Neural network development in the analysis of single top-quark production in association with a Higgs boson and light-quark at ATLAS

Christian Kirfel

15th of March 2020

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

General ML selection

- n-jets: 2 (b-jets: 1)
- b-jet WP: 70 DL1r
- nLeptons & nTaus: $1\boldsymbol{e}/\boldsymbol{\mu}$ $2\boldsymbol{\tau}_{\text{had}}$
- $\boldsymbol{E}_{\text{T,miss}}$: no cut (to 800 GeV)

- jets:
  - $p_T > 35$ GeV
  - $|\eta| < 4.5$
  - EMPFlow
- electrons:
  - $p_T > 20$ GeV leading 27 GeV
  - $|\eta| < 2.5$ not in 1.37 - 1.52
  - WP: LooseAndBLayerLH ; isolation: no requirement
- muons:
  - $p_T > 20$ GeV leading 27 GeV
  - $0.01 < |\eta| < 2.5$
  - WP: Loose ; isolation: no requirement
- taus:
  - $p_T > 20$ GeV leading 27 GeV
  - $|\eta| < 2.5$ not in 1.37 - 1.52
  - WP: RNNLoose
  - ASG recommended OLR (τ_{had} remove jets)

Challenges

Negative weights in Monte Carlo

- Negative weights arise from
- In neural network training negative weights need to an unwanted behaviour
- The impact of the treatment of negative weights on the training has to be explored

Different sizes of background processes

- Dominating background can diminish the significance of secondary backgrounds in Training
- Especially very signal-like backgrounds are likely to be mislabed

Accelerating network optimisation

- Exploration of new features is only possible in an optimised network
- An evolutionary optimisation approach minimises the work effort

Background processes in neural network training

- Impact we see
- Way to treat it

Info about negative weights

- serf

Impact of negative weights on the training

- sbhgfiuw

Initial problems

Obstacles of optimisation processes

- Grid searches are both tedious and ressource intensive
- Knowledge gained in one problem is rarely universal
- Experienced users can develop a bias towards certain hyperparamters

Applications in

- A neural network should be developed in parallel to an ongoing analysis.
- New additions need a new optimisation.

Training specifications

- Training a deep neural network
- Coarse optimisation using an evolutionary neural network
- Fine optimisation doing a grid search
- Optimised hyperparameters:
  - Number of nodes
  - Number of layers
  - Dropout percentage
  - Activation function
  - Weight initialisation
  - Optimiser
- Signal: tHq, tZq
- Background: Diboson, Z+jets, ttbar
- Using absolute weights

Evolutionary optimisation of neural networks

- Combination of a grid searches with a survival of the fittest setup
- Start with a number of random hyperparamters
- Evaluate the set of hyperparameters
- Create a a new set of networks based on the previous best
- Add recombination and variation to avoid local minima and bias

Setup

- $\mathrm{tZ}q$ as signal against $\mathrm{t\bar{t}}$ as background
- Using basic kinematic variables
- No specific region
- Training without weights due to some recent problems
- Testing: nodes, layers, dropout
- Fixed hyperparameters:
  - Optimizer: Adam
  - Activation: relu, sigmoid
  - Batchsize: 1000
  - Epochs: 25

An example of parameter development

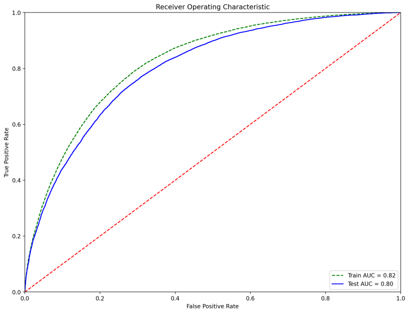
Initial parameters

- Layers: $1 - 10$
- Nodes: $1 - 100$
- Dropout: $0 - 1$

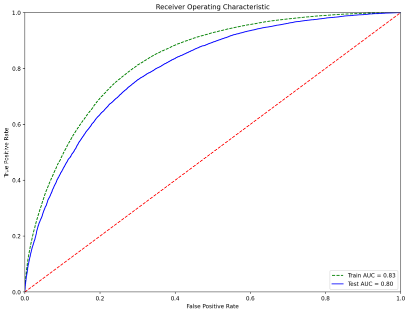
Final parameters

- Layers: 4 ± 2
- Nodes: 67 ± 33
- Dropout: 0.4 ± 0.3

Comparing to a grid search

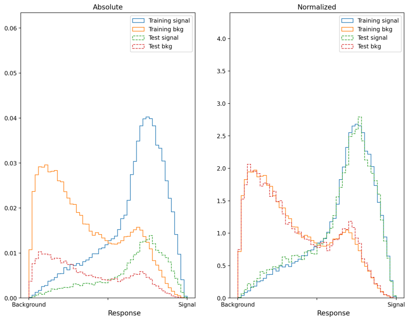


Grid search

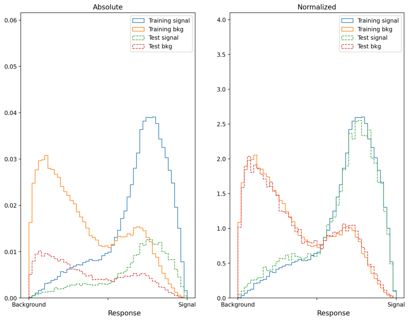


Evolutionary search

Comparing to a grid search



Grid search



Evolutionary search

Conclusions

- ding

Setup on baf

- Create a random set of hyperparameters
- Submit a job to baf for each configuration
- Use dagman to wait for the jobs to finish
- Evaluate the results and create the next set of configurations
- Anji's talk
- Confluence link

Sources