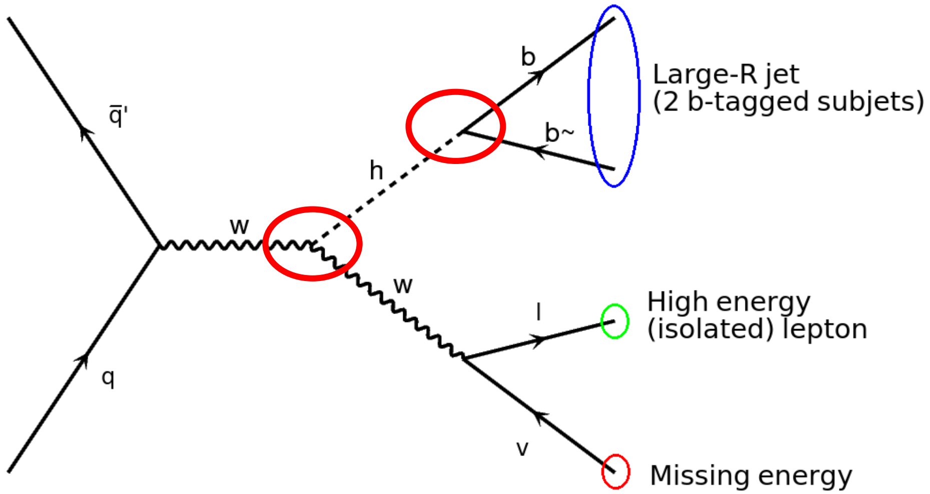
Machine learning to improve the Higgs to b-quark analysis in ATLAS

Dmytro Ostapchuk

Importance of the problem for particle physics



- The Higgs decaying to b-quark pairs finally observed last year
  - Main decay mode (BR = 58%)

- Observation of the high momentum decays important
  - To probe the SM predictions
  - Search for new physics
    - Most sensitive region!!

The main problem: backgrounds!!

