNN)?
- How DNN works?
- Signal & Background Processes
- Variables used in training and testing
- Correlation Plots
- Results and outputs from the DNN
- Conclusion and Future Work

Machine Learning



Learning is a process by which performance of system gets improved from experiences

Machine learning(ML) is a part of artificial intelligence.

ML allows softwares to become accurate without being explicitly programmed

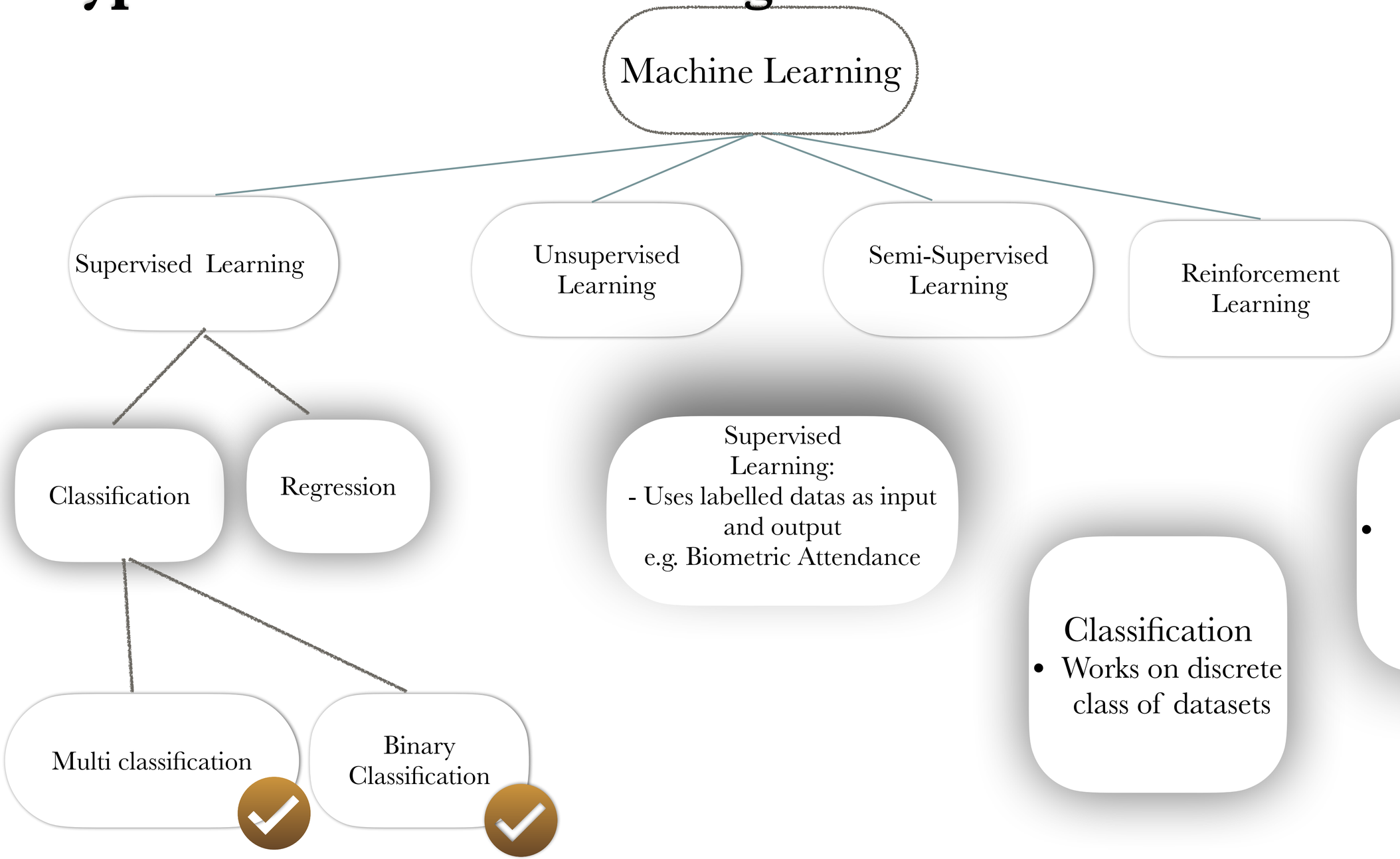
edureka!

Example of ML:-

1. siri, Alexa: virtual personal assistance
2. All social media(use GNN)
3. Email classification as spam or not spam

source: Edureka

Types of machine Learning



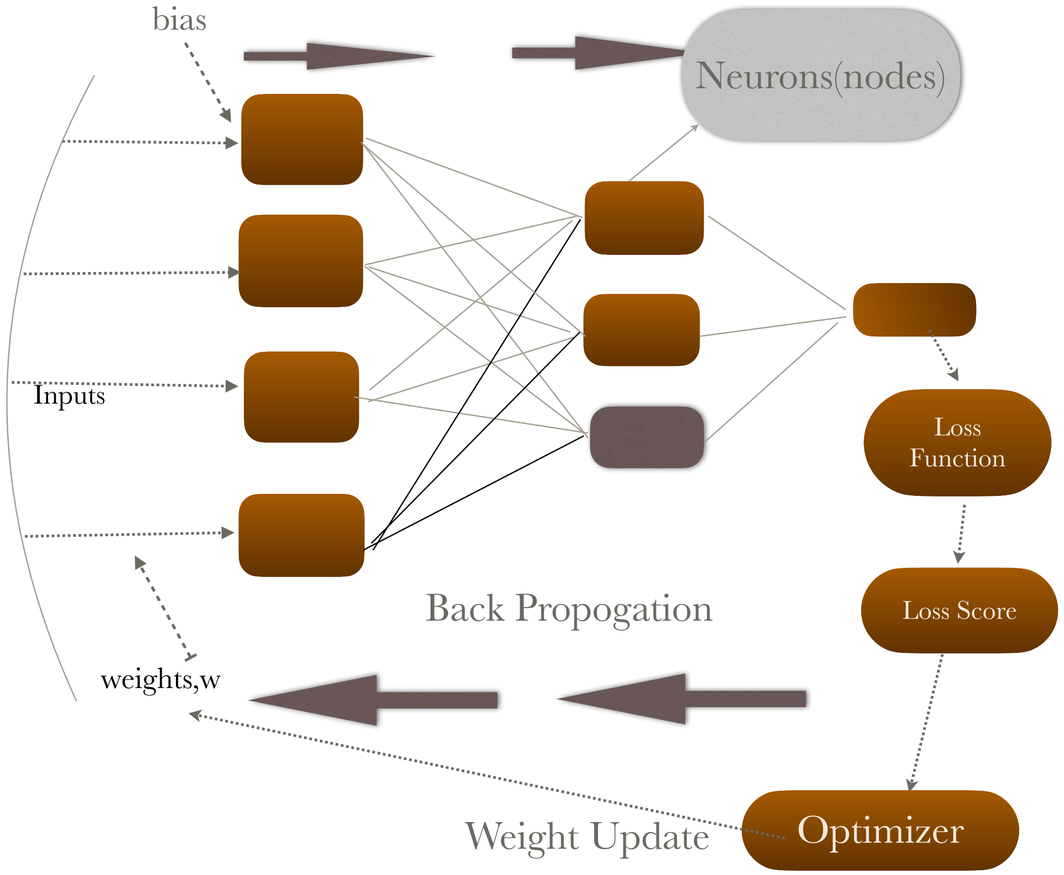
Supervised Learning:
- Uses labelled datas as input and output
e.g. Biometric Attendance

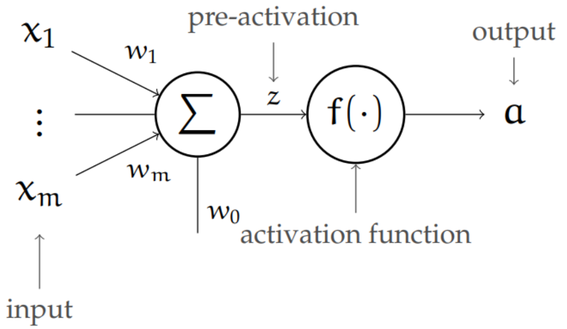
Classification
- Works on discrete class of datasets

Regression
- Works on continuous dataset.

