DeepJet: Generic physics object based jet multiclass classification for LHC experiments

Markus Stoye[*1], **Jan Kieseler**[1], **Huilin Qu**[2], **Loukas Gouskos**[2], **Mauro Verzetti**[1]
on behalf of the CMS collaboration

[1]CERN, European Organization for Nuclear Research, Switzerland
[2]University of California, Santa Barbara, USA
markus.stoye@cern.ch

Abstract

The detectors at the Large Hadron Collider at CERN reconstruct, among other objects, collimated sprays of particles, which are referred to as "jets". An important task is to identify the type of the elementary particle that initiated the jet, i.e. whether it is a light quark, a heavy quark or a gluon, leading to a multiclass classification problem. We present results from a realistic simulation of one of the two multi-purpose detectors at the LHC, the Compact Muon Solenoid. The basic network architecture relies heavily on using convolutional layers on low-level physics objects, like individual particle objects, and uses much more information than previous algorithms in the literature. It stands out as the first proposal that can be applied to multiclass classification for all types of jet initiators as well as for jets of different widths. We demonstrate significant improvements by the new approach on the classification capabilities for several of the tested particle classes. In one specific case, and at high momentum, a decrease of nearly 90% in the rate of false positives is achieved for a constant true positive rate of 40%.

1 Introduction

At the Large Hadron Collider (1) at CERN, where the Higgs boson was discovered leading to the 2013 Nobel prize, the reconstruction of jets is a key process. A jet is a particle spray that is initiated by quarks and gluons, which are elementary particles. Identifying the type of the elementary particle that initiated the jet is of crucial importance for physics analyses. Recently, several studies have proposed the use of deep neural networks for the classification of the elementary particles that initiate a jet (2; 3; 4; 5; 6; 7; 8). Jets typically contain O(10) to O(100) particles. The CMS (9) experiment builds particle objects (10) by combining information from all its detector systems, including the tracking detectors and the calorimeters. The information per particle object is high dimensional, heterogeneous, and differs for the different particle types. Figure 1 shows a slice of the CMS detector along with the different particle types and the signals that are observed in the detector. Some of the methods previously proposed have used either only a subset of the particles or only a subset of the information available for each particle. Others have used information from a single sub-detector, e.g. the calorimeters that are naturally pixelized, to create images, so that neural networks for images can then be applied. In both cases, only a relatively small subset of the available information provided in a jet is used in the classification. As a result, either these classifiers could be used only for a small subset of possible jet initiators or they were suboptimal.

The strategy presented here uses much more information, which is enabled by applying 1D convolutional layers on physics objects like particles. This expanded information allows for the classification of essentially all types of jet initiators and works for different cone sizes of jets. It is thus significantly more generic than any previous strategy.

Workshop on Deep Learning for Physical Sciences (DLPS 2017), NIPS 2017, Long Beach, CA, USA.

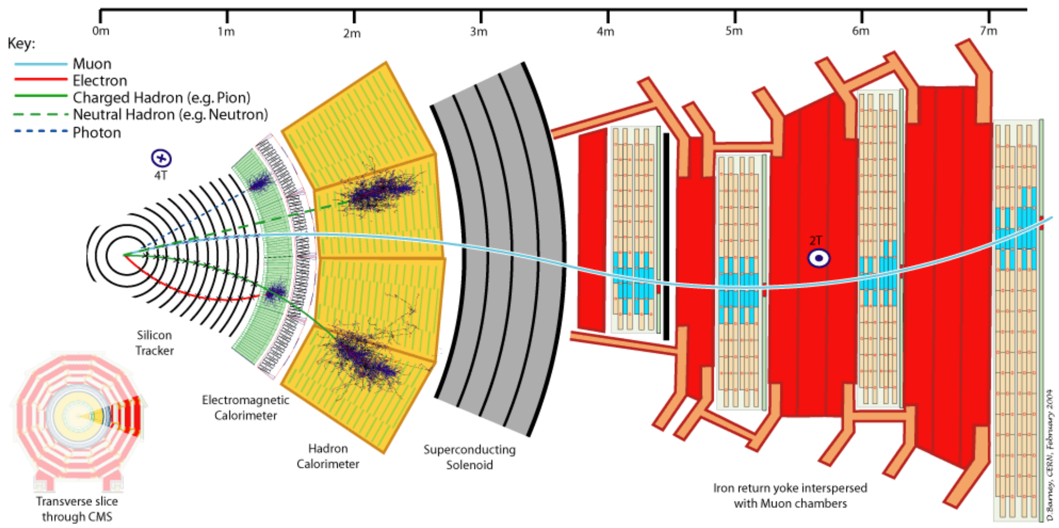
0m
1m
2m
3m
4m
5m
6m
7m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
Transverse slice through CMS
D Barney, CERN, February 2004

Figure 1: Slice of the CMS detector. It illustrates the different detector types and their different response to different particle types, such as muons or hadrons.