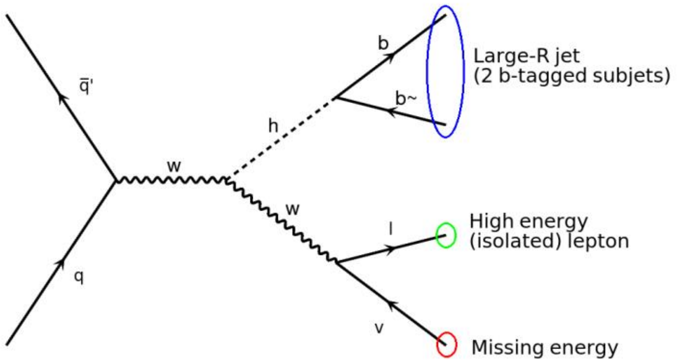


Fig.1 Signal, σ = 1.37 pb

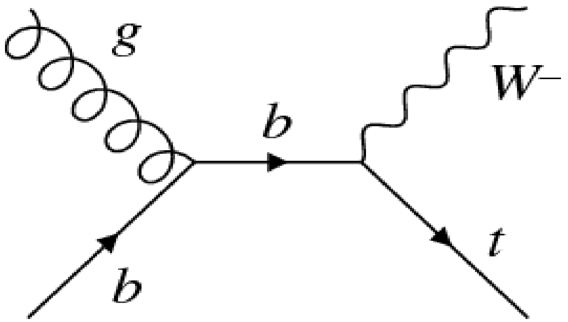


Fig.2 top, σ = 71.7 pb

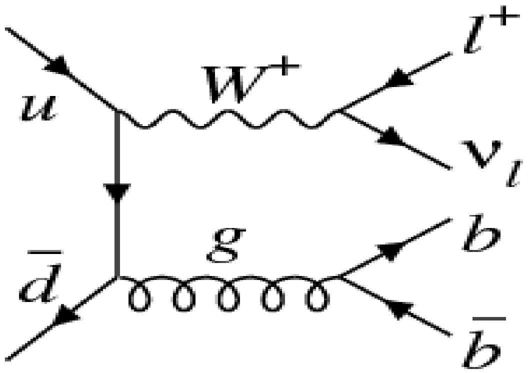


Fig.3 W+Jets, σ = 1976.5 pb

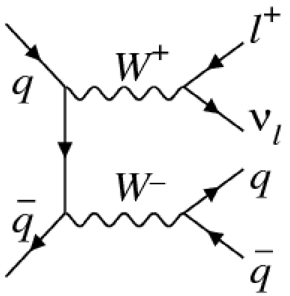


Fig.4 WW, σ = 50.64 pb

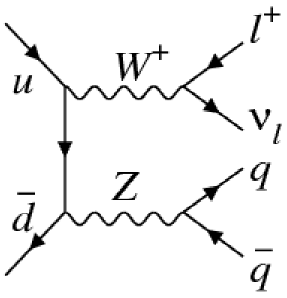


Fig.5 WZ, σ = 11.413 pb

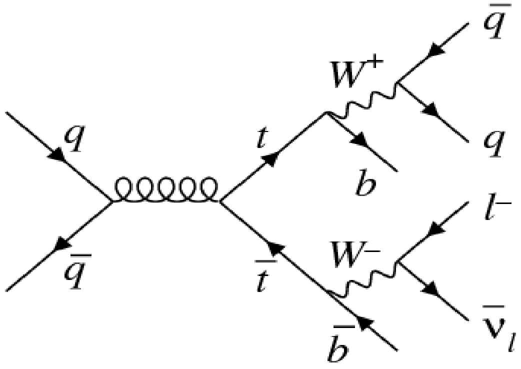


Fig.6 top pair, σ = 452.36 pb

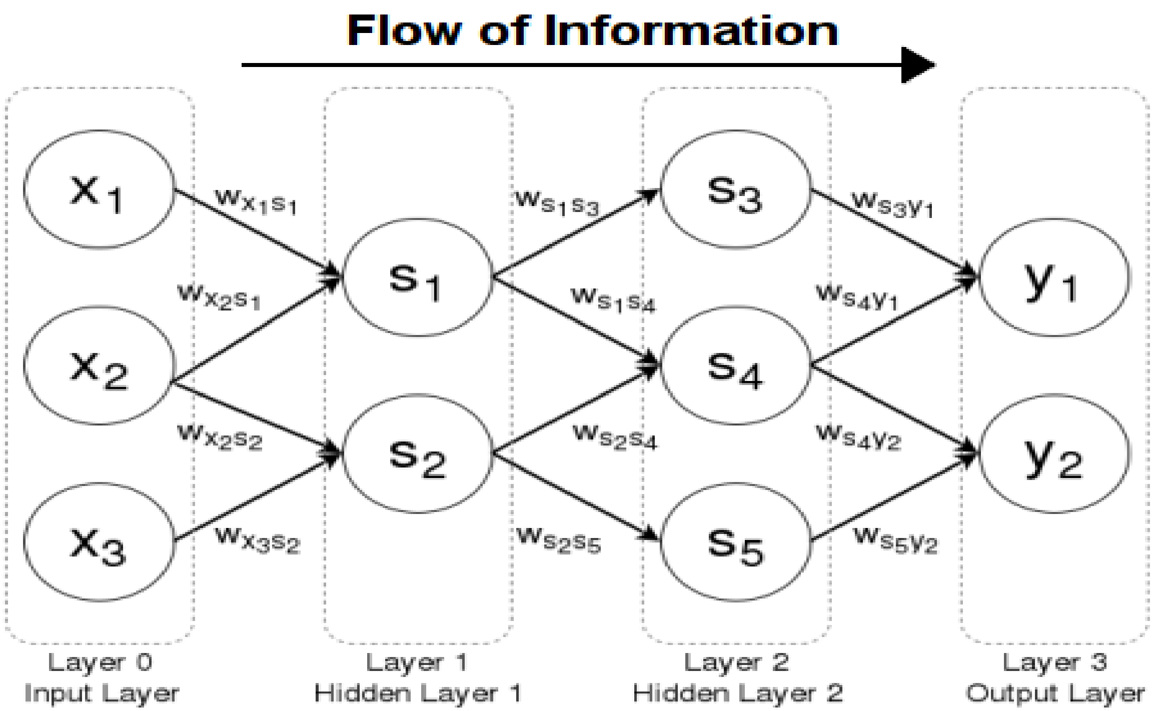
Project goals

- **Goals**
  - Use a NN to optimize the separation between the signal and the different backgrounds
  - Find the best set of hyperparameters to achieve the best performance