Deep Neural Networks

It can be conceived as part of ANN, with similar structure of the human brain. Where neurons can be compared to nodes and connection between nodes as synapses.





$$a = f(z) = f(\sum_{j=1}^{j=m} x_j w_j + w_0) = f(w^T x + w_0)$$

Bias

- To delay the triggering of the activation function

- The output depends on the type of activation function used.

Single network(a input, hidden, and a output) is also known as perceptron, which only works for binary classification task.

- Deep means hidden layer >2.

Low level framework High level framework

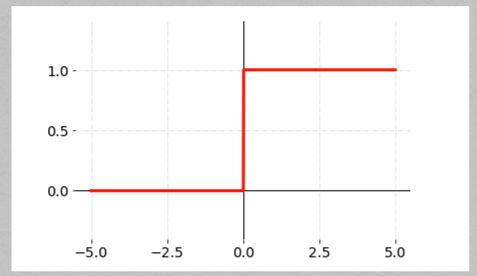
Keras

DNN:Activation function

Three types of Activation Functions:-

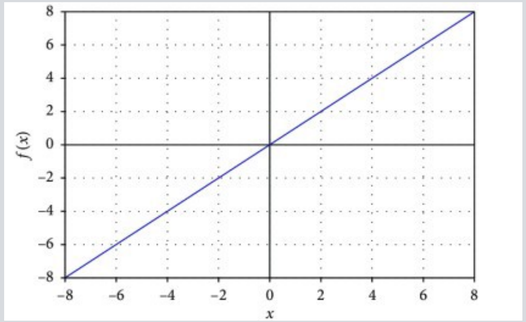
Binary activation Function

$$f(x) = \begin{cases} 0 for x < 0 \\ 1 for x \geq 0 \end{cases}$$

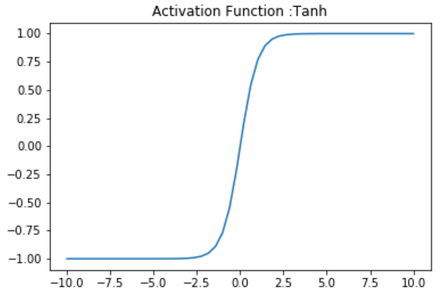


Linear Activation Function

$$f(x) = x$$



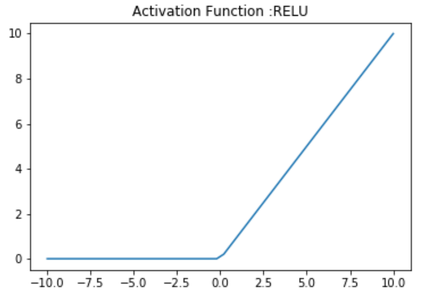
Non-linear Activation functions



tanh

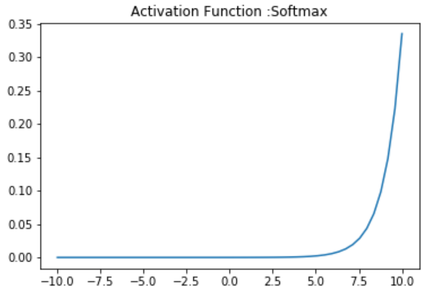
$$g(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}}$$

$$g'(z) = 1 - g(z)^2$$



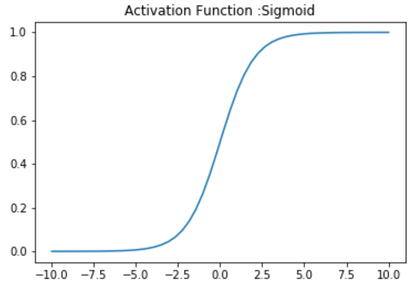
ReLU

$$g(z) = max(0,z)$$



Softmax

$$g(z) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$



Sigmoid

$$g(z) = \frac{1}{1 + e^{-z}}$$

Loss Function

- Loss is calculated for the previously used function as:

$$J(W, W_0) = \sum_{i=1}^{n} L(f((x^i); W), y^i)$$

- For any model, the main goal is to minimize the loss function.
- Also known as cost function, objective function.

$$L(f(x^i; W), y) = Loss(A^L, y)$$

$$A^L = f^L(z^L) and z^L = W^{L^T} A^{L-1}$$

Back Propogation

- To minimize the loss function, we apply chain rule w.r.t the weights in the final layer:

$$\frac{\partial \text{loss}}{\partial W^L} = \underbrace{\frac{\partial \text{loss}}{\partial A^L}}_{\text{depends on loss function}} \cdot \underbrace{\frac{\partial A^L}{\partial Z^L}}_{f^{L\prime}} \cdot \underbrace{\frac{\partial Z^L}{\partial W^L}}_{A^{L-1}}$$

- If we continue to apply chain rule we can write the loss as,

$$\frac{\partial \text{loss}}{\partial Z^l} = \frac{\partial A^l}{\partial Z^l} \cdot W^{l+1} \cdot \frac{\partial A^{l+1}}{\partial Z^{l+1}} \cdot \ldots W^{L-1} \cdot \frac{\partial A^{L-1}}{\partial Z^{L-1}} \cdot W^L \cdot \frac{\partial A^L}{\partial Z^L} \cdot \frac{\partial \text{loss}}{A^L}$$

Loss Optimization

- Optimization through gradient descent

$$W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$$

- η is the learning rate, so our target to setting this learning rate,

There are various methods
1. Use of Gradient Descent Algorithm(SGD, ADAM, Adadelta, Adagrad, RMSProp)
2. Divide the dataset into mini-batches (which leads to faster training)

Motivation

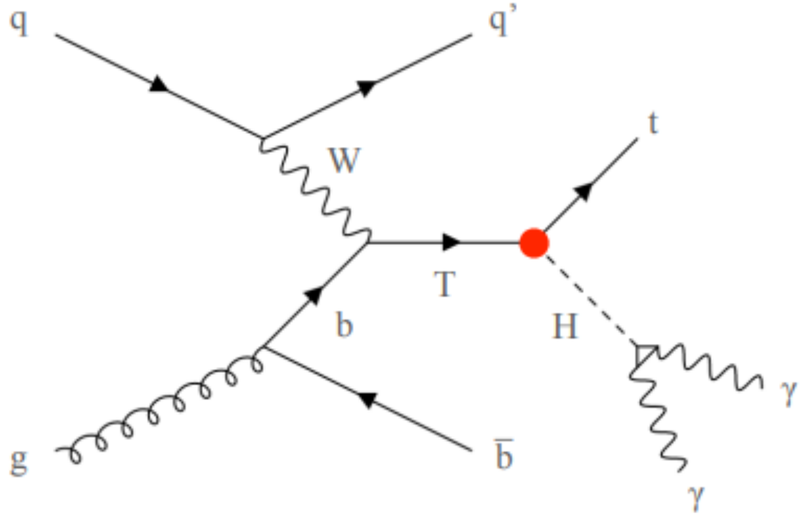
To Classify the signal which has small cross section compared to the background processes

Signal(Resonance Particle)

Resonance Particle

- Particles with very short life time in the order of 10^{-23} s.

- Signal corresponds to the features, processes, or events, in which we are interested.
- The resonance particles T' is used as Signal.
- T' is hypothetical spin-1/2 coloured particles with electric charge of +2e/3.
- T' is a vector like quark(VLQs) with two production modes.
    1. Pair production through strong interaction
    2. Single production through electroweak interaction



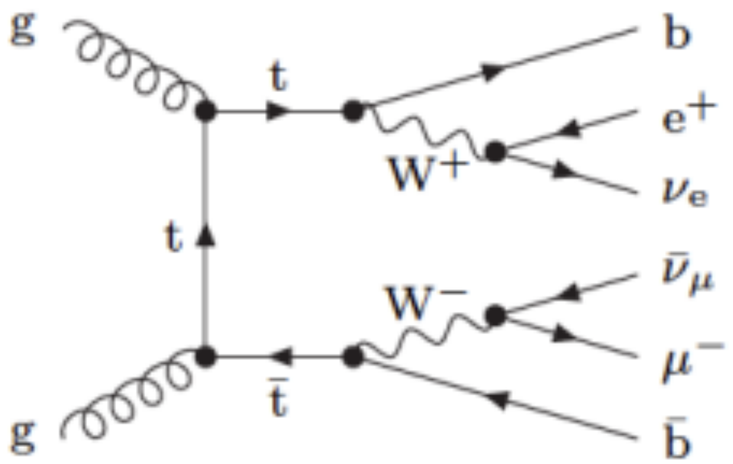
Tprime

Backgrounds

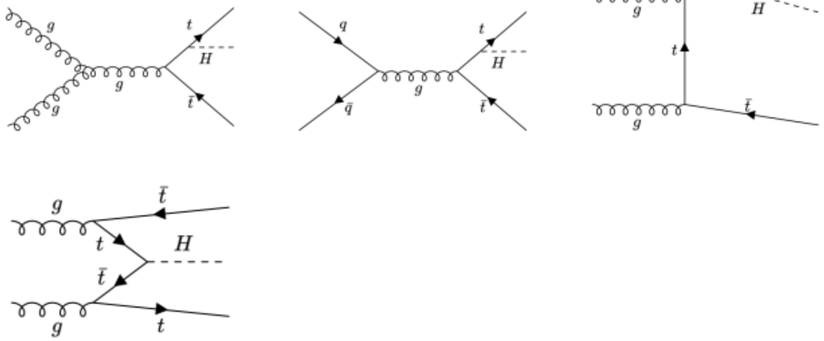
- These backgrounds are the events that may seem like signal but we are not interested in.

