Machine learning techniques for jet flavour identification at CMS

Mauro Verzetti
on behalf of the CMS Collaboration
The problem

AK8 jet
$E_T = 2009$ GeV
$\eta = -0.65$
$\phi = -2.30$

$\Lambda K8$ jet
$p_T = 2088$ GeV
$\eta = 0.63$
$\phi = 0.84$

Muon
$p_T = 20.1$ GeV
$\eta = -0.64$
$\phi = -2.27$
The problem
HF tagging @ CMS — JINST 13 (2018) no.05, P05011

Soft-lepton (SL) based tagger

Super combined

Secondary vertex (SV) based tagger

combined

Track-based tagger

PV

Track selection

Tracks

Jet
DeepCSV

Charged (8 features) x 6
Secondary Vtx (8 features) x 1
Global variables (12 features)

Dense
100 nodes x 5 layers

Output classes:
b, bb, c, l

5.1 The $b$ jet identification

In this figure, the tagging efficiency is integrated over the $p_T$ and $h_d$ distributions of the jets in the $t_t$ sample. The tagging efficiency is also shown for the Run 1 version of the CSV algorithm. It should be noted that the CSV algorithm was trained on simulated multijet events at centre-of-mass energy of 7 TeV using anti-$k_T$ jets clustered with a distance parameter $R = 0.5$. Therefore, the comparison is not completely fair. The performance improvement expected from a retraining is typically of the order of 1%. The absolute improvement in the $b$ jet identification efficiency for the CSVv2 (AVR) algorithm with respect to the CSV algorithm is of the order of 2–4% when the comparison is made at the same misidentification probability value for light-flavour jets. An additional improvement of the order of 1–2% is seen when using IVF vertices instead of AVR vertices in the CSVv2 algorithm. The cMVAv2 tagger performs around 3–4% better than the CSVv2 algorithm for the same misidentification probability for light-flavour jets. The DeepCSV $P(b) + P(bb)$ tagger outperforms all the other $b$ jet identification algorithms, when discriminating against $c$ jets or light-flavour jets, except for $b$ jet identification efficiencies above 70% where the cMVAv2 tagger performs better when discriminating against light-flavour jets. The absolute $b$ identification efficiency improves by about 4% with respect to the CSVv2 algorithm for a misidentification probability for light-flavour jets of 1%. Three standard working points are defined for each $b$ tagging algorithm using jets with $p_T > 30$ GeV in simulated multijet events with $80 < \hat{p}_T < 120$ GeV. The average jet $p_T$ in this sample of events is about 75 GeV. These working points, "loose" (L), "medium" (M), and "tight" (T), correspond to thresholds on the discriminator after which the misidentification probability is around 10%, 1%, and 0.1%, respectively, for light-flavour jets. The efficiency for correctly identifying $b$ jets in simulated $t_t$ events for each of the three working points of the various taggers is summarized in Table 2.

The tagging efficiency depends on the jet $p_T$, $h$, and the number of pileup interactions in the event. This dependency is illustrated for the DeepCSV $P(b) + P(bb)$ tagger in Fig. 17 using...

JINST 13 (2018) no.05, P05011
Trying more complex architectures

- Convolutional NN successfully applied in neutrino physics and image recognition
- Some proposals to treat jets as images

Boosted W

... but
- Jets do not look like normal images!
- CMS events are way more complex and bring more information than a flat image (e.g. tracking information)
Particle-based NN architecture

- Charged (16 features) x25
- Neutral (8 features) x25
- Secondary Vtx (12 features) x4
- Global variables (6 features)

1x1 conv. 64/32/32/8 → RNN 150
1x1 conv. 32/16/4 → RNN 50
1x1 conv. 64/32/32/8 → RNN 50

Dense
200 nodes x1, 100 nodes x6

M. Verzetti (CERN and FWO)
Particle-based NN architecture

Convolutional layers progressively learn a more compact feature representation (automatic feature engineering)

M. Verzetti (CERN and FWO)
Particle-based NN architecture

Charged (16 features) x25
Neutral (8 features) x25
Secondary Vtx (12 features) x4
Global variables (6 features)