- **Task**
  - Develop two NN’s classifiers:
    - Binary classification (Sig vs Bkg)
    - Multiclass classification (Sig vs Bkg1 vs … vs Bkg5) for signal study and background control

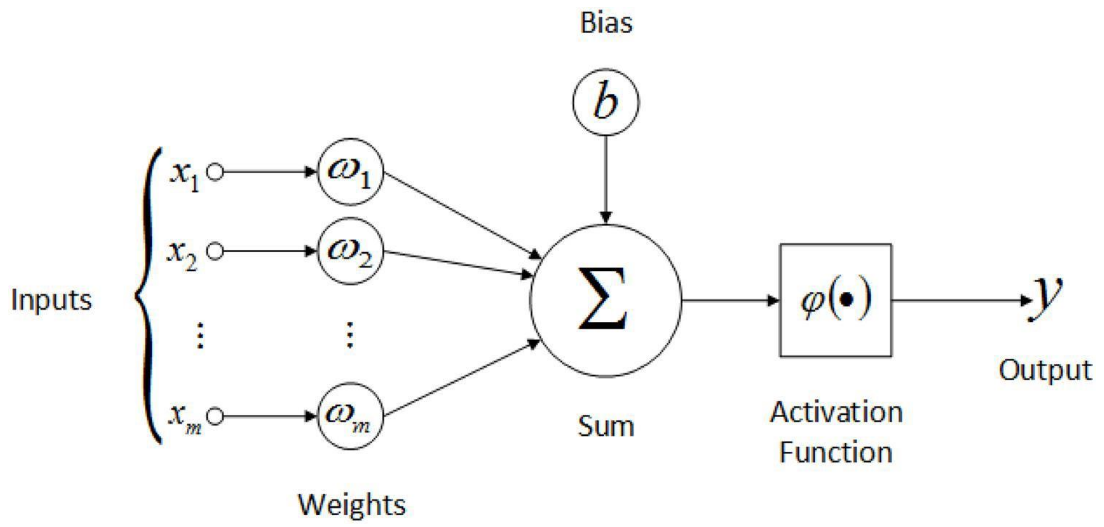
- **Data**
  - Samples produced via Monte Carlo generator (full ATLAS simulation) with several background channels (W+jets, single top, $t\bar{t}$, WlvZqq, WqqWlv)

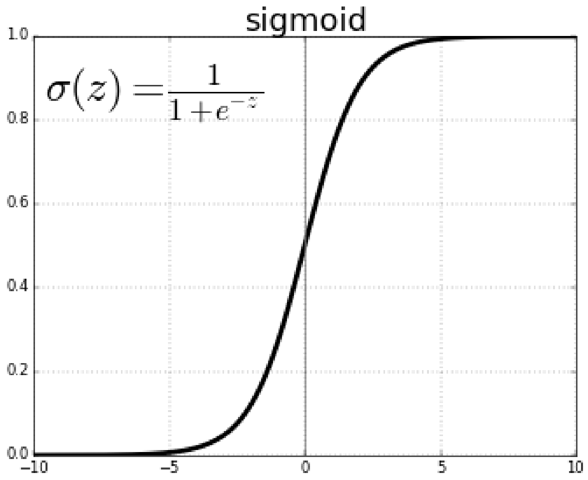
FNN - Architecture

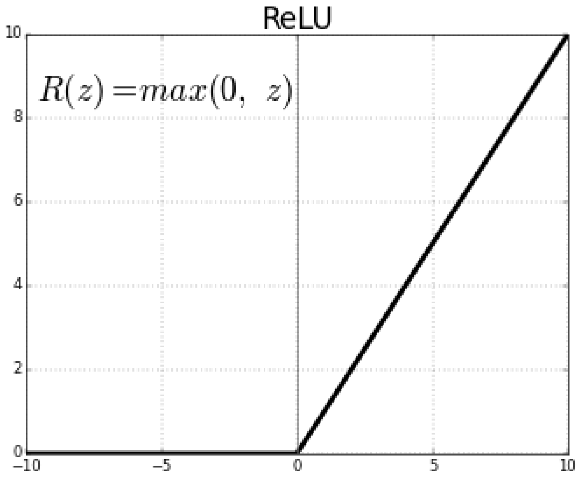


- Architecture:
  - Connections between the nodes do not form a cycle
  - The information moves only one direction , **forward** (from the input nodes, through the hidden nodes and to the output nodes)