2 Neural network design

We define three lists of physics objects which are associated to a jet. Charged particle objects, which we measure accurately in the silicon tracker, neutral particle objects, which are only measured in the calorimeters, and “secondary vertices”, which are points that are displaced from the main collision point and where two or more tracks seem to originate.

A part of the neural network architecture that is described in the following is inspired by the strategy that was previously applied, namely that features were engineered by hand for all physics object types. This handcrafted feature engineering is now replaced by convolutional network layers. We use 18 features for each charged particle, 8 for each neutral particle, and 16 for each secondary vertex. Multiple layers of 1D 1x1 convolutions are applied for each object type to extract features per object. The numbers of nodes of the chained 1D 1x1 convolutional layers on the object lists are 64, 32, 32, and 8 for charged particle objects, 32, 16, and 4 for neutral particle objects, and 64, 32, 32, and 8 for secondary vertex objects. After these convolutional layers, the objects are placed in sequences that are sorted based on physics arguments relevant to the classification. Different sorting schemes are found to have a very modest impact on the result. The sequences are then passed as inputs to recurrent neural networks (LSTM (11)) with an output of 150, 50, and 50 intermediate features for charged, neutral, and vertex objects, respectively. Each jet also has 8 global features. These global features are concatenated with the outputs of the recurrent networks and are then given as input to a dense neural network with 6 layers. We use ReLU (12) activation, DropOut (13) (rate 0.1) and batch normalization (14). For optimization, the Adam optimizer (15) is used with a learning rate of $\epsilon = 10^{-8}$ and no automatic learning rate decay. However, the learning rate is decreased manually during training time in case the loss plateaus. For the technical implementation, we use Keras (16) and Tensorflow (17). The training sample consists of about 100 million jets, of which 20 million are used as development and validation samples, evenly split. We use softmax as activation function in the last layer and categorical cross-entropy as loss function, with the following classes: b-jet, double b-jet, leptonic b-jet, c-jet, uds-jet and gluon-jet. b- and c-jets are also referred to as heavy flavor jets.

3 Performance comparison

The performance of our classifier for the sum of b-jet classes vs. the c-jet class and the combined uds- and gluon-jet classes is compared to the standard jet classification in use in CMS. Figure 2 shows the receiver operating characteristic curve. A significant gain of the DeepJet classifier (called DeepFlavour when used for heavy flavor jet classification) with respect to the standard classifier is observed. The standard classifier has a dense deep neural network that is similar to the one

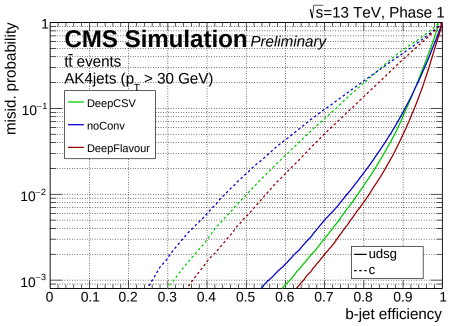

Figure 2: b-jet efficiency vs. misidentification probability for c-jets and uds- and gluon-jets of simulated events. A minimal transverse momentum (p_T) of 30 GeV is required. *DeepCSV* is the standard CMS classifier, DeepFlavour the neural network described in the text, and *noConv* a dense neural network with the same input as DeepFlavour (18).

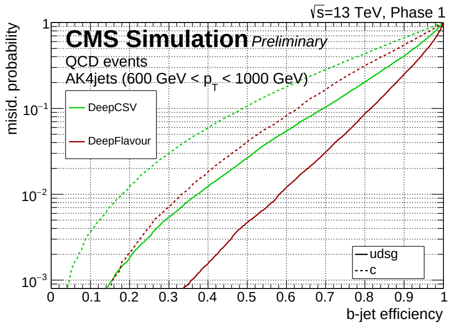

Figure 3: b-jet efficiency vs. misidentification probability for c-jets and uds- and gluon-jets of simulated events. A transverse momentum (p_T) range from 600 GeV to 1000 GeV is selected. *DeepCSV* is the standard CMS classifier, DeepFlavour the neural network described in the text (18).