- $tt\gamma\gamma$, ttH, and tHq have been used as the backgrounds.

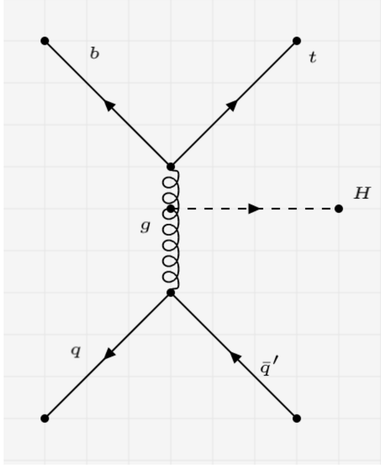
$tt\gamma\gamma$



$\mathrm{ttH}(\rightarrow \gamma\gamma)$



$\mathrm{tH}(\rightarrow \gamma\gamma)\mathrm{q}$

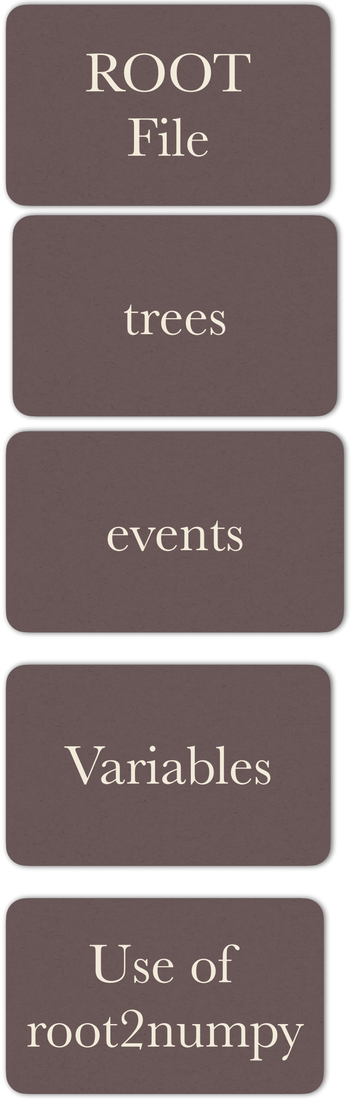


Input Data Variables used for model training

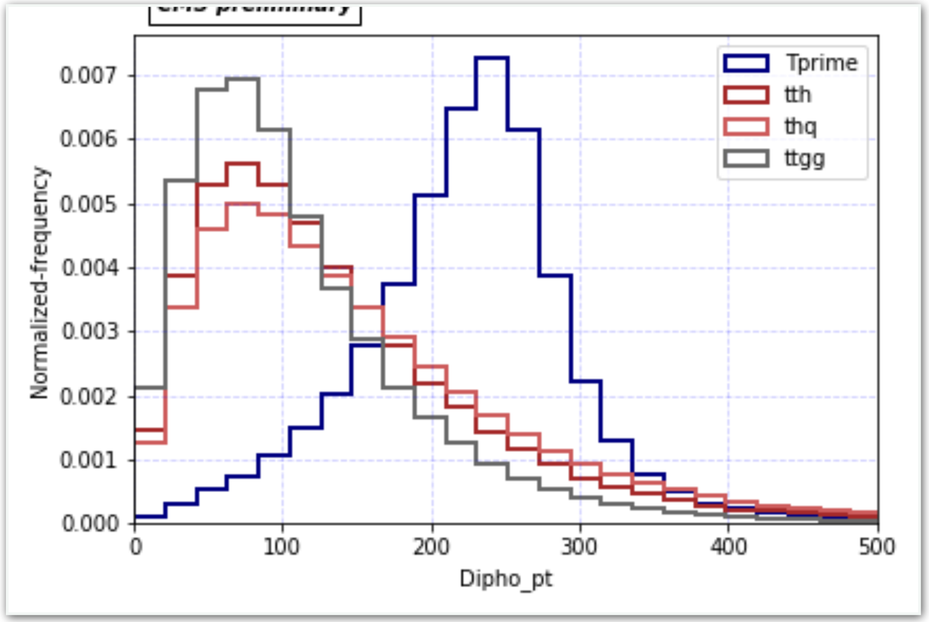
There were total 29 input variables were used from data files:

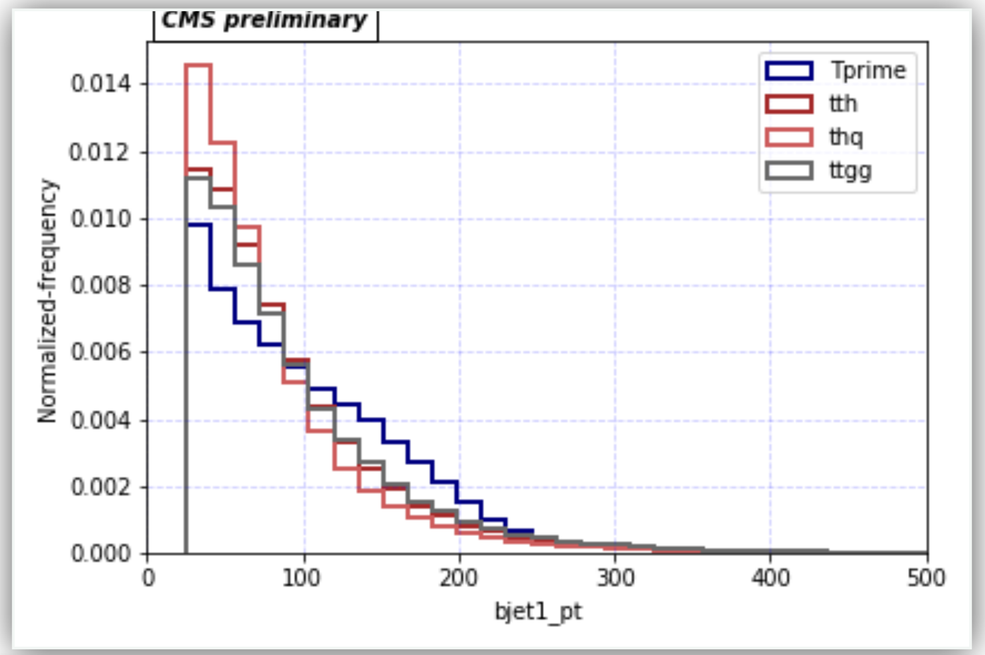
- $dipho_{P_T}$
- $dipho_{\phi}$
- $dipho_{\eta}$
- $dipho_{lead\eta}$
- $dipho_{lead\phi}$
- $dipho_{sublead\eta}$
- $bjet1_{\eta}$
- $bjet2_{\eta}$
- $bjet2_{\phi}$
- $bjet3_{\phi}$
- $dipho_{e}$
- $dipho_{mass}$
- $bjet1_{\eta}$
- $jet1_{P_T}$
- $Jet2_{P_T}$
- n_{jets}
- $jet1_{e}$
- $jet2_{e}$
- $jet3_{e}$