Feedforward NN (FNN) - Neuron







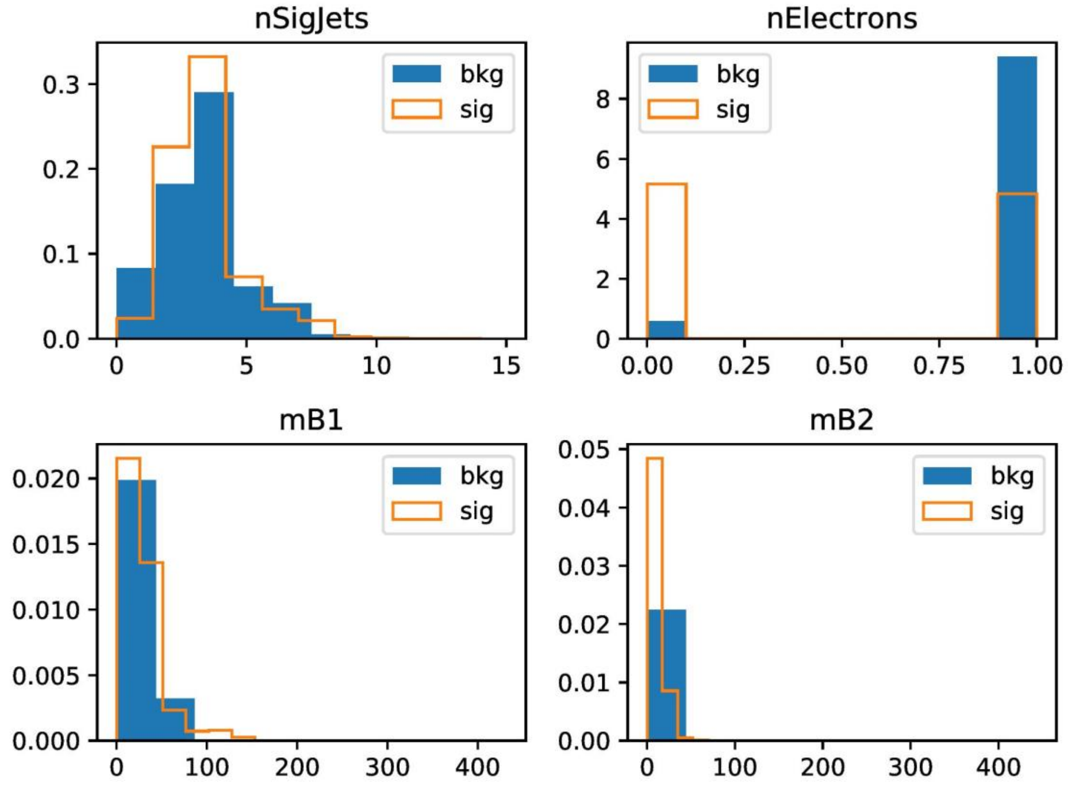
- **Each neuron:**
  - Is a function that takes all the previous neuron's output as inputs
  - Applies a factor (weight) to each input
  - Applies the sum to its activation function
  - Resulting into one output

Input Variables

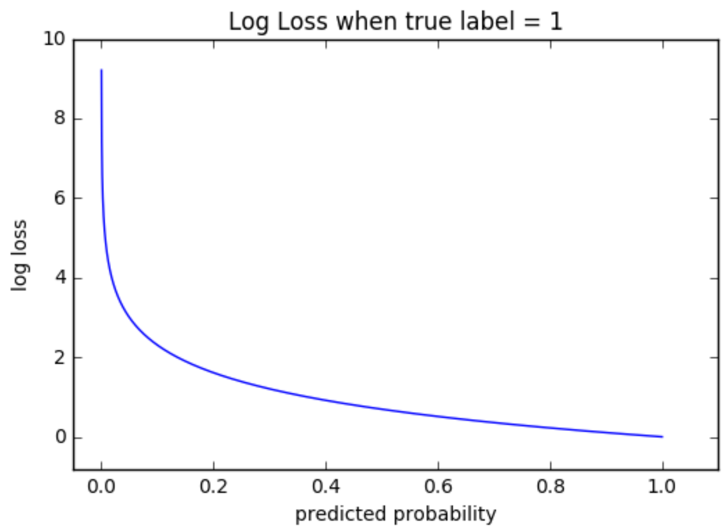
- 53 preprocessed input variables:

