Applying Artificial Neural Networks in Data Analysis

Jacek Biernat

UPPSALA UNIVERSITET

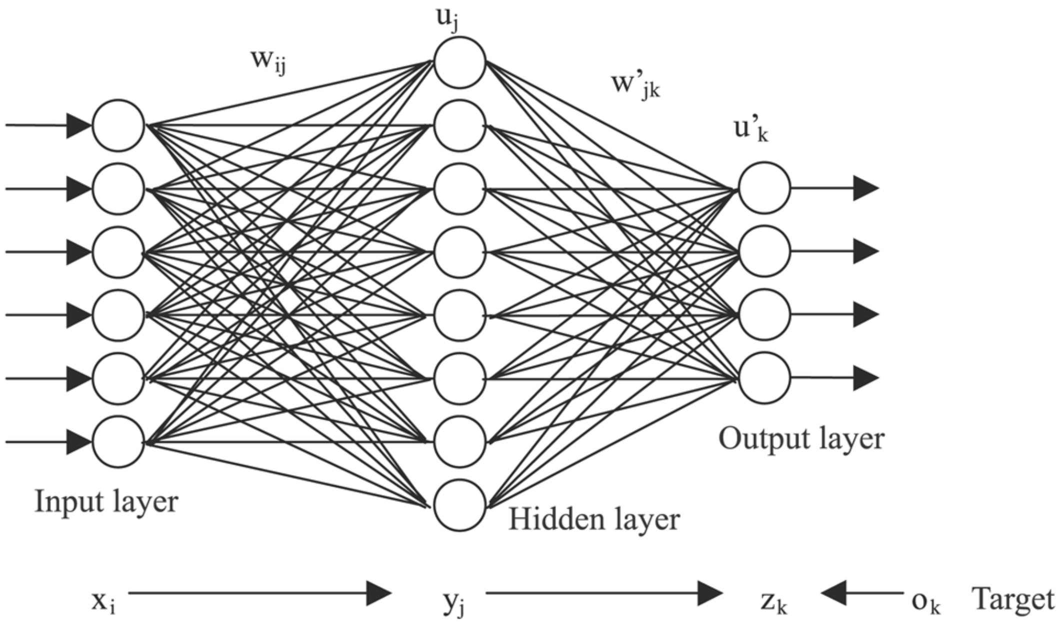
BESIII

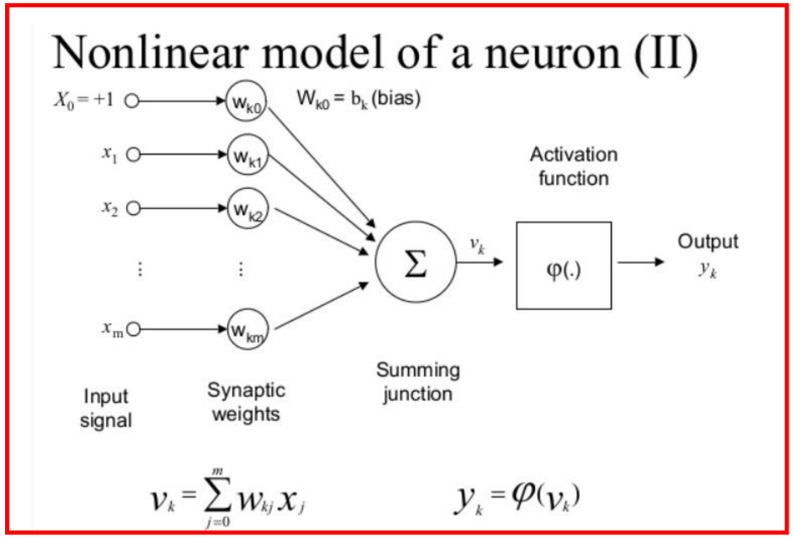
The Menu:

- What is a neural network ?
- Types of neural networks
- Available interfaces
- Few words about BESIII
- BNN for background suppression:
    - Analysis Steps
    - Results
    - Conclusions