1x1 conv. 64/32/32/8
1x1 conv. 32/16/4
1x1 conv. 64/32/32/8

RNN 150
RNN 50
RNN 50

Dense 200 nodes x1, 100 nodes x6

Convolutional layers progressively learn a more compact feature representation (automatic feature engineering)

The recurrent layers (LSTM) builds a “summary” of the information contained in each set of feature types

M. Verzetti (CERN and FWO)
Particle-based NN architecture

- Charged (16 features) x25
- Neutral (8 features) x25
- Secondary Vtx (12 features) x4
- Global variables (6 features)

1x1 conv. 64/32/32/8 -> RNN 150
1x1 conv. 32/16/4 -> RNN 50
1x1 conv. 64/32/32/8 -> RNN 50

Dense
200 nodes x1, 100 nodes x6

CMS Simulation Preliminary
$\sqrt{s} = 13$ TeV

misid. probability

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

b jet efficiency

CMS Simulation Preliminary
$\sqrt{s} = 13$ TeV

misid. probability

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

b jet efficiency

CMS DP-2018/033
Particle-based NN architecture

Charged (16 features) x25
Neutral (8 features) x25
Secondary Vtx (12 features) x4
Global variables (6 features)

1x1 conv. 64/32/32/8 → RNN 150
1x1 conv. 32/16/4 → RNN 50
1x1 conv. 64/32/32/8 → RNN 50

Dense 200 nodes x1, 100 nodes x6

Similar performance to simpler, dedicated binary taggers, but with full multi-class power.

Significantly better performances in given regions with different quark composition

CMS-DP-2017-027
Significantly larger amount of candidates used to accommodate for 90% of the fat jets

Need to learn substructure from both charged and neutral candidates

RNNs become computationally too expensive to train

Use particle-level convolutional layers (P-CNN) where each feature is treated as a “colour”
Performance

• Flavour information largely improves jet tagging

• Large improvement w.r.t to the BDT approach

• Introduces mass sculpting, not necessarily a bad thing

Figure 1. Comparison of the performance of the two BDT taggers and the two particle-based CNN taggers in terms of ROC curves in MC simulated events for top jets as signal and QCD jets as background. The events correspond to AK8 jets with \(1000 < p_T < 1400\) GeV and \(|\eta| < 1.5\).

CMS-DP-2017-049
DeepDoubleB

Conv1D + GRU network topology

- 27 high-level (double-b) features
- 60 $\times$ 8 track features
- 5 $\times$ 2 secondary vertex features per Higgs-candidate jet

- BatchNormalization (BN) to process inputs
- Conv1D with kernel size 1 = Time-distributed dense = apply same dense network to each PF candidate / track / SV
- GRU = Gated Recurrent Unit = Recurrent network to reduce dimensionality of output from Conv1D layers

SV features

Double-b features

Output

Tagging efficiency ($H \rightarrow b\bar{b}$)

Mistagging rate (QCD)

DeepDoubleBvL, AUC = 97.3%
double-b, AUC = 91.3%

Significantly better than current BDT approach!

Some mass sculpting

CMS DP-2018/046

M. Verzetti (CERN and FWO)
Removing mass correlation

Per-batch penalty term proportional to the Kullback-Liebler (KL) divergence

**Figure 7.** Performance of the new DeepDoubleBvL quark-antiquark pair jet identification algorithm and its mass decorrelated version demonstrating the probability of misidentifying QCD jets as a function of the tagging efficiency. These receiver operating characteristic curves are obtained from a combined sample of QCD and Hbb.

**Figure 6.** Effect on the jet soft-drop mass distribution of misidentified events by the DeepDoubleBvL identification algorithm, after it has been decorrelated from the jet mass demonstrating the degree to which the algorithm is dependent on the mass of the jet. These histograms are obtained for a fixed overall mistagging rate from a QCD sample.

**Minimal loss in performance**

**Mass sculpting gone**

**CMS DP-2018/046**
DeepDoubleC!

Figure 2. Performance of the double-b and the DeepDoubleBvL and DeepDoubleCvL quark-antiquark pair jet identification algorithms demonstrating the probability of misidentifying QCD jets as a function of the tagging efficiency. These receiver operating characteristic curves are obtained from a combined sample of QCD and Hcc.

