Machine Learning and Physics
Winter Semester 2024/2025

Exercise Sheet #8

Due: 16.12.24 14h00

Peter Lippmann

General Regulations.

- Please hand in your solutions in groups of two (preferably from the same tutorial group). Submissions by a single person alone will not be corrected.
- Your solutions to theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX. For scanned handwritten notes please make sure that they are legible and not too blurry.
- For the practical exercises, the data and a skeleton for your jupyter notebook are available at `https://github.com/sciai-lab/mlph_w24`. Always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter. Please hand in your notebook (`.ipynb`), as well as an exported pdf-version of it.
- Submit all your files in the Übungsgruppenverwaltung, only once for your group of two. Specify all names of your group in the submission.

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? (4 pts)

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

1. A convolutional layer with kernel size 5 and 8 output features, ReLU activation and 2×2-max-pooling,
2. A convolutional layer with kernel size 5 and 16 output features, ReLU activation and 2×2-max-pooling,
3. 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? (2 pts)

(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. (4 pts)

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

(e) Add the functionality to your training loop to save the model after each epoch, if the validation loss is the lowest encountered so far. After training, load this model and evaluate the performance on the test set. Create and plot a confusion matrix of the trained model on the test set. (2 pts)

(f) 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 pts)

2 Contrastive learning as an example of self-supervised learning

In self-supervised learning, the model learns to solve a task the ground truth for which can be easily created from the raw data itself. The motivation is the hope that the model will learn an informative representation of the data in the process; a representation which might help it to solve the actual tasks of interest later on.

In this exercise, we consider contrastive learning. The idea of **SimCLR** ("A Simple Framework for Contrastive Learning of Visual Representations", https://arxiv.org/pdf/2002.05709) is to use a family of transformations $\mathcal{T}$ to generate multiple views of the original observations. The model the authors propose is

$$x_i \xrightarrow{t\sim\mathcal{T}} x_{it} \xrightarrow{f(\cdot;\theta)} h_{it} \xrightarrow{g(\cdot;\phi)} z_{it} \tag{1}$$

Here, $x_i, i = 1, \ldots, n$ is an original observation; t is a transformation sampled from family $\mathcal{T}$; f is the encoding network with parameters θ; and g is a projection network with parameters ϕ.

The model then learns to create an embedding such that different views of the same sample are close-by in representation space, as measured by cosine similarity[2]; and that embeddings of views of different objects are less similar. The authors propose to train the model such that the similarity of pairs of transformed observations $(x_{it}, x_{jt'})$ match corresponding ground truth labels $y_{itjt'} = \delta_{ij}$ (Kronecker delta). After training, the network g is discarded and f is used as preprocessing for any downstream tasks.

For a given data set $\{x_i\}$, family of transformations $\mathcal{T}$ and network architectures f and g, the parameters θ and ϕ are found by minimizing the InfoNCE loss. This loss measures the cross-entropy between the ground truth labels $y_{itjt'}$ and the current predictions soft(arg)max$(sim(z_{it}, z_{jt'})/\tau)$ where τ is a user-adjustable pseudo temperature.

(a) Spell out the soft(arg)max in an explicit formula. (1 pt)

(b) Spell out the InfoNCE loss in our notation. (1 pt)

(c) Simplify this loss, remembering that $0 \cdot \log 0 = 0$. (1 pt)

(d) Anaylize the resulting loss function: How do its ingredients relate to classification? A typical problem in representation learning is mode collapes (all inputs are mapped to the same embedding). Why does this loss yield useful, non-trivial embeddings? Why are the h used as representation for downstream tasks rather than the z? (3 pts)

Hint: in this application of InfoNCE, it is the pairs *of observations that take center stage; each pair of observations can be considered one sample.*

[2]The cosine similarity is defined as the cosine of the angle between two embeddings $sim = \frac{z_{it} \cdot z_{jt'}}{\|z_{it}\|_2 \|z_{jt'}\|_2}$.

3 Positional Encoding

In transformers, the features $X \in \mathbb{R}^{p\times n}$ are combined with the positional embeddings E of the tokens. We will consider the simplest case of self-attention where the attention weights are computed as the soft(arg)max over the "scores" K^TQ with (i) $K = Q = X + E$ in case features and positional embeddings are combined by adding them up and (ii) $K = Q = \mathrm{cat}(X, E)$ in case they are concatenated. Let the positional embeddings $E \in \mathbb{R}^{p\times n}$ be defined as

$$E_{(2k),i} = \sin\left(i \cdot \exp\left(-\frac{2k \cdot \log(10000)}{p}\right)\right) \tag{2}$$

$$E_{(2k+1),i} = \cos\left(i \cdot \exp\left(-\frac{2k \cdot \log(10000)}{p}\right)\right), \tag{3}$$

for $k \in \{0, 1, 2, \ldots, \frac{p}{2} - 1\}$, taken from "Attention Is All You Need". The index i runs over the samples (which are assumed to be ordered). log is the natural logarithm.

(a) Expand the scores in X and E both for addition and concatenation. Discuss the different resulting terms. (2 pts)

(b) Implement the positional encoding as defined above and plot E^TE for 64 tokens with an embedding dimension of 256. What do you observe? (2 pts)

(c) Plot K^TQ for some random features (with the same variance as the positional embedding) for both addition and concatenation and discuss what you see. (2 pts)

(d) Bonus: Assume that both the positional encodings and the features of all tokens lie in two (comparatively) low-dimensional subspace of the high-dimensional euclidean space. Argue why under these assumptions, adding the positional encodings will lead to similar results as concatenating them with the features. (2 pts)