Deep Learning Top Tagger
- or the end of QCD?

Gregor Kasieczka
CERN IML Workshop
2017-03-22
A Very Simple Network

\[ y = f(f(x_1)w_1 + f(x_2)w_2) \]
\[ f(x) = \Theta(x) \cdot x \]
How do networks learn?

- Backpropagation + Gradient descent
- Pass input \((x_1, x_2)\) to ANN
- Calculate output \((y)\) and difference to true value \((\hat{y})\)
  This is the loss function \(L\)
- Find gradient of loss function with respect to weights
- Use gradient to find new weights

\[
L(y, \hat{y}) = (y - \hat{y})^2
\]

\[
w'_i = w_i + \alpha \cdot \frac{\partial L}{\partial w_i}
\]
Convolutional Network

Fold a mask with the input to get output
What is learned are the parameters of the mask
Convolutional (conv) layer

That’s the weights we want to train

• How to build a convolutional network
  • Chain multiple conv layers
  • Reduce image resolution in between (optional)
  • Use multiple masks per layer
  • Add linear ANN in the end (optional)

(This is still a network. We just use a fancy idea to decide which nodes to connect to each other)
Deep-learning Top Taggers or The End of QCD?

Gregor Kasieczka,\textsuperscript{1} Tilman Plehn,\textsuperscript{2} Michael Russell,\textsuperscript{3} and Torben Schell\textsuperscript{2}

\textsuperscript{1}Institute for Particle Physics, ETH Zürich, Switzerland
\textsuperscript{2}Institut für Theoretische Physik, Universität Heidelberg, Germany
\textsuperscript{3}School of Physics and Astronomy, University of Glasgow, United Kingdom
Problem

• Identify hadronically decaying top quarks from QCD jets
• $p_T$ large enough so that the top quark is contained in one jet (boosted topology)
• Many QCD inspired variables available
  • jet mass = top mass
  • also find $W$ mass inside jet
  • n-subjettiness (3-prong structure of the top)
• \( \rightarrow \) Top tagging is a well understood problem to test DNNs on

Wednesday 22 March 17
Technical interlude

- 14 TeV hadronic $t\bar{t}$ vs QCD, both simulated with Pythia 8
- Delphes 3 detector simulation
- Cluster with Anti-$k_T$(R=1.5), recluster with Cambridge/Aachen (R=1.5)
- Jet $p_T$: 350..450 GeV, $|\eta| < 1.0$
- Signal is truth matched to top within Delta R < 1.2
- Samples (signal+background):
  - 150k+150k for training
  - 150k+150k for checks during training
  - 300k+300k for final test
Image approach

- Jets = 2d grayscale images:
  - 1 pixel = 0.1 in eta, 5 degree in phi
  - pixel energy: calorimeter ET

- Preprocessing
  - Center maximum
  - Rotate so that second maximum is 12 o’clock
  - Flip so that third maximum is on the right side
  - Crop to 40x40 pixels

Overlay of 10k images
Network architecture

Inputs 1@40x40

Feature maps 8@39x39

Feature maps 8@38x38

Feature maps 8@18x18

Feature maps 8@17x17

Hidden units

Hidden units

Hidden units

Outputs

Convolution 4x4 kernel

Convolution 4x4 kernel

MaxPooling

Convolution 4x4 kernel

Convolution 4x4 kernel

Flatten

Fully connected

Fully connected

Fully connected

Disadvantage of DNN approach: choosing network architecture is a bit of voodoo

This seems to work though
classification is a parameter that allows to link the signal efficiency $S$ with the mis-tagging rate of background events $B$.

In Sec. III we will use this trained network to test the performance in terms of ROC curves, correlating the signal efficiency and the mis-tagging rate.