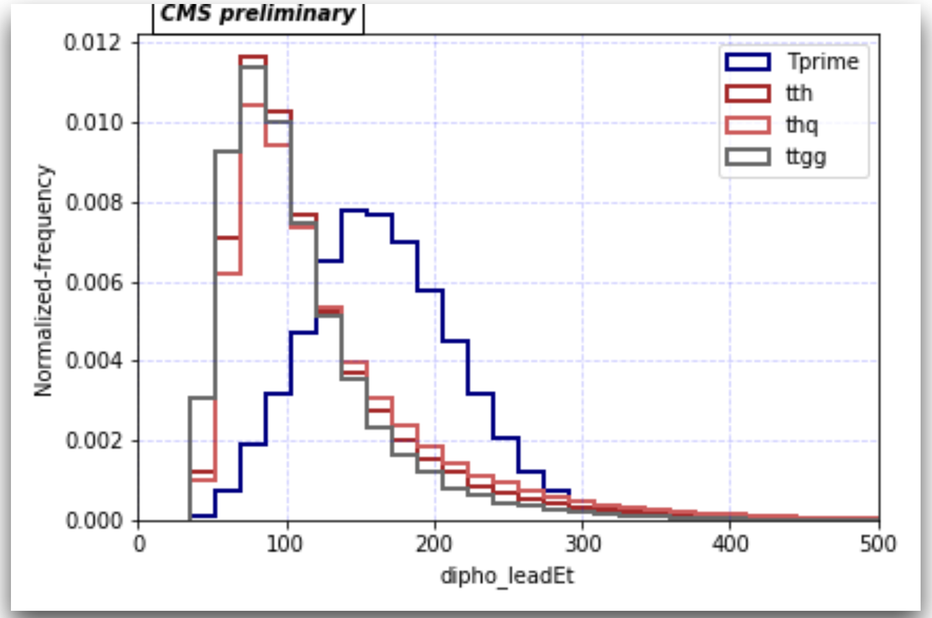
How to Read Data from Root File in python?

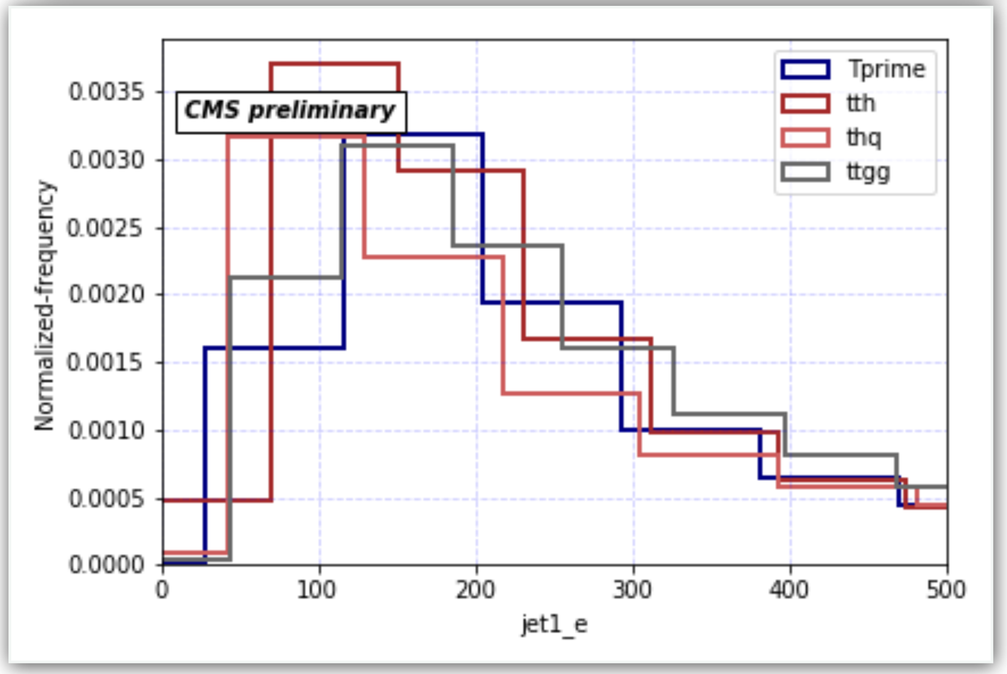


plot of input variables

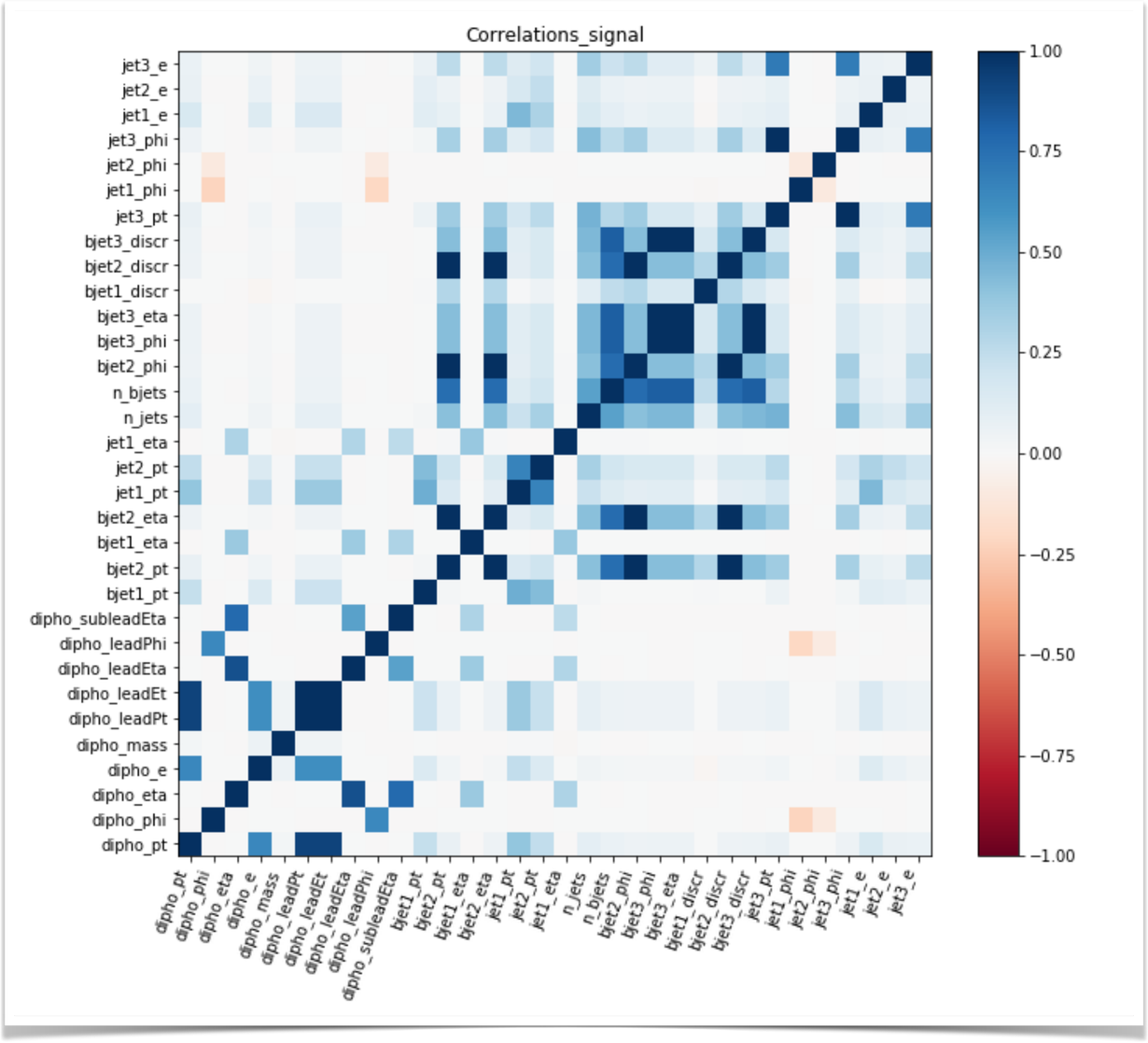








Correlation Plot



correlation coefficient formula

$$r = \frac{n(\sum xy) - \sum x \sum y}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