in DeepFlavour, but uses handcrafted object features. To gain insight into the improvement, we also trained a wide dense neural network with the complete input of DeepFlavour, but without the convolutional layers; this is found to lead to a clear performance degradation. Figure 3 compares the performance of the two classifiers for very high momentum jets. Here DeepFlavour outperforms the old classifier by a large margin. A standard working point for jet classification is defined at 1% false positive rate; DeepFlavour leads to only 0.12% false positive rate for the same rate of true positives. A large fraction of this gain is due to the preselection of charged particle objects and object features used in the standard classifier; passing to DeepFlavour only the standard classifier inputs results in a performance similar to the latter.
For gluon- vs. uds-jet separation it was previously shown that deep neural networks outperform classical procedures (4). We directly compare DeepJet to a 2D convolutional network architecture similar to the one proposed in (4) and to a narrowed down version of DeepJet, where only 4 features per particle object and no convolutional layers are used; this is labeled as *recurrent* in the Figures. The sequence is sorted in descending order in the transverse momentum of the particle objects. Different methods of sorting have been tested in (7), which is similar to the narrowed down DeepJet neural network, with similar performance. The three architectures are compared in Figure 4. DeepJet and the custom classifiers give similar results. The image approach is slightly less performant, which might be due to information loss when transforming the continuous position information into an image with discrete position information. In real data, jet classification is always a multiclass classification problem, as a priori all classes are possible for a given jet. The DeepJet multiclass approach gives much more information to the user and performs as well as or better than all binary classification schemes.

4 Adaptation for wide jets

So far we have discussed the so-called "narrow" jets in which the constituent particles are contained in a cone of a relatively small radius. These narrow jets are typically used to identify particle sprays initiated by a single elementary particle, a quark or a gluon. However, when the quarks originate in the decay of a high-momentum heavy particle, the subsequent particle sprays from different quarks become highly collimated and cannot be resolved into separate narrow jets. In such cases, a "wide" jet, with a larger cone radius, is often used to include all particle sprays from the heavy particle decay in a single jet.

An approach similar to the one used for narrow jets can also be applied for wide-jet classification, where the task is to identify a wide jet as originating from a top quark, a W boson, a Z boson, a Higgs boson, or background processes. Due to different characteristics between narrow and fat jets, a few modifications are made on the neural network architecture to better adapt to the problem. Due to its

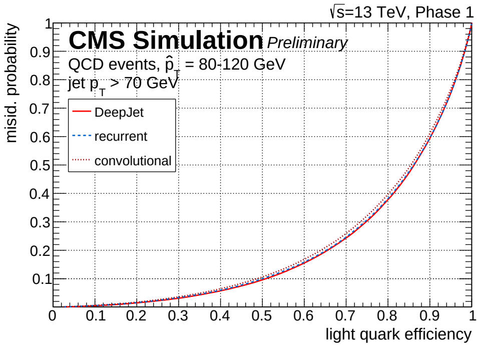

Figure 4: uds-jet (light quark) efficiency vs. gluon-jet misidentification probability of three different neural networks (DeepJet, *recurrent* and *convolutional*) described in the text (19).

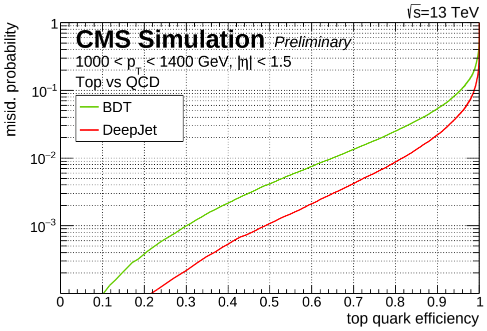

Figure 5: Top jet identification efficiency vs. background misidentification probability of DeepJet for wide jets and a BDT based algorithm (23).

larger cone size, a typical wide jet has at least twice as many particles as a typical narrow jet. The recurrent units used in narrow-jet classification become significantly slower to train, as the length of the input list grows. Moreover, the relations among the particle objects in a wide jet are more involved than the ones in a narrow jet, because a wide jet tends to contain more than just one quark. Thus, an architecture based solely on 1D convolutions is adopted to mitigate the high computational cost of recurrent units. The kernel size of the 1D convolutions is changed from 1 to 3 and the depth of the convolutional layers is increased in order to better exploit the rich interrelationship among particle objects in a wide jet.

The convolution-only architecture for wide jet classification is largely based on the ResNet model (20). We adapt it to work with 1D particle lists instead of 2D images, but keep the main structure and all important ingredients such as residual connection (21), batch normalization (14), and ReLU (12) activation function. The depth of the convolutional network is 14 layers for inclusive and charged particle objects, and 10 for secondary vertex objects. The inclusive particle object list is a combination of charged and neutral particle objects with their common features. This combination is found to improve the performance slightly. The filter size of each convolutional layer ranges between 32 to 128. The outputs from the three separate convolutional neural networks are combined in a fully connected layer with 512 units, followed by a DropOut layer with a drop-rate of 0.2, and then connected to the output layer. The neural network is implemented using the MXNet package (22) and trained with the Adam (15) optimizer with a learning rate of 0.001 to minimize the cross-entropy loss. The learning rate is reduced by a factor of 10 at the 10th, 20th, and 30th epochs to improve convergence. 40 million samples are used for training and 5 million for each, development and validation.