Neural Network

- Mathematical tool
- Suitable for finding patterns in big data sets
- Similar to a “human brain” ☺





Neural Networks

$$a_{t+1,j}(\mathbf{x}) = \sum_{r:\,(v_{t,r},v_{t+1,j})\in E} w((v_{t,r}, v_{t+1,j}))\, o_{t,r}(\mathbf{x}),$$

Output of a ANN where:
V_{ti} – is the **i`th** neuron of the **t`th** layer
O_{ti} – neuron output
X – input vector

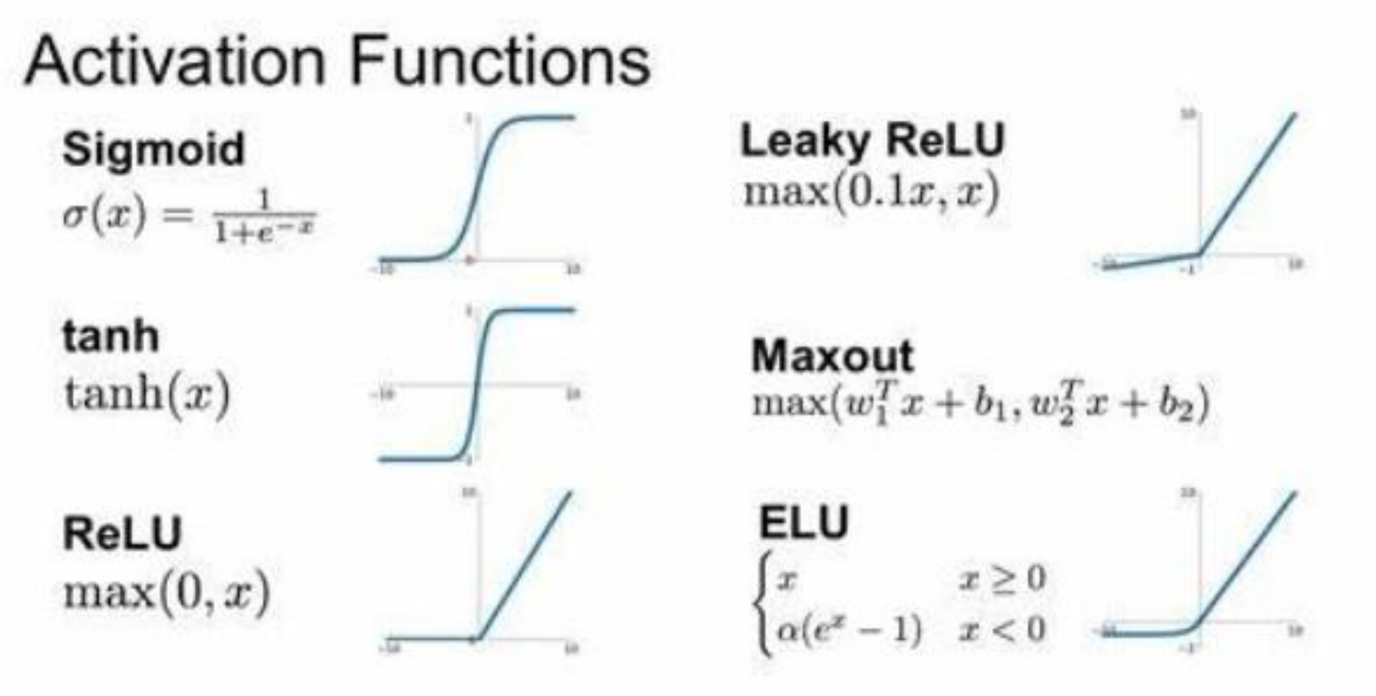
One can define the neuron output as:

$$o_{t+1,j}(\mathbf{x}) = \sigma\left(a_{t+1,j}(\mathbf{x})\right).$$

σ – neuron activation function (ϕ – in previous slide)

