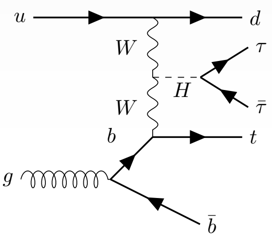
Neural network development in the analysis of single top-quark production in association with a Higgs boson and light-quark at ATLAS

Christian Kirfel

15th of March 2020

BMBF-ErUM-Forschungsschwerpunkt
ATLAS-EXPERIMENT
Ausbau von ATLAS am LHC: Physik mit dem ATLAS-Experiment
ErUM-FSP T02
ATLAS

tHq ditau hadhad channel selection



- n-jets: 2 (b-jets: 1)
- b-jet WP: 70 DL1r
- nLeptons & nTaus: 1e/μ 2τ_{had}
- $E_{\text{T,miss}}$: 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

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 mislabeled

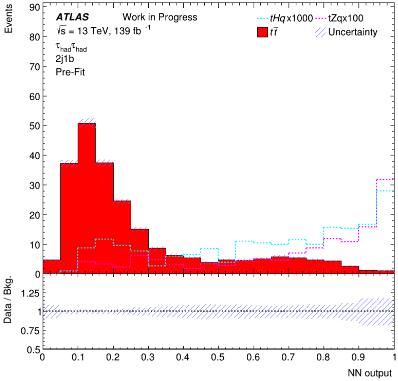
Negative weights in Monte Carlo

- In neural network training negative weights lead to an unwanted behaviour

Accelerating network optimisation

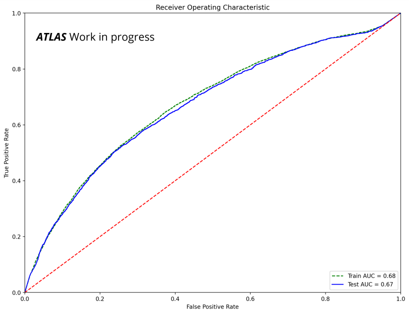
- Exploration of new features is only possible in an optimised network
- An evolutionary optimisation approach minimises the work effort

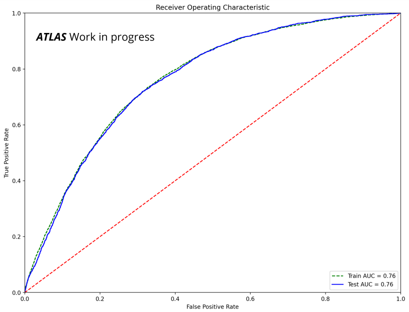
Background processes in neural network training



- $t\bar{t}$ dominates the training
- tZq classified as signal
- Approaches:
  - multiple networks
  - multiple targets
  - reweighting samples

Impact of negative weights on the training





- About 35% negative weights
- Breaks the networks training
- Possible approaches: absolutes or just positive weights for training

Problems in network optimisation

Obstacles of optimisation processes

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

Applications of evolutionary optimisation

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

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

Example of an evolutionary training

- Signal: $\mathrm{tH}q$
- Background: $\mathrm{t\bar{t}}$
- Testing: nodes, layers, dropout
- Fixed hyperparameters:
  - Optimizer: Adam
  - Activation: relu, sigmoid
  - Batchsize: 1000
  - Epochs per generation: 25

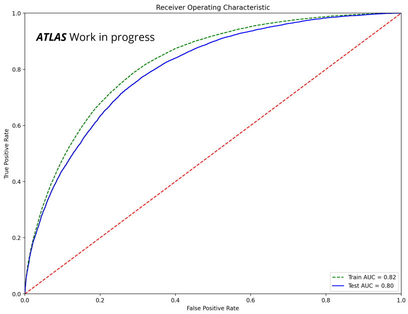
Initial parameters

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

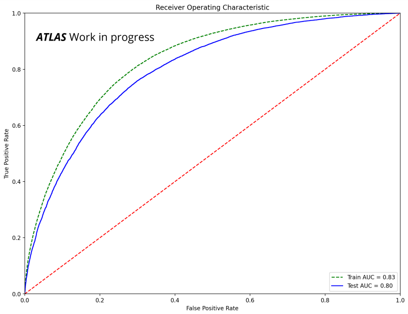
Final parameters

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

Comparing ROC to a grid search

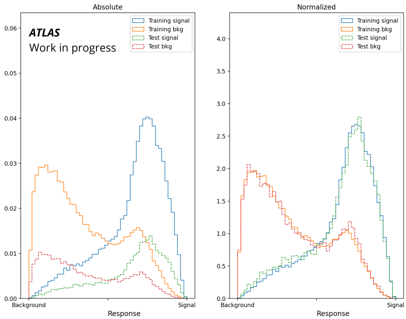


Grid search

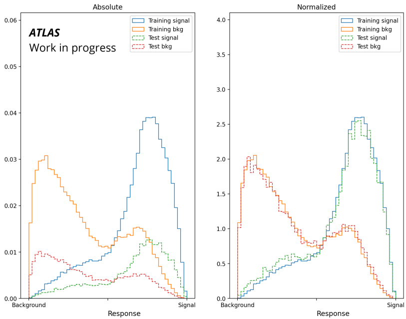


Evolutionary search

Comparing response to a grid search



Grid search



Evolutionary search

Conclusions

- Using neural networks for signal to background separation in the $\mathrm{tH}q$ channel is an interesting analysis that offers many challenges.
- Negative weights resulting from Monte Carlo generators cannot easily be handled by machine learning algorithms.
- Different sizes of background samples result in smaller backgrounds getting ignored or even classified as signal
- Evolutionary optimisation of neural networks is a nice approach to reoptimise a network automatically in an ongoing analysis.