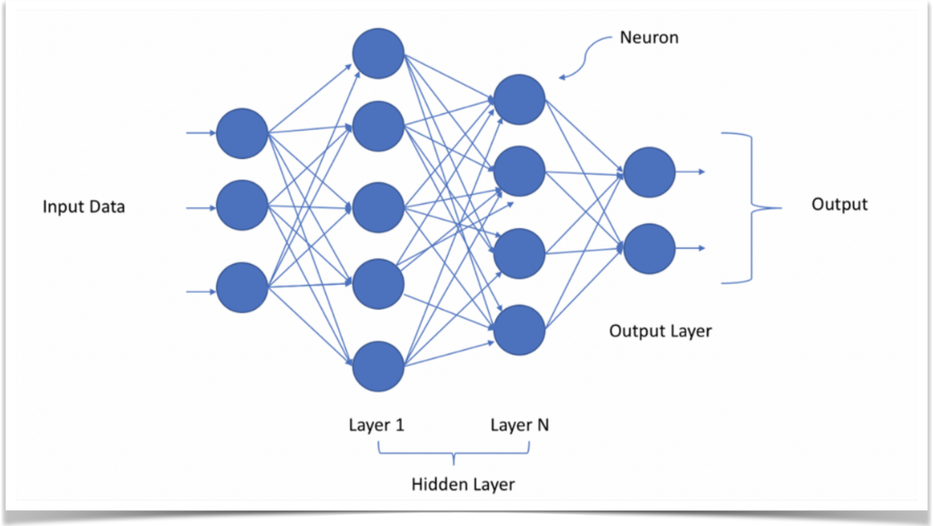
- Represents strength of linear dependence between variables
- Two types of correlation
  - Positive correlation: Both variable moving in same direction
  - Negative correlation: Both variable moves in opposite direction
- +1 is perfect positive correlation -1 Shows perfect negative correlation.

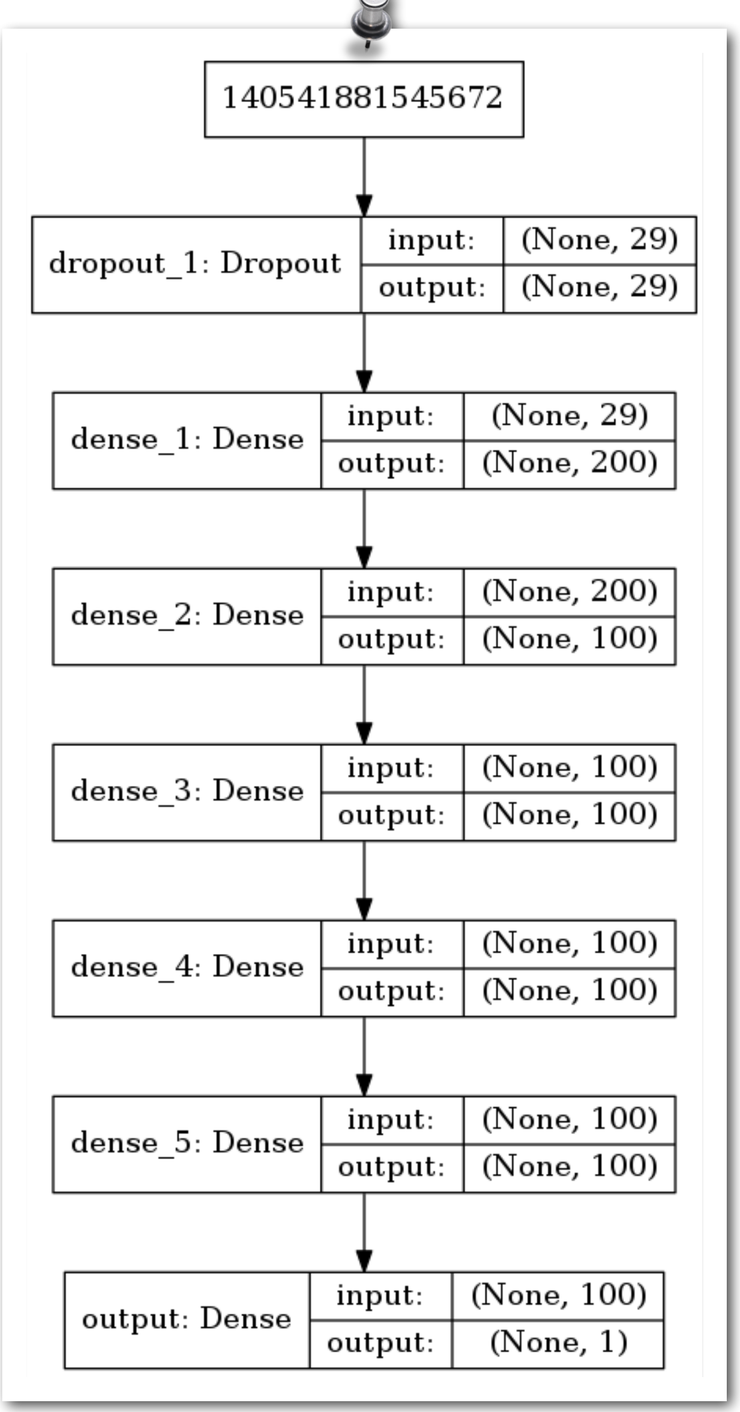
Deep Neural Network(DNN)

Model Summary

Total number of input variable :- 29
Initial weight =5
Total number of hidden layer :- 5
Activation Function(Hidden):- "ReLU"
Activation function(output)="sigmoid"
Loss function:- "binary_crossentropy"
Optimizer:- ADAM
Batch size:- 900
No. of epochs:-100

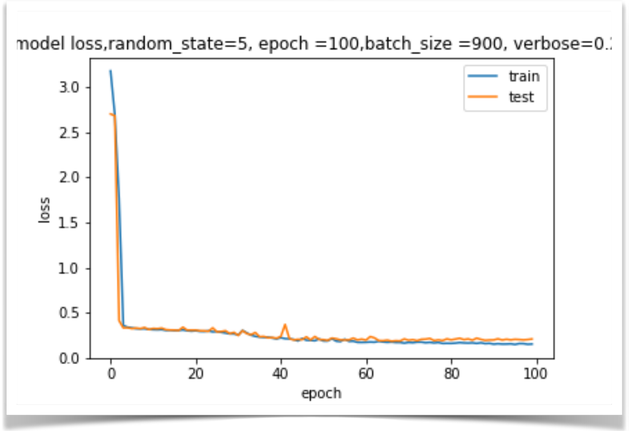
- whole dataset divided into the ratio of 80:20.

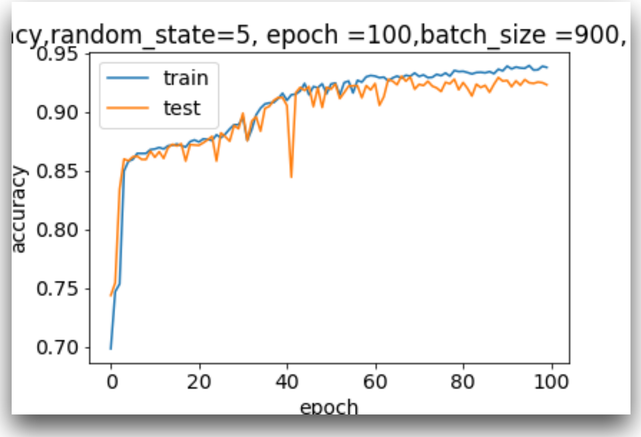




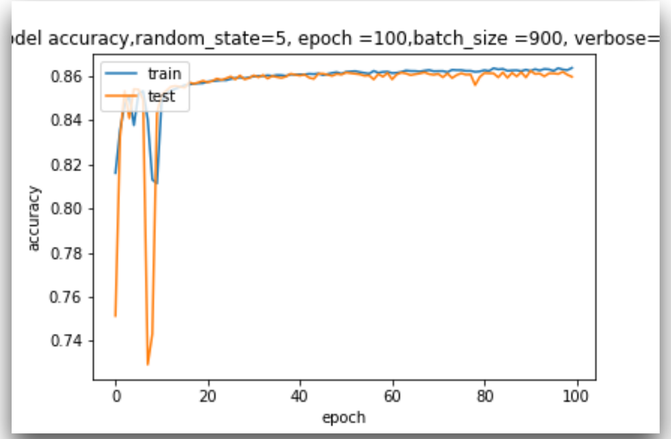
Output (DNN)
Binary Classification

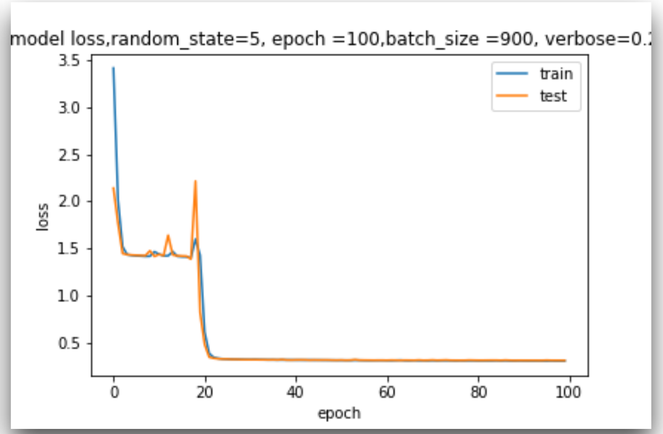
Signal	Background	Training Accuracy	Testing Accuracy
T'	$tt\gamma\gamma$	93.30%	92.06%
T'	$tt\gamma\gamma$ & tth	89.84%	89.07%
T'	$tt\gamma\gamma$ & tth &thq	86.36%	86.10%