  - standardize features by removing the mean and scaling to unit variance

  - principal component analysis (PCA):
    - Linear dimensionality reduction of the data to project it to a lower dimensional space



FNN - Training



- Minimizing the loss function
  - Binary cross entropy

$$L(y,\hat{y}) = -\frac{1}{N}\sum_{i=0}^{N} y_i * log(\hat{y}_i) + (1-y_i) * log(1-\hat{y}_i)$$

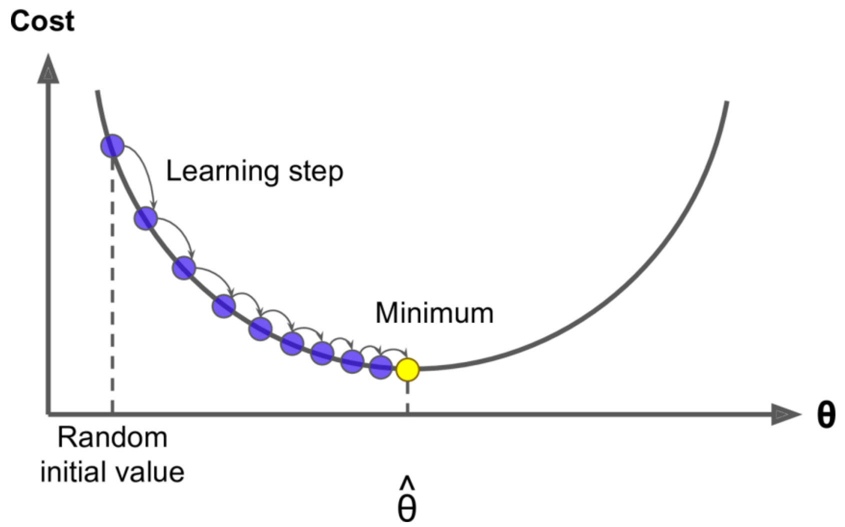
($\hat{y}$ is the NN's output and y the label)

  - Categorical cross entropy

$$L(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N} y_{ij} * log(\hat{y}_{ij})$$