Figure 3. Performance of the double-b and the DeepDoubleBvL and DeepDoubleCvB quark-antiquark pair jet identification algorithms demonstrating the probability of misidentifying Hbb jets as a function of the tagging efficiency. These receiver operating characteristic curves are obtained from a combined sample of Hbb and Hcc.

CMS Simulation Preliminary 2016 (13 TeV)

300 < jet $p_T$ < 2000 GeV
40 < jet $m_{SD}$ < 200 GeV

DeepDoubleBvL, AUC = 86.3%
DeepDoubleCvL, AUC = 91.4%
double-b, AUC = 73.7%

CMS DP-2018/046
Deploying the model — Two worlds colliding

Training / Analysis:
- Keras + TensorFlow
- Python-based
- Private productions
- Minimal interaction with ROOT
- Few processes, single threads
- Little memory constraints
- Expendable jobs

Production:
- Custom framework
- C++ based (speed!)
- Mostly ROOT-centric (at least I/O)
- Many processes, multiple threads
- Many other concurrent activities → memory constraints
- Processes cannot die (e.g. trigger)

Deployment of the model required extensive efforts to integrate in the core CMS Software, which now fully supports TensorFlow models.

Still looking into optimising the inference performance of the models through model distilling/pruning and AOT compilation.
Summary

- Jet tagging is of paramount importance for the CMS Physics program

- Lots of development in the last ~1.5 years to apply modern machine learning techniques to this field
  - Large improvements in performance
  - Still some room for new developments, especially in the boosted regime

- Flavour tagging is not only fancy algorithms, but solid and performing computing infrastructures as well
Backup
Figure 3: Performance of the $b$-jet identification algorithms demonstrating the probability for non-$b$ jets to be misidentified as $b$ jet, as a function of the efficiency to correctly identify $b$ jets. The curves are obtained on simulated $t\bar{t}$ events using jets within $|\eta| < 2.4$ and with $p_T > 30$ GeV. The $b$ jets from gluon splitting to a pair of $b$ quarks are considered as $b$ jets. The lines shown are for DeepCSV (retrained for the Phase 1 detector geometry), NoConv, and DeepFlavour. The NoConv algorithm serves only for comparison. The absolute performance in this figure serves as an illustration since the $b$-jet identification efficiency depends on the $p_T$ and $\eta$ distribution of the jets in the topology as well as the amount of $b$ jets from gluon splitting in the sample.
Realistic MC simulation

CMS Simulation Preliminary

$\sqrt{s} = 13$ TeV

$\bar{t}t$ events

AK4jets ($p_T > 30$ GeV)

- DeepFlavour phase 1
- DeepCSV phase 1
- DeepCSV phase 0

udsg

c
Mass Sculpting

Figure 5. Effect on the jet soft-drop mass distribution of misidentified events by the DeepDoubleCvL identification algorithm demonstrating the degree to which the algorithm is dependent on the mass of the jet. These histograms are obtained for a fixed overall mistagging rate from a QCD sample.
Particle-based NN architecture

Charged (16 features) x25

Neutral (8 features) x25

Secondary Vtx (17 features) x4

Global variables (6 features)

1x1 conv. 64/32/32/8

RNN 150

1x1 conv. 32/16/4

RNN 50

1x1 conv. 64/32/32/8

RNN 50

Dense

200 nodes x1, 100 nodes x6

b
bb
cc
l

Figure 3: Performance of the b jet identification algorithms demonstrating the probability for non-b jets to be misidentified as b jet, as a function of the efficiency to correctly identify b jets. The curves are obtained on simulated ttbar events using jets within $|\eta|<2.4$ and with $p_T>30$ GeV. The b jets from gluon splitting to a pair of b quarks are considered as b jets. The lines shown are for DeepCSV (retrained for the Phase 1 detector geometry), NoConv, and DeepFlavour. The NoConv algorithm serves only for comparison. The absolute performance in this figure serves as an illustration since the b jet identification efficiency depends on the $p_T$ and $\eta$ distribution of the jets in the topology as well as the amount of b jets from gluon splitting in the sample.