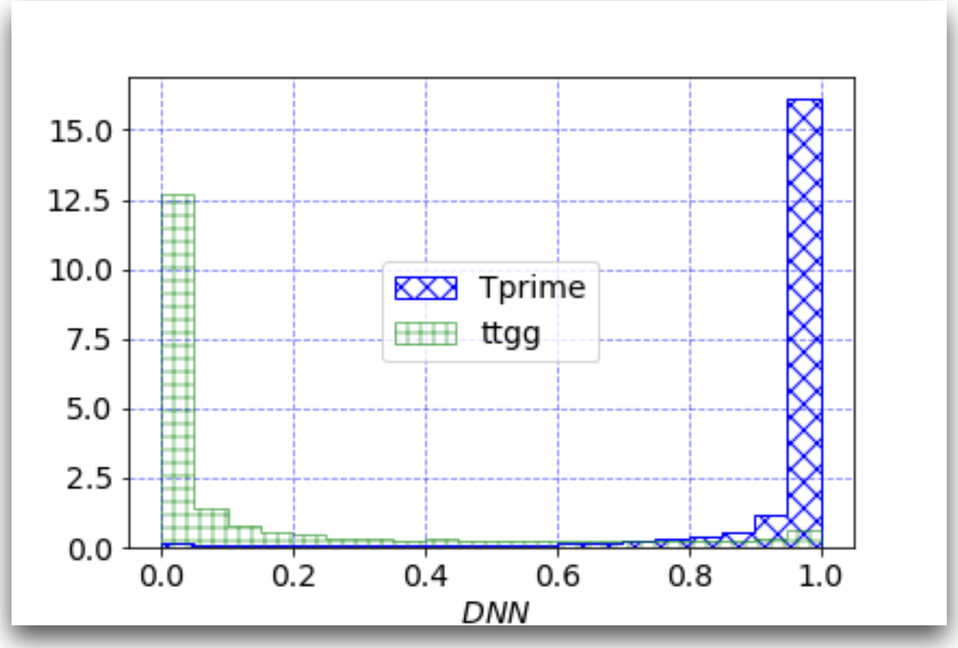
Model output when Tprime used as Signal and $tt\gamma\gamma$ used as the background





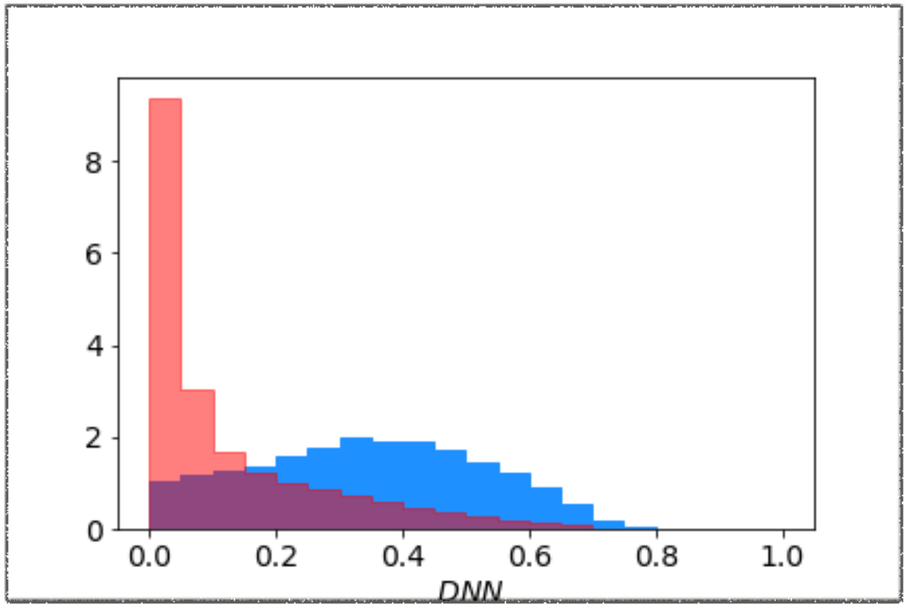
T' as signal & $tt\gamma\gamma$, ttH, and tHq as background

Output(DNN)

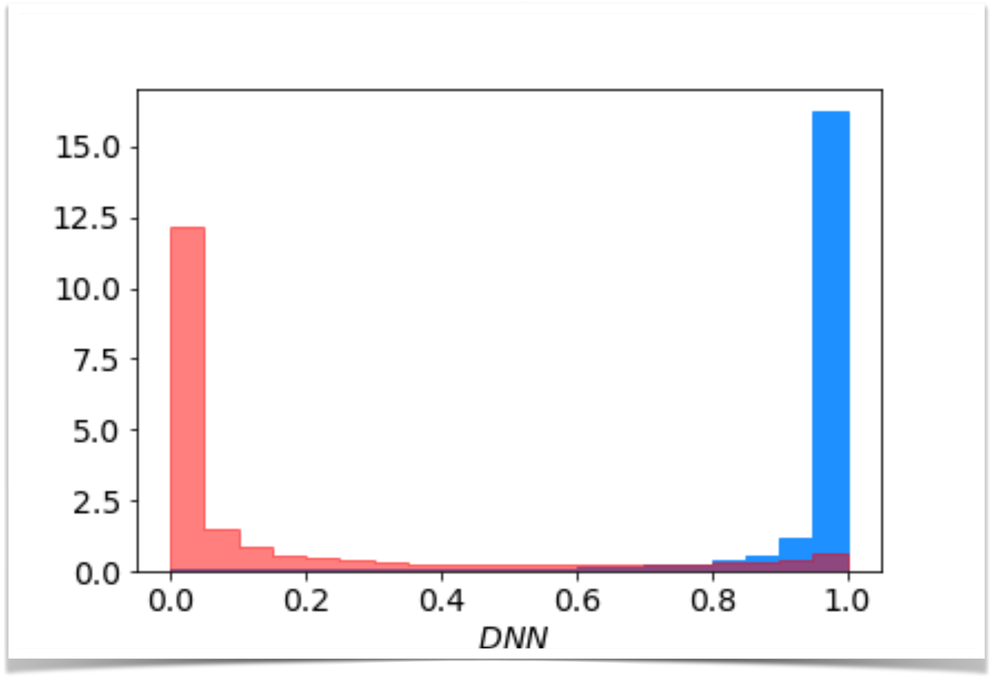


T' as signal and $tt\gamma\gamma$ as the background

T' as signal and,$tt\gamma\gamma$,ttH, and thq as background



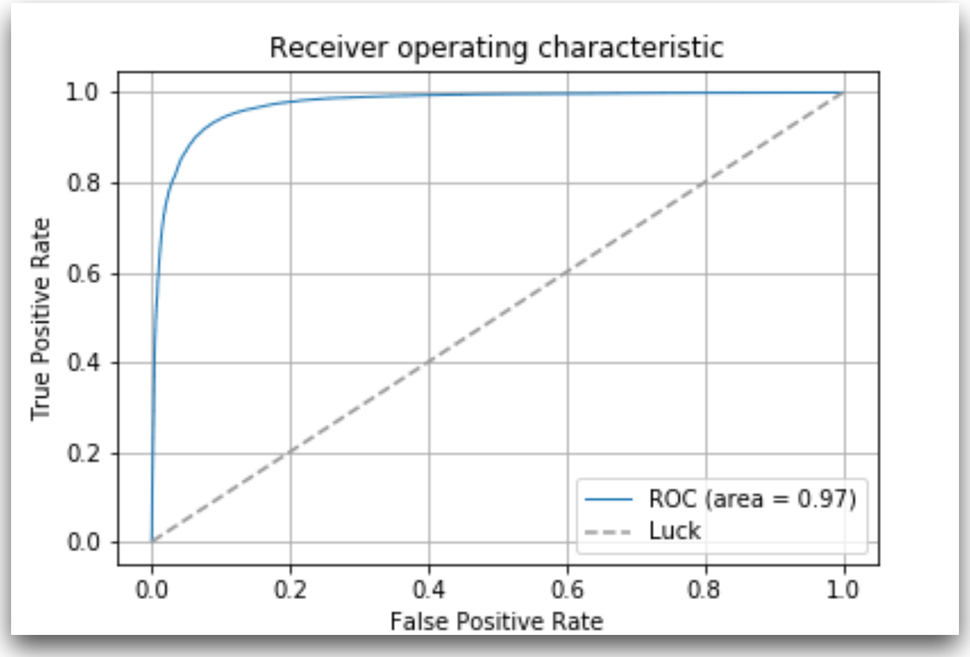
$tt\gamma\gamma$ and ttH as the background