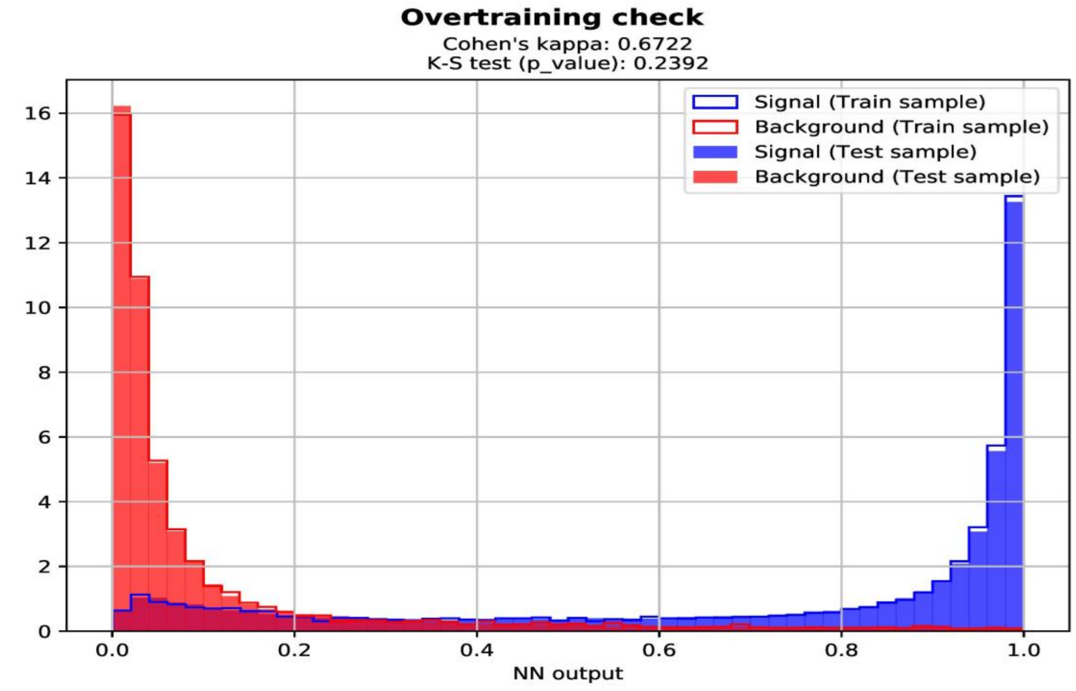
- Learning Rate or Step Size

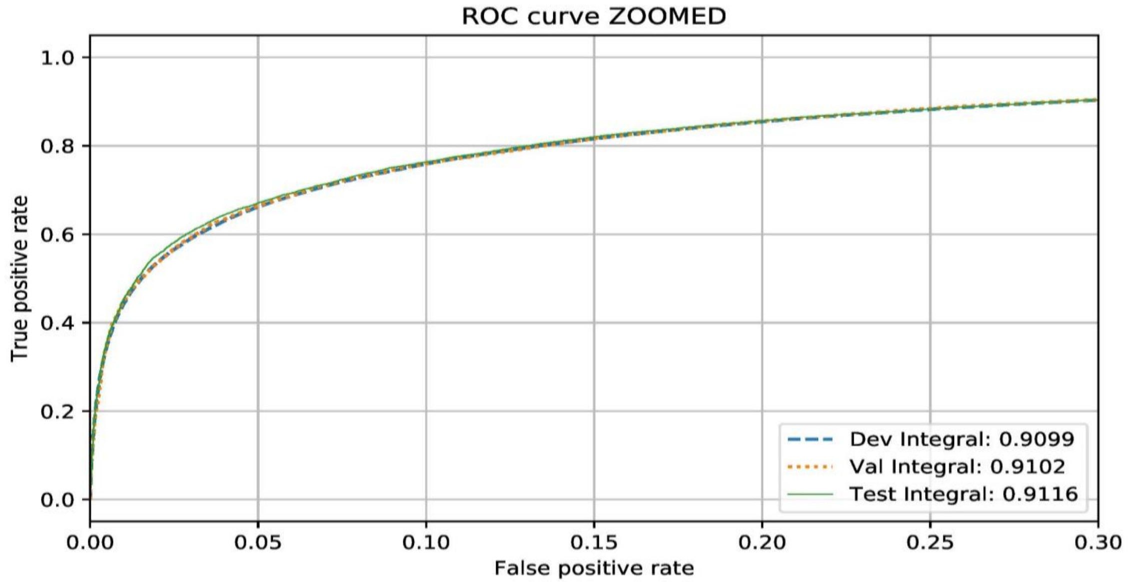
$$w_i' = w_i + \lambda \frac{dL}{dw_i}$$



Method

1. Try an architecture
   - Adapt learning rate
   - Adapt weight initializer and activation function
   - Repeat
2. Checks
   - Take most promising models
   - Check overtraining (training vs test samples)
   - Average (Statistical fluctuations)
     - Run same configuration x5
3. Compare models' AUCs and Cohen's Kappa
   - Compare the highest values
   - Check if AUC is higher in desired region or not



