Karlsruhe Institute of Technology

FAKULTÄT FÜR PHYSIK
Institut für Experimentelle Kernphysik

Eventklassifikation am Castor-Kalorimeter mithilfe von CNNs (Classification of Events with CNNs at the Castor Calorimeter)

Bachelorarbeit
von

Isabel Haide

am Institut für Experimentelle Kernphysik

Reviewer: Prof. Dr. F. Bernlochner
Second Reviewer: Dr. R. Ulrich

Bearbeitungszeit: 15.03.2018 – 01.07.2018

Erklärung zur Selbstständigkeit

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der gültigen Fassung vom 17.05.2010 beachtet habe.

Karlsruhe, den 01.07.2018, ____________________________

Isabel Haide

Als Ansichtsexemplar genehmigt von

Karlsruhe, den 01.07.2018, ____________________________

Prof. Dr. F. Bernlochner

Contents

List of Figures

List of Tables

1. The detector

2. Neural networks

3. Data analysis

If the energy deposit of every section of every tower is shown in a 2D histogram, it can be treated a picture with 16x14 pixels. The information of how high the energy deposit was is transmitted through the intensity of the black colour filling. With a convolutional neural network it should be possible to learn the distinctive patterns of different kind of particles and therefore to classify events correctly.

3.1 Data

To train the network a sufficiently high amount of data is needed. For training and validation Monte Carlo events generated by the PYTHIA8 and the EPOS generator are used. To separate actual events from background noise the CaloTowers are used which combine information of the electromagnetic and hadronic calorimeter of CMS. Only events which trigger an energetic response higher than 5 GeV in the CaloTowers with a pseudorapidity range of $3.1 < |\eta| < 5$ are used in the analysis. To simplify the work for the neural network, only events containing isolated particles are used for differentiation. Here an isolated particle is defined by a minimum distance of $\Delta\mathrm{R} = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2} = 0.8$ to the next particles generated by the same event. Therefore every generated event is checked for isolated particles travelling through CASTOR.

Within CASTOR different kinds of particles leave different kinds of tracks which can be used to identify the particles. In fig. 3.1 the different kinds of signals can be seen. Electromagnetic light particles, such as electrons, positrons and photons, have a very large cross section and interact with the electromagnetic part of the calorimeter. In the energy deposit histogram very high energies are measured within the first two sections while nearly no energy is deposited in the hadronic sections. When hadrons reach the calorimeter they interact and produce several lighter secondary particles which in turn interact again. The energy deposit is therefore distributed wider and reaches its peak within the hadronic sections of the calorimeter. Muons very sparsely interact with the material of the calorimeter and pass it nearly without losing energy. A muon track is normally contained within one tower but can be seen as a very small energy deposit in every section.

Every histogram containing the energy deposit has a corresponding label vector. This is a binary vector, where the first index corresponds to isolated electron, positron or photon, the second to isolated hadron and the third to background. Here background is everything besides an isolated electron/photon or hadron. Therefore the categories do not exclude each other, since it is possible that an isolated electron and an isolated