http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf

Neural Networks – activation functions



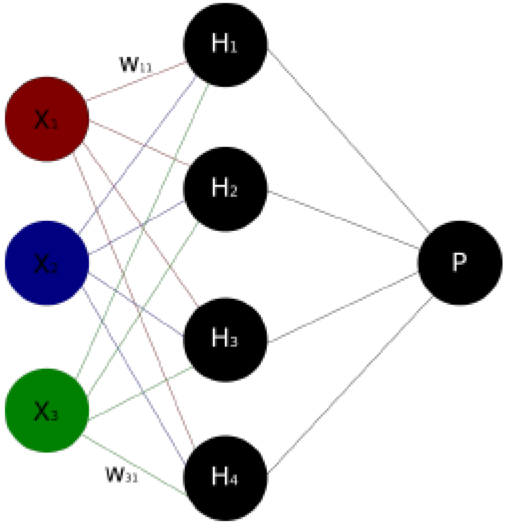
Types of Neural Networks

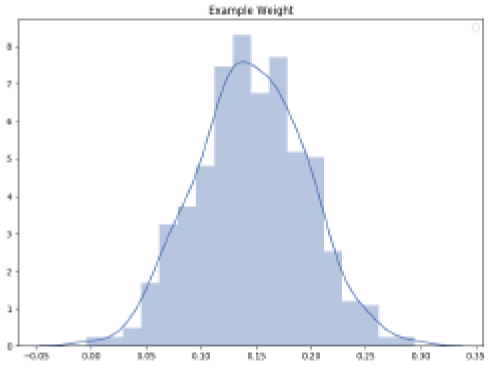
Conventional NNs

- Convolutional NN (require Image-Data)
- Multi-Layer Perceptron (MLP)

Bayesian NN[1]

- weights are distributions
- gives a set of NNs in one Model





Interfaces

ROOT and TMVA

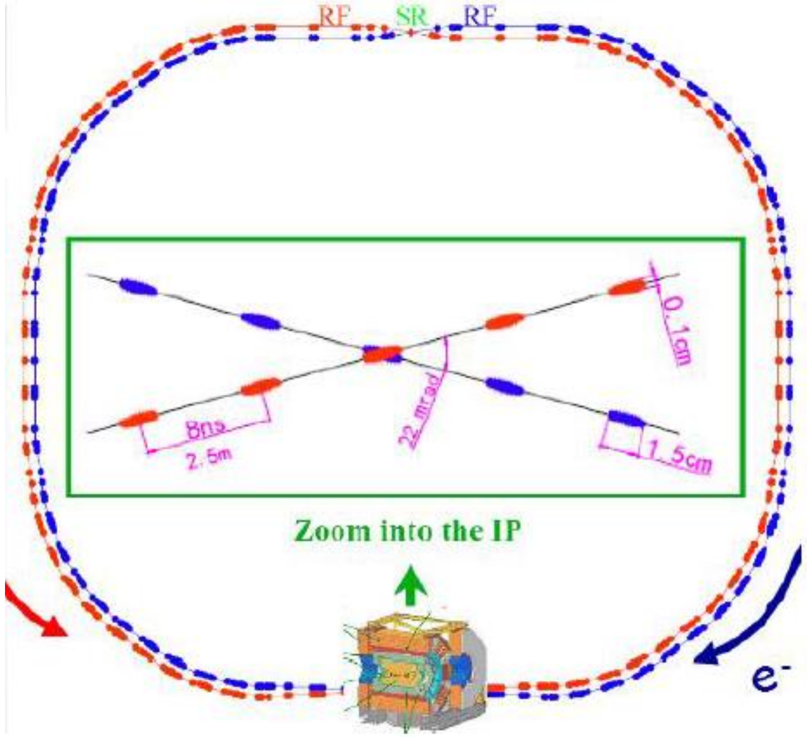
- ROOT: tool for Data Analysis
- widely used in physics
- TMVA: Toolkit for Multivariate Analysis

