Machine Learning and Physics
Winter Semester 2023/2024

Exercise Sheet #6

Due: 04.12.2023 11h00

Jan Pawlowski, Tilman Plehn
Jonas Spinner

General Regulations.

- Please hand in your solutions in groups of three people. A mix of attendees from different tutorials is fine. We will not correct submissions from single students.
- Your solutions to theoretical exercises can be either handwritten notes (scanned to `pdf`), typeset using LaTeX, or directly in the jupyter notebook using Markdown.
- For the practical exercises, the data and a skeleton for your jupyter notebook are available at https://github.com/heidelberg-hepml/mlph2023-Exercises. Always provide the (commented) python code as well as the output, and don't forget to explain/interpret the latter, we do not give points for code that does not run. Please hand in both the notebook (`.ipynb`) and an exported pdf. Combine your the pdfs from theoretical and notebook exercises into a single pdf.
- Submit all your files in the Übungsgruppenverwaltung, only once for your group of three. Please list all names and tutorial numbers to simplify things for us.

1 Top Tagging with CNNs

Whenever colour-charged particles like quarks and gluons are produced at the Large Hadron Collider (LHC), they shower and hadronize to form so-called jets, or in other words collimated sprays of particles. The internal structure of jets contains important information about the original particle forming the jet. For example, top quark jets typically have a 3-prong structure, originating from the decay $t \to bW^+ \to bq\bar{q}'$ happening early in the evolution of the jet. In contrast, so-called QCD jets originating from light particles like gluons or the light u, d, s quarks usually do not have such a structure. Classifying these two types of jets was the first mainstream application of machine learning in particle physics, and improving as well as extending this classification is an active field of research.

(a) The original dataset, available on https://www.thphys.uni-heidelberg.de/~plehn/pics/toptagging-short.zip, is provided as a list of 4-momenta for each particle in the jet, with zero-padding for non-existent jet constituents. Use the provided functions to transform this point cloud into an image as one would see it in a particle detector. Visualize examples of a top jet and a QCD jet, as well as the *mean* top and QCD jet that you obtain by averaging over all events in the training dataset. (2 pts)

(b) Implement your own version of the binary cross-entropy loss of sheet 5, exercise 1, assuming logits $D(x) \in [0, 1]$ as inputs. Check your implementation by comparing with `torch.nn.BCELoss`. (1 pts)

(c) Implement a CNN with two convolutional layers, max-pooling and one linear layer in the end. Train your CNN on the training dataset while tracking the validation loss. Choose appropriate values for the hyperparameters. (2 pts)

(d) Evaluate the trained CNN on the test dataset. Compare the accuracy on training and test dataset. Use the sklearn library to compute true positive rates (TPR) and false positive rates (FPR) of the classifier test statistic. Use them to plot the ROC curve (TPR over FPR), the SIC curve ($\mathrm{TPR}/\sqrt{\mathrm{FPR}}$ over TPR) as well as the AUC value. Discuss the performance of your classifier based on these three characteristics. Hint: `sklearn.metrics.roc_curve`, `sklearn.metrics.roc_auc_score` (2 pts)

(e) What do you think about training a fully connected network on this task? Discuss aspects like the number of learnable parameters and weight sharing. (1 pts)

(f) Instead of transforming the point cloud data into images as we did in part (a), one can design architectures that work directly on point clouds. Discuss pros and cons of the point cloud representation compared to the image representation of jets. Explain your favourite point cloud architecture. Hint: Lecture (2 pts)

2 Implementing Transformers with a Transformer

Using ChatGPT[1] only, try and implement a multi-head attention layer. Demonstrate that the code runs (if you ask it nicely, ChatGPT might also help you write some tests). (2 pts)

[1] https://openai.com/blog/chatgpt/. Sometimes the servers are overloaded, but it seems like you have a good chance to reach it in the early morning.