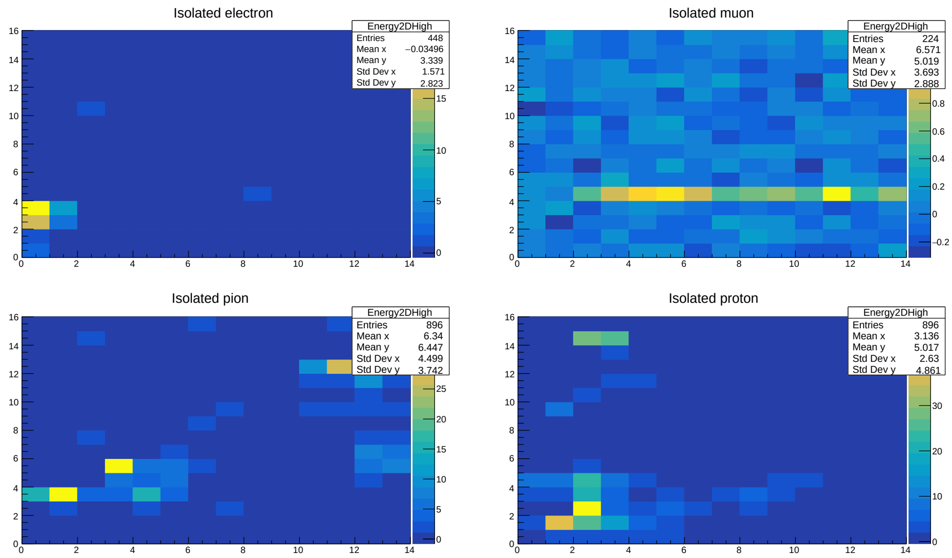

Figure 3.1: Energy deposit of single particles travelling through CASTOR. An electron or photon (upper left) deposits its whole energy in the electromagnetic sections, while a muon travels with only a little energy loss through every section (upper right). The signals of pions and protons is spread wider and reaches into the hadronic sections of the calorimeter (lower left and right).

hadron travel through CASTOR during the same event. In table 3.1 the particles and their frequency are listed. 500.000 Monte Carlo events were evaluated and only those particles with the beforementioned minimum distance were counted. For a neural network at least 1000 samples per class are needed to pick up the identifying features. As can be seen most particles are too few to be correctly identified, whereas the differentiation between electrons/photons (which produce nearly the same signal) and hadrons is possible. Even though muons have an especially distinctive signal, their number is too small to include them as a separate category.

As seen in fig. 3.1, the energy deposited in CASTOR by different particles varies much. In different events particles with very different energies are created. For example the kinetic energy of a proton can be between one and several thousand GeV. To learn that these very different signals are generated by the same particle would pose a challenge for the neural network. Every histogram is therefore normalized to 1.

3.2 Network design

The energy deposit histograms of CASTOR can be treated as pictures with 16 to 14 pixels and varying amounts of black colour. To design a neural network which can classify events into particle categories techniques in image recognition should be used. In normal classification problems convolutional layers with a small and quadratic kernel size are used. Here the information given by the underlying physics can be used to the advantage of the network.

As particles reach the calorimeter from the left, it is not useful to use small kernel sizes to cover the whole image. Small, light particles are normally contained in one

Table 3.1: Number of isolated particles travelling through CASTOR in 500.000 Monte Carlo generated events. If not specified otherwise, the numbers include the particle and the antiparticle. As can be seen, hadrons and electrons/photons dominate the spectrum.

Leptons/Bosons	Count	Hadrons	Count
Photon	170047	π^+	106904
Positron	243	π^-	103648
Elektron	250	K^+/K^-	27591
Muon	29	Proton	15782
ν_μ	27	Neutron	15232
ν_e	11	K_L^0	13604
ν_τ	1	Λ	5308
		Σ^+	2016
		Σ	1993
		Σ^-	1993
		K_s^0	1323
		Ξ^0	511
		Ξ^-	471
		Ω^-	16

tower or at most two. Therefore convolutional layers with kernel sizes of one or two towers, meaning one or two pixels in the height and 14 in the width, should be able to recognize the important features of the tracks. Several layers, each with kernel sizes with a width of 14 pixels but varying heights are concatenated in the beginning of the network. In the deeper layers of the network, smaller kernel sizes can be used, since the convolution shrinks the size of the signal travelling through the neurons.

3.3 Monitoring

For monitoring the progress of a neural network, the first important feature is the loss function. As the classification problem is a multi-label classification, the loss function used here is the binary crossentropy loss. Every epoch it is calculated for the training and for the validation data set. Both variables need to decrease during the training. To determine if the input is correctly put into the network the first step was to train without any dropout or pooling layers. The network should overfit on the training set if given enough time. In fig. the loss and the accuracy of the training while overfitting can be seen.

As standard networks normally deal with multi-class and not multi-label problems, it proved to be useful to additionally monitor variables commonly used in classification problems. Here recall and precision were implemented.