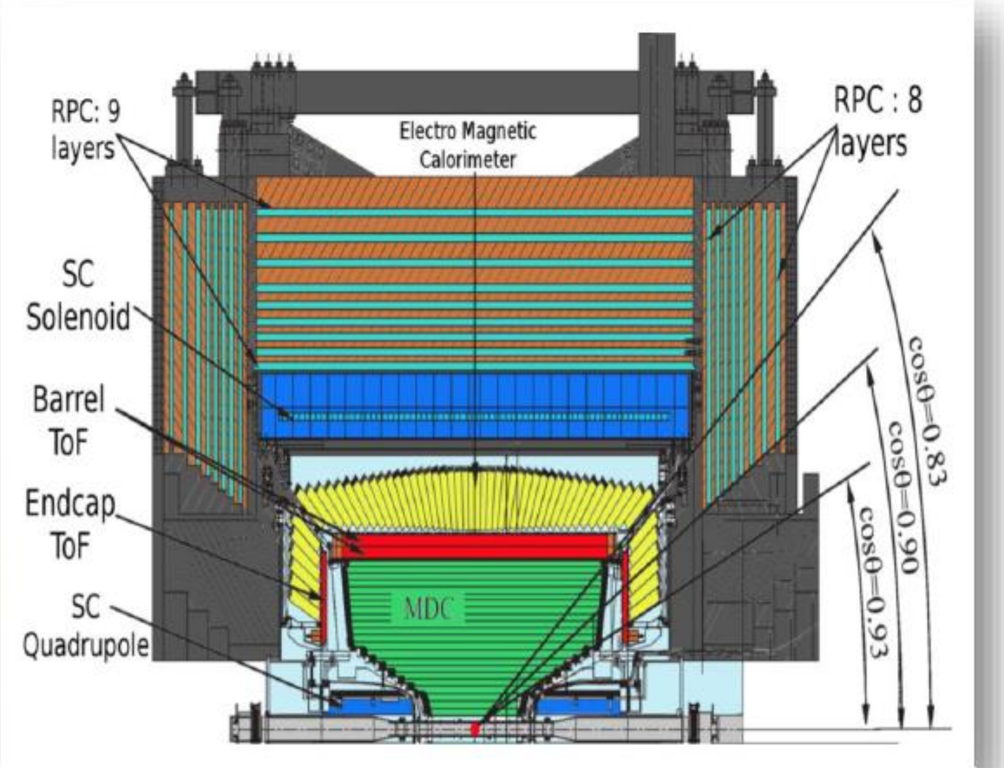
Tensorflow and Keras

- Keras simplifies construction of NN-models
- with Python: Quick way to test your model
- Additional Tensorflow Libraries for Bayesian Approach
- NVIDIA GPU Cards with CUDA-capability are supported
- GPU computations improve training and inference process

BES III- Introduction

- BEPC = Beijing Electron Positron Collider.
- Operates in the τ-charm mass region

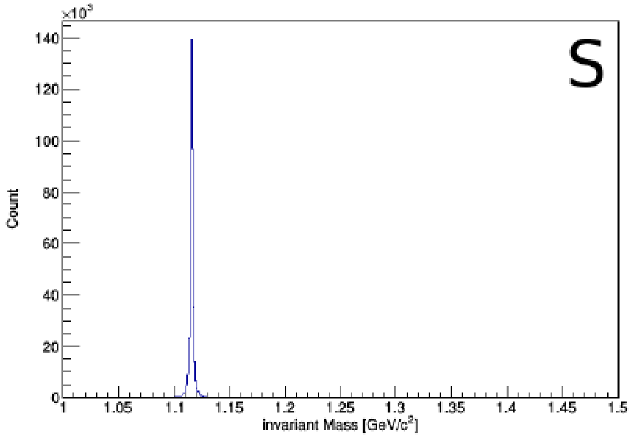




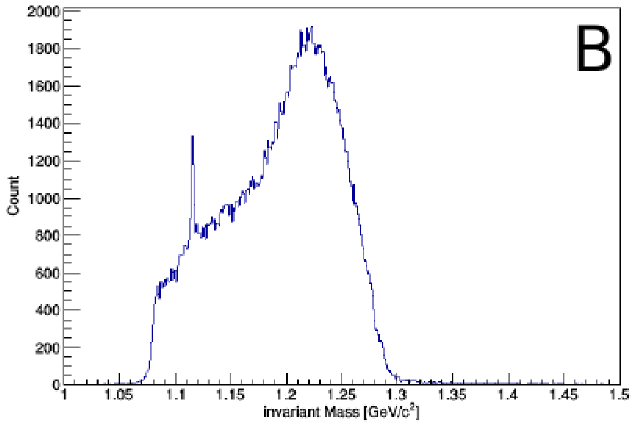
- Charmonium physics
- Light hadron
- Hyperon physics:
- Hyperon form factors
- Decay asymmetry parameters of hyperons