Before we move to the performance study, we can get a feeling for what is happening inside the trained ConvNet by looking at the output of the different layers in the case of fully pre-processed images. In Fig. 5 we show the difference of the averaged output for 100 signal and 100 background images. For each of those two categories, we require a classifier output of at least 0.8. Each row illustrates the output of a convolutional layer. Signal-like red areas are typical for jet images originating from top decays; blue areas are typical for backgrounds. The first layer seems to consistently capture a well-separated second subjet, and some kernels of the later layers seem to capture the third signal subjet in the right half-plane. However, one should keep in mind that there is no one-to-one correspondence between the location in feature maps of later layers and the pixels in the input image.
Pearson correlation coefficient

\[ r_{ij} = \frac{\sum_{\text{images}} (x_{ij} - \bar{x}_{ij}) (y - \bar{y})}{\sqrt{\sum_{\text{images}} (x_{ij} - \bar{x}_{ij})^2} \sqrt{\sum_{\text{images}} (y - \bar{y})^2}} \]
Performance

- Train a BDT on a set of standard tagging variables

SoftDrop + n-subjettiness:
\[ \{ m_{sd}, m_{fat}, \tau_2, \tau_3, \tau_{2sd}^{sd}, \tau_{3sd}^{sd} \} \]

MotherOfTaggers:
\[ \{ m_{sd}, m_{fat}, m_{rec}, f_{rec}, \Delta R_{opt}, \tau_2, \tau_3, \tau_{2sd}^{sd}, \tau_{3sd}^{sd} \} \]
Technical Interlude II

- Training done on Piz Daint nodes equipped with *Nvidia Tesla P100 GPUs* at CSCS
  - Applied for small development project of 25k node hours for a year
  - Also played with Amazon cloud (very easy and convenient, but they want money!)
- Network trains in $O(\text{some hours})$
  - *Depends on $N(\text{events})$, network architecture, learning parameters. Can probably optimize a bit*
- Use
  - Keras with Theano backend for DNN
  - scikit-learn for BDT
  - NumPy and Pandas for processing & storage
Appendix A: What the machine learns

For our performance comparison of the QCD-based tagger approach and the neural network it is crucial that we understand what the DeepTop network learns in terms of physics variables. The relevant jet substructure observables differentiating between QCD jets and top jets are those which we evaluate in the MotherOfTaggers BDT, Eq. (17).

To quantify which signal features the DNN and the BDT tagger have correctly extracted we show observables for signal event correctly identified as such, i.e. requiring $y > 0$. Following Fig. 3 this cut value captures a large fraction of correctly identified events. The same we also do for correctly identified background events with $y < 0$.

The upper two rows in Fig. 10 show the different mass variables describing the fat jet. We see that the DNN and the BDT tagger results are consistent, with a slightly better performance of the

$\text{Sliced Masses}$

Compare:
Selection on truth
Selection using BDT ($<0.2$ / $>0.8$)
Selection using DNN (($<0.2$ / $>0.8$)
Train and test network on images with higher pixel threshold (and set values below threshold to zero)

We are not sensitive to very soft information (this is good)
Detector effects

- **Default**
  - Train network on JES=1.0
  - Test network on images with JES scaled up/down
- **Hardened**
  - Train network while randomly smearing JES
  - Test network on images with JES scaled up/down

*Control what the network is sensitive to*
Conclusions

• DNNs for top jet tagging work, achieve decent performance and are surprisingly stable

• Techniques available to understand what it going on

• More things to try out (data, regression, RNNs, multi-object,..)

• Ready for physics analysis?
Bonus Slides
Appendix A: What the machine learns

For our performance comparison of the QCD-based tagger approach and the neutral network it is crucial that we understand what the DeepTop network learns in terms of physics variables. The relevant jet substructure observables differentiating between QCD jets and top jets are those which we evaluate in the MotherOfTaggers BDT, Eq.(17). To quantify which signal features the DNN and the BDT tagger have correctly extracted we show observables for signal event correctly identified as such, i.e. requiring $y > 0.8$ for the signal events. Following Fig. 3 this cut value captures a large fraction of correctly identified events. The same we also do for correctly identified background events with $y < 0.2$.

The upper two rows in Fig. 10 show the different mass variables describing the fat jet. We see that the DNN and the BDT tagger results are consistent, with a slightly better performance of the mother of taggers.