Machine Learning and Physics
Winter Semester 2025/26

Exercise Sheet 7

Due: 08.12.25 14h00

Christof Gehrig, Gerrit Gerhartz

General Regulations.

- Please hand in your solutions in groups of two (preferably from the same tutorial group).
- Your solutions to theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX. For scanned handwritten notes, please ensure they are legible and not blurry.
- For the practical exercises, always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter.
- Please hand in a **single PDF** that includes both the exported notebook and your solutions to the theoretical exercises. Submit the PDF to the Übungsgruppenverwaltung once per group, making sure to include the names of both group members in the submission.
- You can find all the data in the GitHub Repository.

1 CNNs for Galaxy Classification

The Galaxy10 SDSS[1] dataset consists of 21785 colored (green, red and near-infrared band) images of galaxies with a resolution of 69 by 69 pixels, taken from the Sloan Digital Sky Survey (SDSS). They have been annotated by human volunteers: Each image is assigned to one of nine classes which describe the morphology of the depicted galaxy. In this task, you will train Convolutional Neural Networks (CNNs) to classify the images into those classes.

(a) Load the data and visualize 3 instance of each class in a figures with 3×10 subplots. Split the data into train, validation and test set and convert each split to a `torch.utils.data.TensorDataset`. The validation and test set should each contain 10% of the data points. Then, normalize the images using `torchvision.transforms import Normalize`. Which mean and standard deviation do you have to compute? (2 pts)

(b) Implement a small CNN for the classification task: It should consist of three consecutive blocks:

(i) A convolutional layer with kernel size 5 and 8 output features, ReLU activation and 2×2-max-pooling,

(ii) A convolutional layer with kernel size 5 and 16 output features, ReLU activation and 2×2-max-pooling,

(iii) A flattening layer that reshapes the channel, width and height axes into a single axis, followed by an MLP with two hidden layers of size 64 and 32. (2 pts)

(c) Instantiate the optimizer and criterion: Use Adam with the default learning rate of 10^{-3}. Which loss function do you suggest for classification? (1 pt)

(d) Implement the training loop and train the neural network, for at least 30 epochs. Plot the training loss over the training iterations. After each epoch, compute the training accuracy of that epoch and evaluate the model on the validation set to compute the validation loss and accuracy. Plot both accuracies versus the training iterations and discuss the results. Why does the validation loss increase after some epochs while the validation accuracy still improves / stays constant? (3 pts)

(e) After training, create and plot the confusion matrix of the model on the train and test set. (1 pt)

(f) Plot the learned convolutional kernels of the first CNN layer. Can you interpret them? (2 pts)

[1] https://astronn.readthedocs.io/en/latest/galaxy10sdss.html

(g) **Bonus:** In contrast to natural images, which often have a preferred orientation, the galaxies in this dataset are oriented arbitrarily. Hence, it would be desirable to adapt the model such that its prediction is invariant with respect to at least some rotations, e.g. that rotating the input image by a multiple of 90° does not change the outputs. How could this be achieved? Can you also come up with a method that would work for a larger set of rotations? (2 bonus pts)

2 Receptive Field of VGG16

(a) Using pen and paper, compute the field of view of the famous VGG16 convolutional network (https://arxiv.org/abs/1409.1556) for perceptrons just before the fully connected layers. That is, compute the set of input pixels that the prediction in a certain output pixel depend upon. All filters have size 3×3 and the max pooling is over 2×2 pixels. (2 pts)

(b) Compute the number of parameters in the convolutional layers (A) and (B). Compare your results to the number of parameters in the fully connected layer (C). (1 pt)