The Analysis

- Training Model:
  - Input vector ?
  - Signal Sample
  - Background Sample
- Principal Component Analysis
- Training the NN
- Application on unlabelled data



S : Signal labelled



B : Background labelled

S + B = Training Set

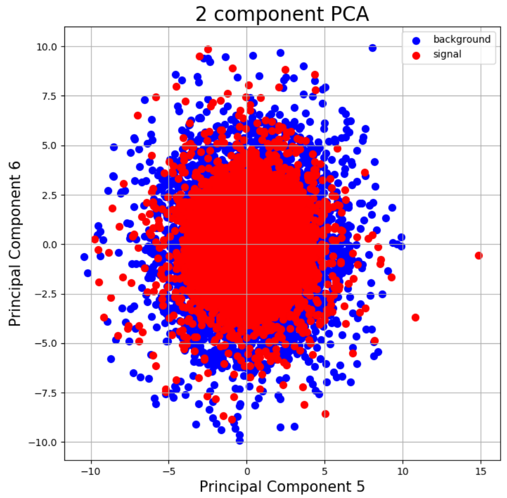
$$y^{`}[i] = vertex_{x,y,z}[i] + DecayLength[i] + MDC_{x,y,z}[i](\pi; p)$$

Degree project by my two students: Tim Littau and Tobias Nordahl

Principle Component Analysis

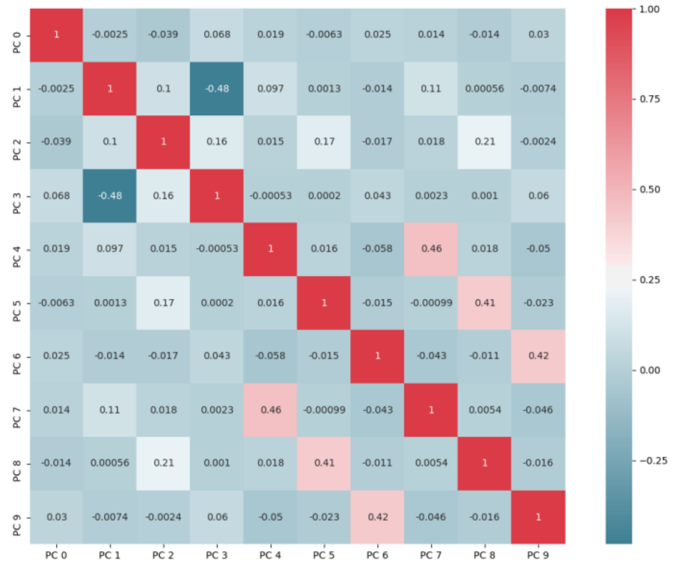
- Decorrelation
- Dimensionality reduction
- Separation (Unsupervised Learning)

Original



Full overlap – no separation, a more advanced method required <u>BNN</u>

After PCA



For example:
Principal component analysis

Jake Lever, Martin Krzywinski & Naomi Altman *Nature Methods* volume14, pages641–642 (2017)