With the significantly improved performance observed for DeepJet in high-momentum b-jet identification, as shown in Figure 2, it is expected that top-jet identification will benefit from a similar approach because a b quark is present in top quark decays. Figure 5 shows the performance of DeepJet for the identification of wide jets originating from top quark decays. We compare the performance of DeepJet with one of the most performing top jet identification algorithms (24) used in CMS, which is based on boosted decision trees (BDT) and physics inspired input features. A significant improvement in performance is observed for DeepJet, where the false positive rate is reduced by a factor of 4 compared to the BDT-based algorithm at the same top-jet true positive rate of 50%.

5 Conclusion

We present new custom neural network architectures for generic classification of jets of various cone sizes and classes. Our proposals outperform significantly the standard classification procedures in the CMS experiment in all classes tested. If this performance gain in the simulation is confirmed using real collision data, the new classification will result in a significant improvement in the CMS detector capabilities and will therefore have a significant positive impact on the research output of the CMS experiment.

References

[1] L. Evans and P. Bryant, “LHC Machine,” JINST **3**, S08001 (2008)

[2] J. Cogan, M. Kagan, E. Strauss and A. Schwarztman, “Jet-Images: Computer Vision Inspired Techniques for Jet Tagging,” JHEP **1502**, 118 (2015)

[3] G. Kasieczka, T. Plehn, M. Russell and T. Schell, “Deep-learning Top Taggers or The End of QCD?,” JHEP **1705**, 006 (2017)

[4] P. T. Komiske, E. M. Metodiev and M. D. Schwartz, “Deep learning in color: towards automated quark/gluon jet discrimination,” JHEP **1701**, 110 (2017)

[5] P. Baldi, K. Bauer, C. Eng, P. Sadowski and D. Whiteson, “Jet Substructure Classification in High-Energy Physics with Deep Neural Networks,” Phys. Rev. D **93**, no. 9, 094034 (2016)

[6] D. Guest, J. Collado, P. Baldi, S. C. Hsu, G. Urban and D. Whiteson, “Jet Flavor Classification in High-Energy Physics with Deep Neural Networks,” Phys. Rev. D **94**, no. 11, 112002 (2016)

[7] G. Louppe, K. Cho, C. Becot and K. Cranmer, “QCD-Aware Recursive Neural Networks for Jet Physics,” arXiv:1702.00748 [hep-ph] (2017)

[8] ATLAS Collaboration, “Identification of Jets Containing b b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment,” ATLAS note: ATL-PHYS-PUB-2017-003, http://cds.cern.ch/record/2255226, (2017)

[9] S. Chatrchyan *et al.* [CMS Collaboration], “The CMS Experiment at the CERN LHC,” JINST **3**, S08004 (2008)

[10] A. M. Sirunyan *et al.* [CMS Collaboration], “Particle-flow reconstruction and global event description with the CMS detector,” JINST **12**, no. 10, P10003 (2017)

[11] S. Hochreiter and J. Schmidhuber, “Flat Minima,” Neural Computation 9(1):1-42, `http://people.idsia.ch/~juergen/fm/` (1997)

[12] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” Proceedings of the 27th international conference on machine learning, pp. 807–814. (2010)

[13] N. Srivastava *et al.*, “Dropout: a simple way to prevent neural networks from overfitting,” Journal of machine learning research **15**, no. 1, 1929–1958. (2014)

[14] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” Proceedings of the 32nd International Conference on Machine Learning, pp. 448–456. (2015)

[15] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv: 1412.6980 [cs.LG] (2014)

[16] Chollet, François et al., “Keras,” `https://github.com/fchollet/keras`. (2015)

[17] Martín Abadi et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” `https://www.tensorflow.org` (2015)

[18] CMS Collaboration, “CMS Phase 1 heavy flavour identification performance and developments,” Detector Performance Figures: CMS-DP-17-013, `http://cds.cern.ch/record/2263802`, (2017)

[19] CMS Collaboration, “New Developments for jet substructure reconstruction in CMS,” Detector Performance Figures: CMS-DP-17-027, `https://cds.cern.ch/record/2275226`, (2017)

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” European Conference on Computer Vision, pp. 630–645, Springer. (2016)

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778. (2016)

[22] T. Chen et al., “Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems,” arXiv:1512.01274 [cs.DC] (2015)

[23] CMS Collaboration, “Boosted jet identification using particle candidates and deep neural networks,” Detector Performance Figures: CMS-DP-17-049, (2017)

[24] A. M. Sirunyan *et al.* [CMS Collaboration], “Search for direct production of supersymmetric partners of the top quark in the all-jets final state in proton-proton collisions at $\sqrt{s} = 13$ TeV,” JHEP **1710**, 005 (2017)