Figure 5: Performance of the DeepCSV (retrained for the Phase 1 detector geometry) and DeepFlavour b jet identification algorithms demonstrating the probability for non-b jets to be misidentified as b jet, as a function of the efficiency to correctly identify b jets. The curves are obtained on simulated QCD multijet events using jets within $|\eta|<2.4$ and with $300$ GeV < $p_T$ < $600$ GeV. The b jets from gluon splitting to a pair of b quarks are considered as b jets. The absolute performance in this figure serves as an illustration since the b jet identification efficiency depends on the $p_T$ and $\eta$ distribution of the jets in the topology as well as the amount of b jets from gluon splitting in the sample.
P-CNNs

\[ Z^{\alpha_m} = \sum_a \sum_j k^{\alpha}_{a,j} X_{a,(m+j-1)} \]
P-CNNs

\[ z^\alpha_m = \sum_a \sum_j k^\alpha_{a,j} x_{a,(m+j-1)} \]

- Loop over contiguous elements of the kernel
- Sweep over the elements
P-CNNs

Multiple features ("colours") are accounted computing the transformation

\[ Z^\alpha_m = \sum_a \sum_j k^{\alpha}_{a,j} \times a, (m+j-1) \]
P-CNNs

Different filters/kernels learn different transformations

\[ Z^{\alpha_m} = \sum_a \sum_j k^{\alpha}_{a,j} X_{a,(m+j-1)} \]
Deploying the model
Integration of DeepJet (AK4) into CMSSW. PR #19893

Tensorflow-based integration of new DeepFlavour tagger

#19893

Merged

cmsbuild merged 150 commits into cms-sw:master from pablodecm:ceep_flavour_tf_rebased_20_07 on 25 Jan

Conversation 830 Commits 150

Files changed 54

pablodecm commented on 25 Jul 2017 - edited

This pull request integrates the new DeepFlavour tagger, using the library CMSSW-DNN by @riga (the required part is also included) and adds it to the standard sequences. You can find an overview of the reason and design behind this PR in this BTV WG presentation.

PAT vs reference training framework (latest version)

Here are some checks of compatibility of CMSSW pat-based discriminators computed using the producers develop for this PR with the output from the training framework (DeepJet) as 2D histograms.
Integration of DeepJet (AK4) into CMSSW. PR #19893

Tensorflow-based integration of new DeepFlavour tagger
#19893

Merged: cmsbuild merged 150 commits into cms-sw:master from pablodecm:ceep_flavour_tf_rebased_26.07 on 25 Jan

pablodecm commented on 25 Jul 2017 - edited -

This pull request integrates the new DeepFlavour tagger, using the library CMSSW-DNN by @riga (the required part is also included) and adds it to the standard sequences. You can find an overview of the reason and design behind this PR in this BTV WG presentation.

PAT vs reference training framework (latest version)

Here are some checks of compatibility of CMSSW pat-based discriminators computed using the producers develop for this PR with the output from the training framework (DeepJet) as 2D histograms
Backend choice

**X** Interface based on TF python API:
- Uses python C API and a pre-built TF package
- Large overhead and no handle on memory/threading

**X** Interface based on TF C API:
- Low level and not very convenient
- Lots of customisations and ad-hoc handling needed

**✓** Interface based on TF C++ API:
- Access to all the needed internals for production usage with minimal need for custom code
- Shallow interface to connect TF to the CMSSW internals (e.g. logging)

For more information look [here](#).
Remaining issues

Multithreading:

- TF **loves** threads
- Normally a good thing, has a critical impact on memory consumption in HEP frameworks, which have their own thread schemes/pools (CMSSW uses TBB)
- Solved with the implementation of two custom sessions: **without threading** and **sharing the threading pool**

Memory footprint

- DeepJet model initially very big (~150MB)
- A careful optimisation for inference only can brought O(10-100) gain in memory reduction
- Further improvements from separating graph storage (common) and graph evaluation (one each thread)
- Exploring AOT compilation as future option