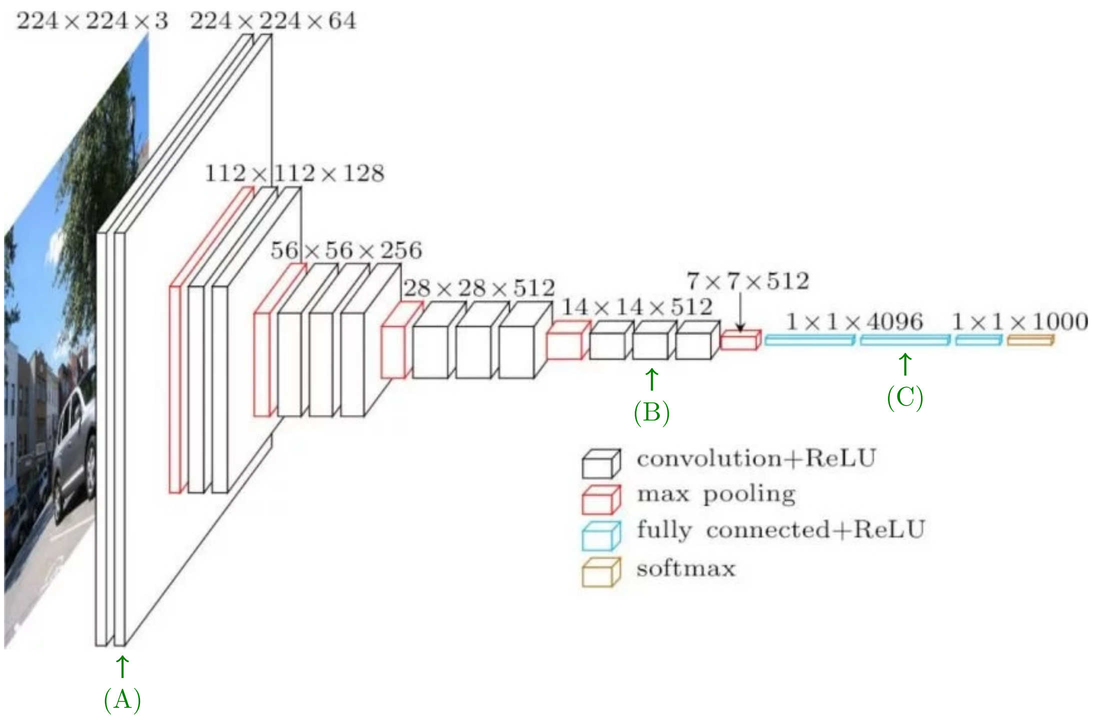

Figure 1: Schematic VGG16 architecture, taken from https://neurohive.io/en/popular-networks/vgg16/.

3 Autoencoders and PCA

Assume you have data with $p > 2$ dimensions and that you would like to find a two-dimensional approximation of your data. You train the following simple autoencoder: The encoder is a single linear layer with two neurons; the decoder is a single linear layer with p neurons.

(a) Sketch the network architecture for $p = 4$, and also describe the architecture in terms of a single linear algebra formula. (2 pts)

(b) Where in the network can you read off the directions of the two axes spanning the two-dimensional plane on which the input data is projected? (1 pt)

(c) Standard autoencoders use squared reconstruction error as loss function. Give its formula. (1 pt)

(d) **Bonus:** Assume the above architecture has been successfully trained with squared reconstruction loss. Are the axes spanning the projection plane necessarily identical to those which principal component analysis (PCA) would find? (2 bonus pts)

(e) **Bonus:** Now assume your data has outliers and you want to robustify the search for a good subspace. What loss function could you use? (2 bonus pts)

Note: This exercise is taken verbatim from last year's exam (except that none of the tasks were marked as "bonus").

4 Understanding decoders

Assume you have a deep MLP autoencoder which was trained on observations $\mathbf{x}_i \in \mathbb{R}^p$ using squared reconstruction loss. You now fix the parameters in the encoder E and train the decoder D from scratch. Assume that a nonempty subset $\mathcal{S}$ of your training points are mapped to the identical position in latent space, i.e. $\mathbf{z} = E(\mathbf{x}_i)$, $\forall i \in \mathcal{S}$. Conversely, assume that no other training data are mapped to that particular location. Given a successfully trained decoder that minimizes the squared reconstruction error, derive the value of $D(E(\mathbf{x}_i))$ for $i \in \mathcal{S}$. What is its geometric interpretation? (2 pts)

Note: This exercise is taken verbatim from last year's exam.