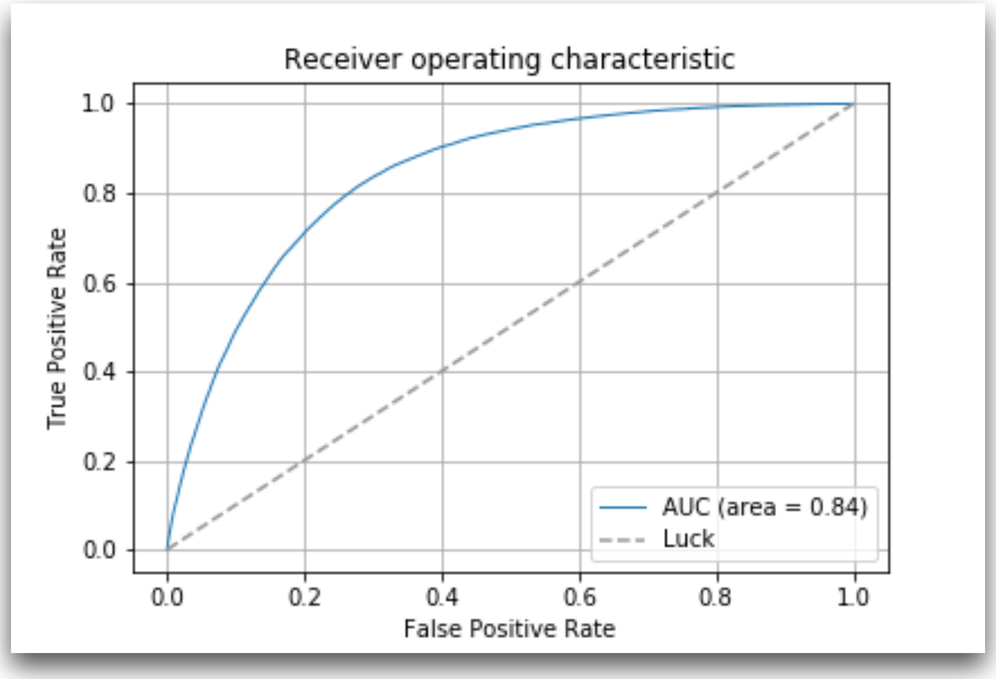
Output(DNN)

- Receiver Operating Characteristic Curve(ROC Curve)
- True positive rate is signal efficiency and False positive rate is background efficiency.
- A graph showing the performance of a classification model at all threshold

ROC Curve for Tprime and $tt\gamma\gamma$



ROC for Tprime as signal and ttH, tHq, and $tt\gamma\gamma$ as background



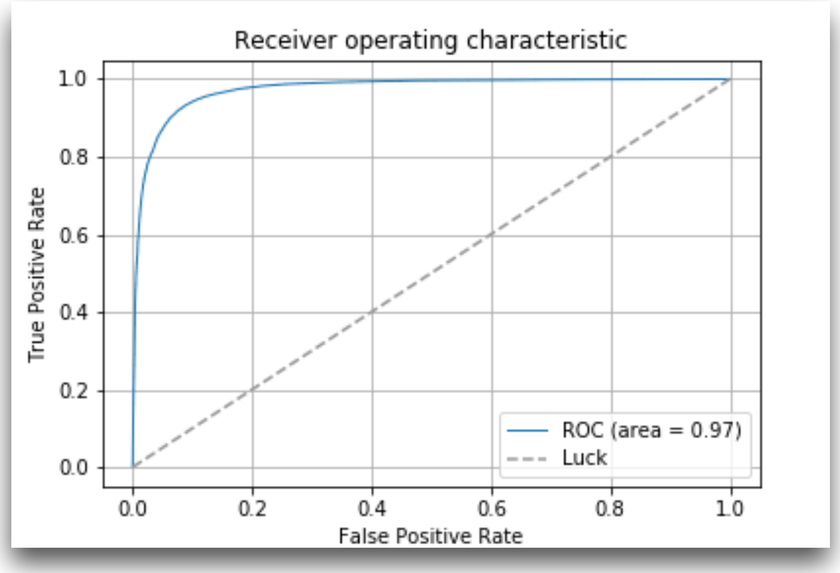
Comparison of Output from DNN and BDT

- ROC for DNN are better compared to BDT
- This data comparison done over same sample.
- BDT are simple to implement.

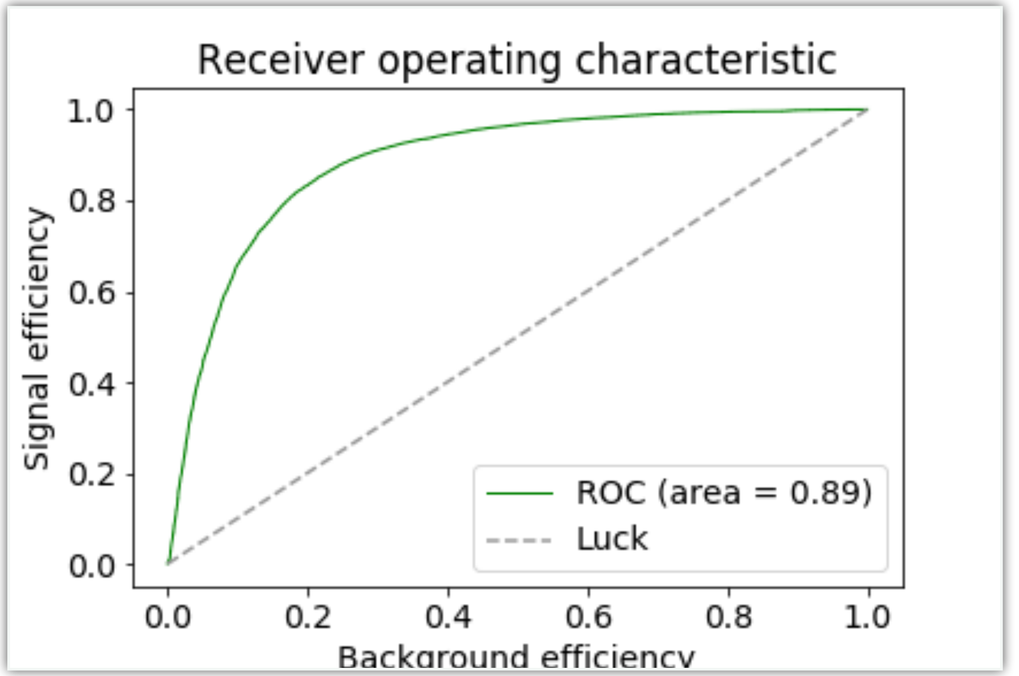
Boosted Decision Tree

- Consists of leaves
- Classify the data into classes and then predict the output after training.
- It train on sequence of data.