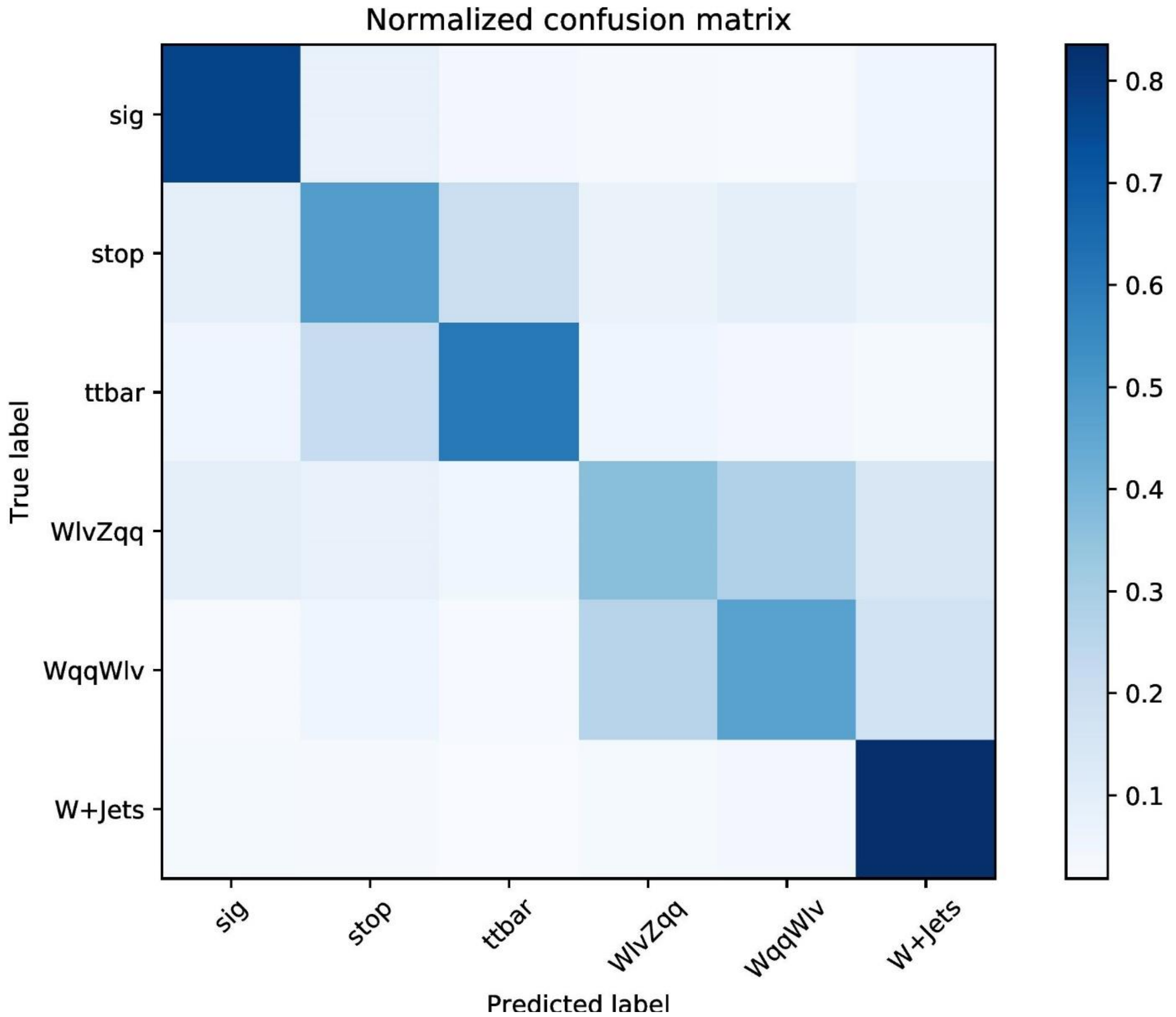


Results - Models

Model	Accuracy TEST	Loss TEST	ROC AUC TEST	Cohen's Kappa	Fraction	Lerning Rate	Epocs	Neuron-Layers	Activation	Weight Initializer
1	0.8696	1.5817E-06	0.9419	0.7085	0.3	0.01	28	53 71 71 71 1	relu	he_normal
2	0.8692	1.5860E-06	0.9417	0.7076	0.3	0.01	28	53 71 71 71 71 1	relu	he_normal
3	0.8709	1.5942E-06	0.9413	0.7118	0.3	0.01	28	53 71 71 71 71 71 1	relu	he_normal
4	0.8684	1.5815E-06	0.9419	0.7051	0.3	0.01	30	53 71 51 41 31 21 11 1	relu	he_normal
5	0.8712	1.5552E-06	0.9429	0.7122	0.3	0.01	39	53 71 51 41 31 21 11 1	selu	lecun_normal
6	0.8680	1.5780E-06	0.9422	0.7038	0.3	0.01	28	53 71 71 71 71 71 1	relu	he_normal
28	0.8717	1.1650E-06	0.9430	0.7123	0.4	0.01	33	53 71 71 71 71 71 1	relu	he_normal