The Analysis – Bayesian Neural Network

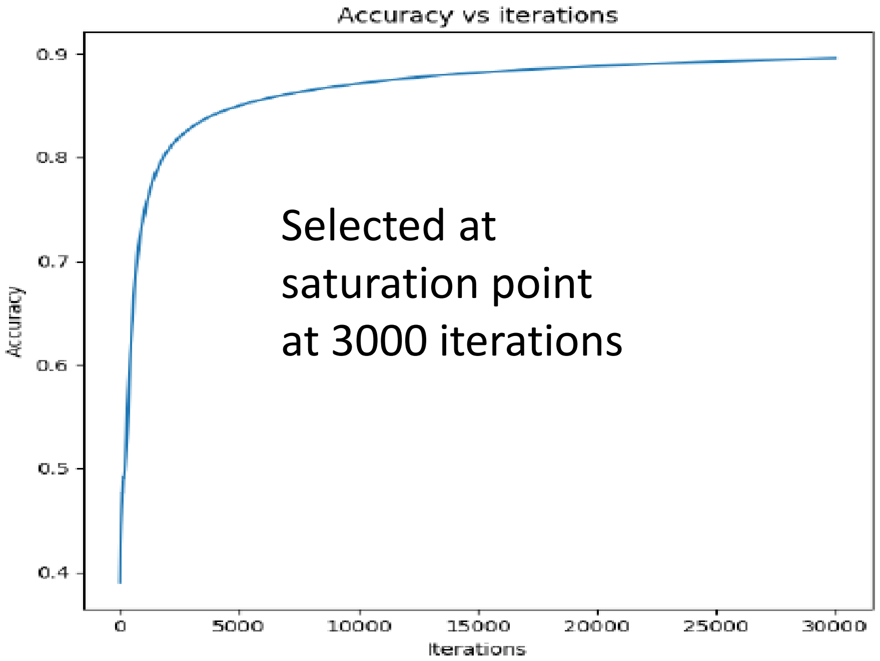
- BNN for S/B separation (background suppression):

✓10 element input vector

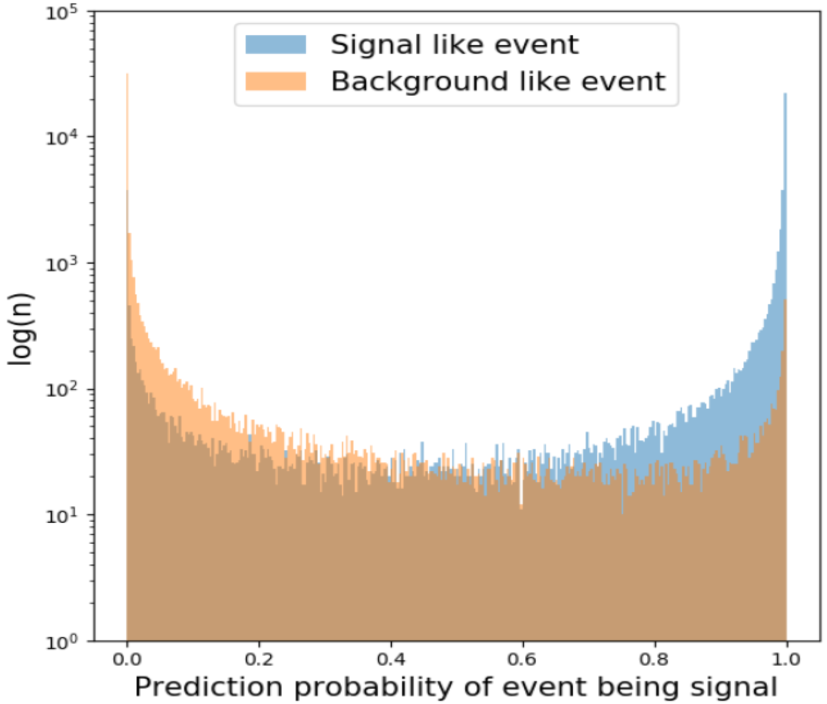
✓22 nodes (neurons) in hidden layer (simple architecture = stability)