ROC for DNN



ROC for BDT



Summary

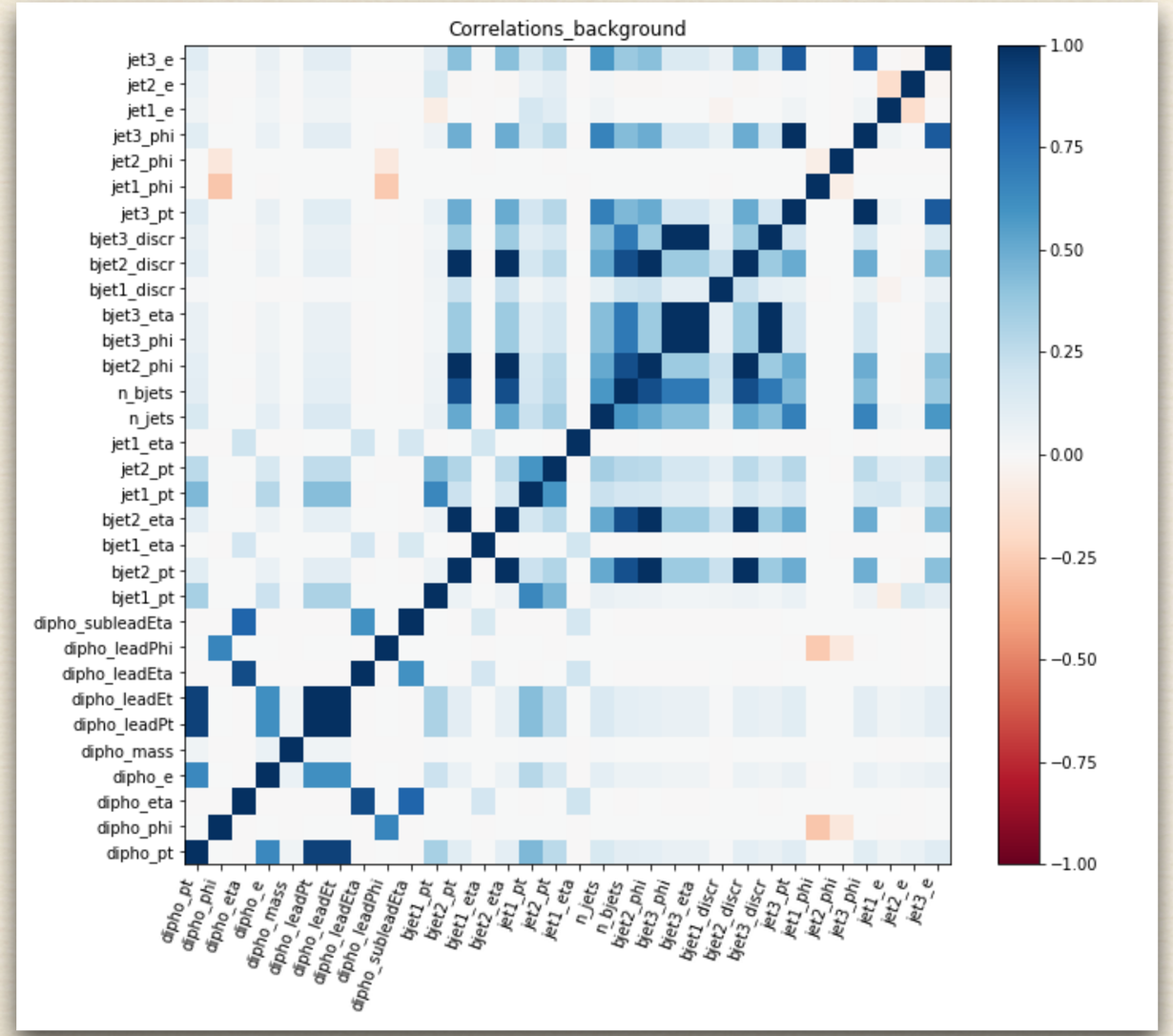
- Different machine learning techniques have been explored in this project.
- Extensively studied multivariate techniques to separate arbitrary signal (T') processes from background ($t\bar{t}\gamma\gamma$, ttH, thq) processes using CMS simulations.

- DNN seems to have better sensitivity in terms of signal-background separation.

- In future, DNN performance needed to be checked on pseudo-data with quantitive statistical interpretation.

Thank You

Correlation plot for Background(ttH)



Loss Functions

Binary classification

Multi classification

Cross entropy

$$CE = -\sum_{i}^{C} t_i log(s_i)$$

Binary cross entropy

$$CE = -\sum_{i} t_i log(s_i) = t_1 log(s_1) - (1 - t_1) log(1 - s_1)$$

Categorical Cross entropy

Why ReLU in Hidden Layers?

- Since ReLU activates only a certain number of neurons, so help in computation efficiently.
- "ReLu" gives out a real no. output 0

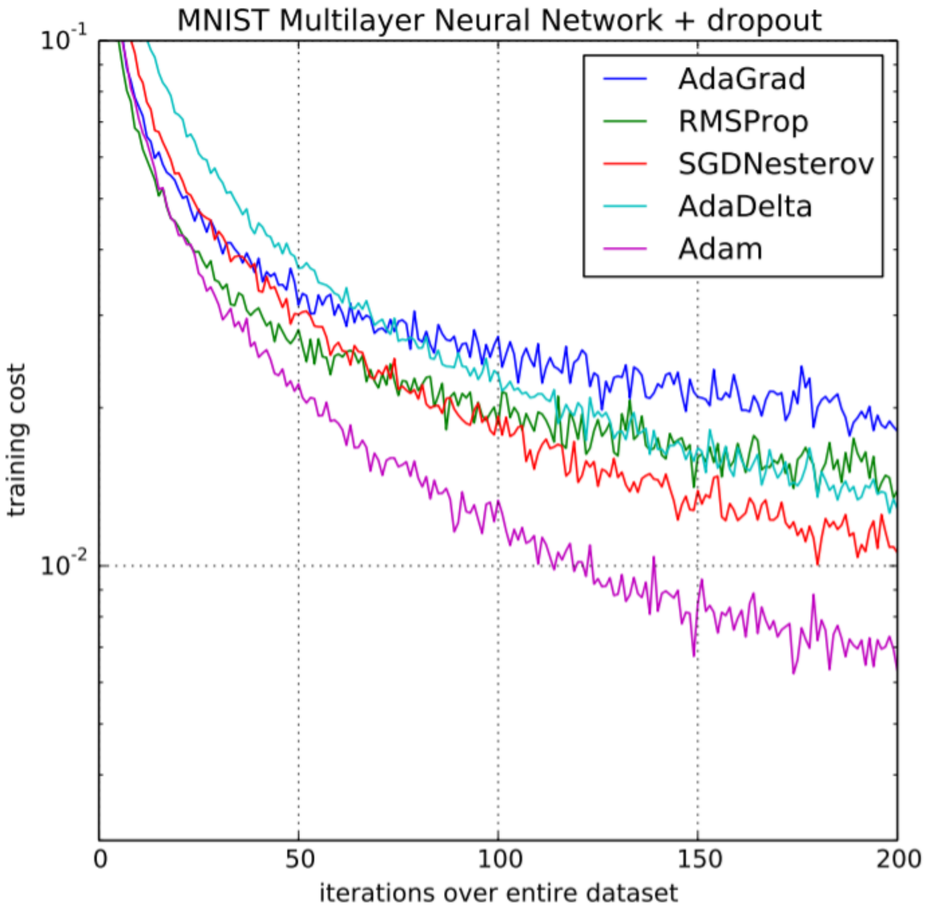
Why sigmoid in last layer?

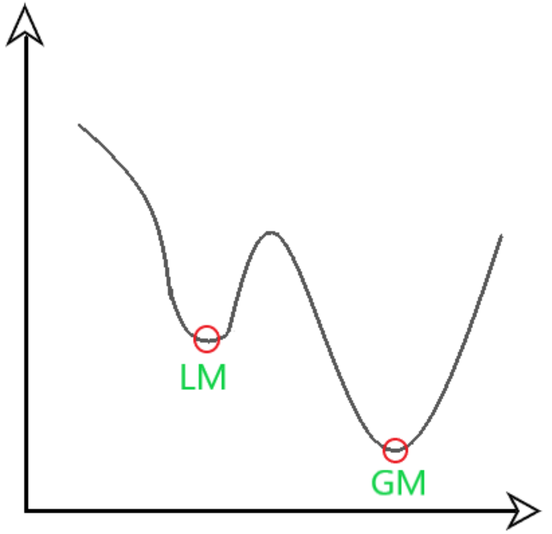
- For "Sigmoid" function output is [0,1]
- for binary classification we check if output >0.5 then class 1, else 0.
- outputs are NOT mutually exclusive

Why not Softmax insted of Sigmoid?

- Softmax also have output in [0,1]
- sigmoid function is used for the two-class
- Whereas, softmax function is used for the multiclass logistic regression
- If the outputs are mutually exclusive, then use a softmax function

ADAM





GM - Global Minimum
LM - Local Minimum