With recall, the false negatives are taken into account, while precision monitors the false positives. Recall is also called the sensitivity of a network and is calculated by

$$\text{Recall} = \frac{tp}{tp + fn} \quad . \tag{3.1}$$

Here tp stands for true positive, tn for false negative. The recall therefore is the fraction of true positive labels over all actually positive labels. In the case of all labels being correctly predicted, the recall is 1.

Precision is also referred to as the positive predictive value. It is the rate of all correctly classified positive labels over all positively predicted labels.

$$\text{Precision} = \frac{tp}{tp + fp} \tag{3.2}$$

The nomenclature is the same, tp for true positives and fp for false positives. When all labels are correctly predicted, the precision also equals 1.

The precision and the recall can be calculated separately for the different categories. It can be easily monitored if the networks has more problems recognizing one category over the others.

Another number which was monitored was the number of falsely labelled electrons. It was obvious after a few training sessions that the network had most problems classifying electrons. A counter of the falsely labelled electrons was therefore implemented in the callbacks. With $\mathrm{y_{true}}$ as the true labels and $\mathrm{y_{pred}}$ as the predicted labels, the counter worked as seen below.

$$\text{count} = (1 - y_{\text{true}}) \cdot y_{\text{pred}} - (1 - y_{\text{pred}}) \cdot y_{\text{true}} \tag{3.3}$$

It is the same as the crossentropy loss without the logarithm, but with two main differences. The first one is that here not the predicted probalities are used as input put the absolute prediction. That means if $\mathrm{y_{pred}}$ is higher than 0.5 it is rounded to 1, lower than 0.5 means being rounded to 0. The second one is the sign between the two parts. With the minus sign it can be directly evaluated if the network either classifies more or less input as being part of the electron category than there actually is.

3.4 Evaluation

In the process of training it became apparent that no real physical features were being learned by the network. The training loss decreased but the validation loss more or less stayed the same over several hundred epochs (fig.). As mentioned before, without regulations the network overfitted on the training data without problem. This lead to reevaluating the input data, as it seemed to be difficult for the network to generalize the underlying physical laws.

One problem was the definition of background events. Since at first only isolated particles were considered background events could also contain hadrons or electrons and photons, just not in the right distance to other particles travelling through CASTOR. The background category therefore contained input data which were not distinguishable from an photon or hadron event. To solve this problem, two different approaches were chosen. The first one was to completely ignore the background category and turn the classification into a binary problem, only distinguishing between electromagnetic particles, photons and electrons, and hadrons. The second was the redefinition of the background category, where only true background was used. True background meant, that no particle with an energy higher than 5 GeV had travelled through CASTOR in the corresponding signal.

As precision and recall were implemented as callbacks, their value was returned after every epoch. In the first few training sessions the monitoring of these showed that while the mean loss decreased and the accuracy increased, the recall especially of electrons remained really low. To counteract this, the binary crossentropy loss was defined as a separate function. The crossentropy is the sum of two parts, one evaluating and penalizing the number of false positives, the other watching the number of false negatives. As recall, as seen in eq. 3.1, evaluates the false negatives, they were stronger penalized than the false positives.

Bibliography

[1] "Desy previous activities: Physics in the very forward region." http://cms.desy.de/sites/site_cms/content/e53612/e90916/e277689/e115497/e261362/e261366/cms_with_castor_drawing.png. Accessed: 2018-06-05.

[2] P. Göttlicher, "Design and test beam studies for the castor calorimeter of the cms experiment," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 623, no. 1, pp. 225 – 227, 2010. 1st International Conference on Technology and Instrumentation in Particle Physics.

[3] "Cs231n: Convolutional neural networks for visual recognition." http://cs231n.github.io/assets/nn3/nesterov.jpeg. Accessed: 2018-06-27.

[4] "The cms experiment at cern." https://cms.cern/detector. Accessed: 2018-06-03.

[5] "Castor detector." http://cmsdoc.cern.ch/cms/castor/html/. Accessed: 2018-06-07.

[6] "Cs231n: Convolutional neural networks for visual recognition." http://cs231n.github.io/assets/nn3/accuracies.jpeg. Accessed: 2018-06-27.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[8] F. Chollet *et al.*, "Keras." https://keras.io, 2015.