Multiclass classifier

- Classify the events in six categories:
  - Signal
  - W+jets, ttbar, single top, WW, WZ
- Optimized the NN parameters in the same way as before
- Confusion matrix almost diagonal: the NN identifies the different classes
- Higher confusion in separating
  - Single top from ttbar
  - WW from WZ
  - Very similar processes!!
- The results suggest that three background categories would be better



Conclusions

- Developed and optimized ML algorithm to separate high momentum Higgs to bb decays from background
  - Achieved accuracy ~ 87 %
  - Best model overall is the model 28:
    - Activation for hidden layers: ReLu
    - Weight initializer: he_normal
    - Adaptive Learning Rate with decay rate = 0.1
- Went further to identify the different backgrounds with multiclass NN
  - Achieved accuracy ~ 65 %
  - Multiclass output suggests to group background in 3 categories only
- Next steps
  - Compare signal significance between binary classifier and multiclass model

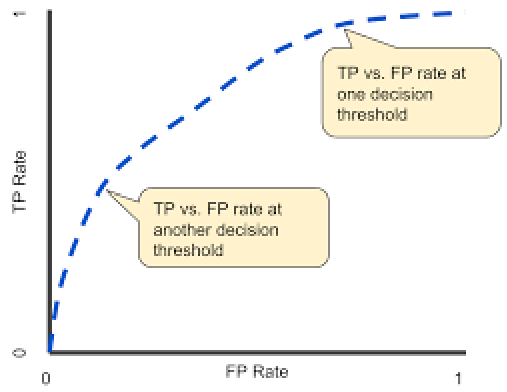
Backup Slides

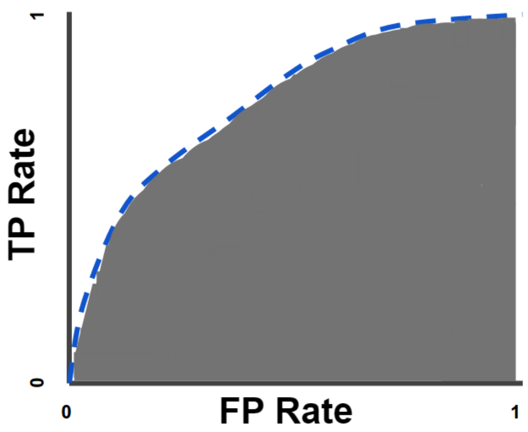
ROC - AUC

- ROC curve (receiver operating characteristic curve)

$$TPR = \frac{TP}{TP+FN} \qquad FPR = \frac{FP}{FP+TN}$$

- AUC (Area Under the ROC Curve)
  - measure of performance across all possible classification thresholds





Cohen's kappa

- Measure of agreement between the two individuals (NN's predictions and true labels)
  - p_0 observed level of agreement
  - p_e the value that you would expect if the raters were totally independent

		Rater #1 1	Rater #1 2	Total
Rater #2	1	p_{11}	p_{12}	$p_{1.}$
	2	p_{21}	p_{22}	$p_{2.}$
	Total	$p_{.1}$	$p_{.2}$	1

$$p_o = p_{11} + p_{22}$$

$$p_e = p_{.1}p_{1.} + p_{.2}p_{2.}$$

Agreement	Value for Kappa
Poor	Less then 0.20
Fair	0.20 to 0.40
Moderate	0.40 to 0.60
Good	0.60 to 0.80
Very good	0.80 to 1.00

$$k = \frac{p_o - p_e}{1 - p_e}$$

Conclusions

- ➢ New NN architectures (N layers, M nodes) were developed to separate signal from background with
  - Different activation functions
  - Different (Weight Initializer)

- ➢ In doing so, the search was guided by two criteria:
  - Performance: checked by Cohen's Kappa, AUC and ROC curve
  - Validity: checked by over training test
- ➢ Best model overall is the model 28:
  - Activation for hidden layers: ReLu
  - Weight initializer: he_normal
  - Adaptive Learning Rate with decay rate = 0.1