✓2 output nodes

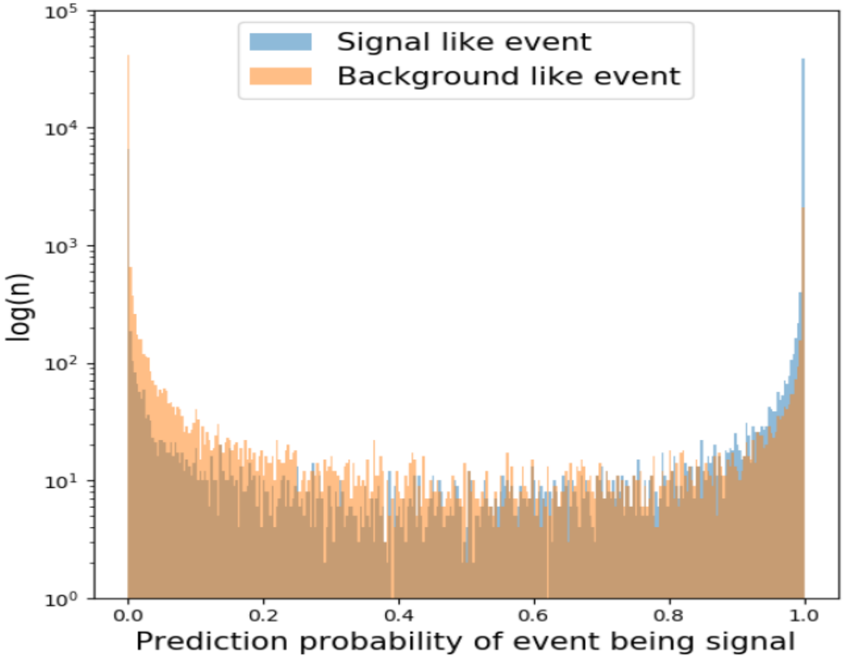
The analysis – training the NN



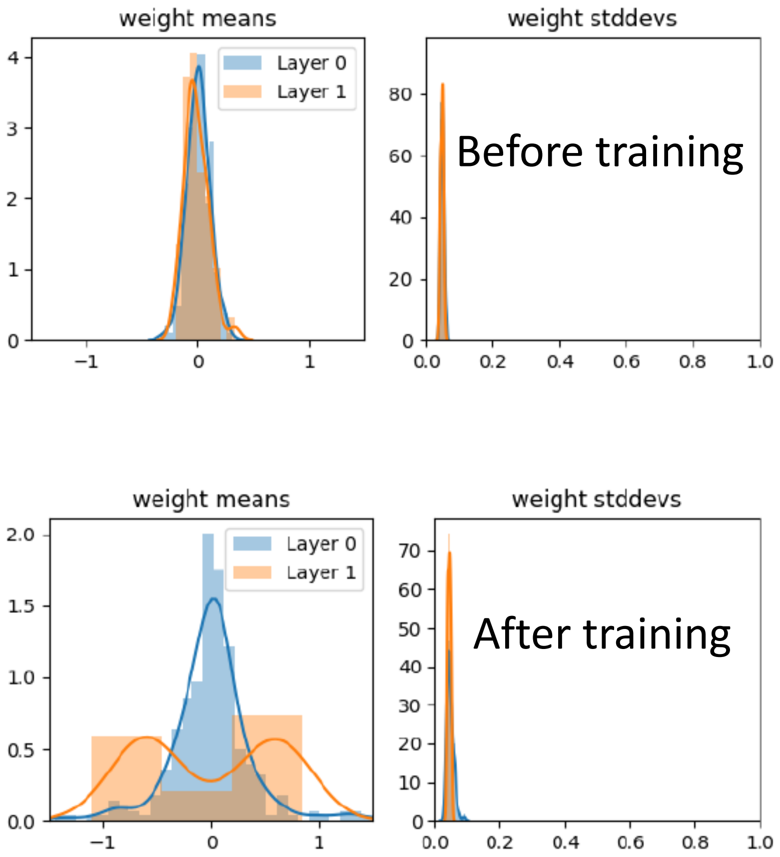
Network trained at 3000 iterations



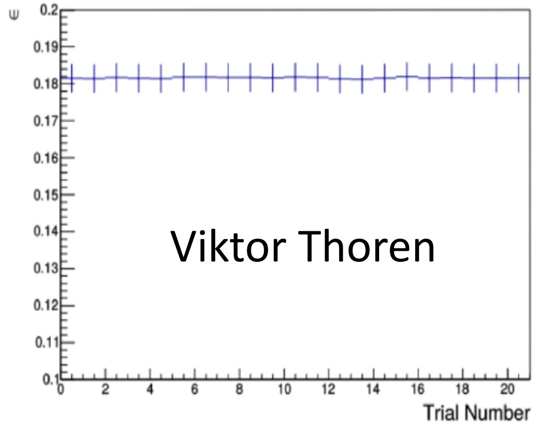
Network trained at 30000 iterations. **<u>Over trained</u>**



The Analysis- Sampling Weight distributions



- ➢ Instability coming form sampling weight distributions?
- ✓ Evaluate this effect with MC sample for 20 applications!

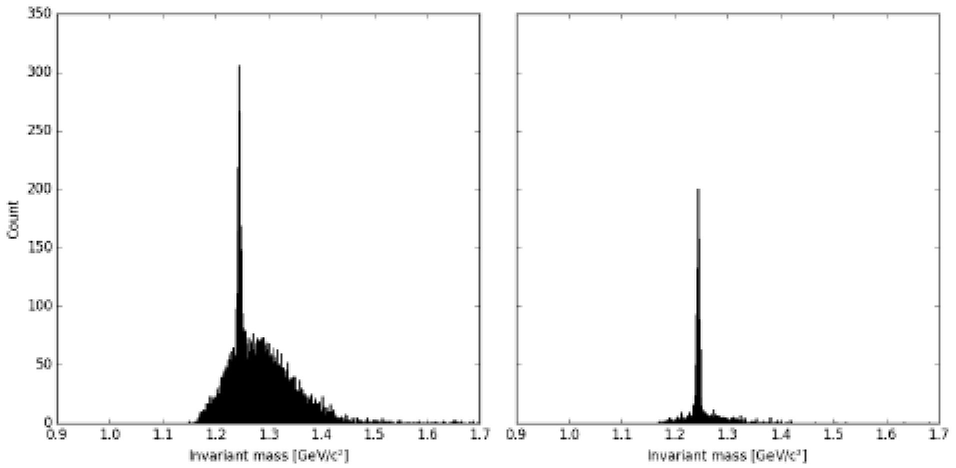


$$\sqrt{\frac{\varepsilon_i(1-\varepsilon_i)}{N_i(MCtruth)}}.$$

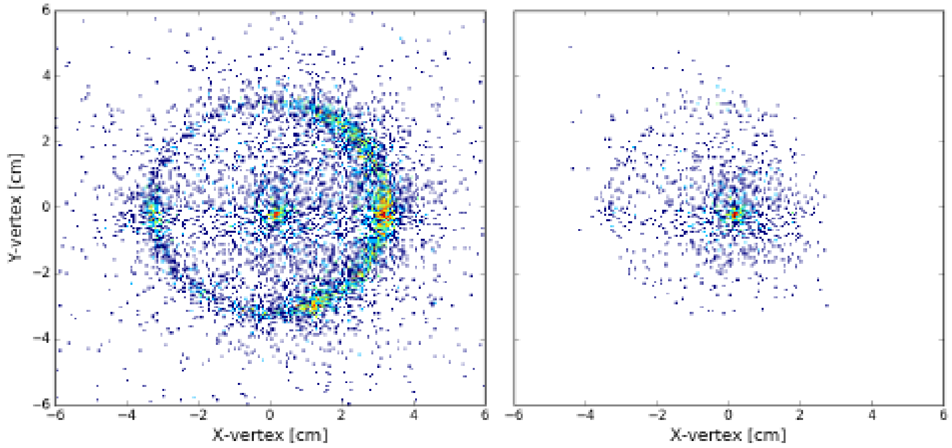
The Results

Before

After



Decay points of a given object projected on a X vs Y plane



Decay points of a given object projected on a X vs Y plane

Conclusions

- The method provides background reduction in the signal region
- The applied neural network infrastructure is simple therefore the training and application does not require a lot of computing power or time
- This method has been used and verified by Viktor Thoren (For BESIII members -> see plenary talk at Shanghai CM meeting)