Machine Learning and Physics
Winter Semester 2024/2025

Exercise Sheet #6

Due: 02.12.24 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 Autoencoders: theory and practice

In this exercise, we study autoencoders. Consider a toy dataset of $n = 100$ points sampled uniformly from the rectangle $[-2, 2] \times [-1, 1]$. Assume you work with three autoencoders (AEs) with a bottleneck dimension of one, with the following numbers of neurons: $\{20, 10, 1, 10, 20, 2\}$, $\{50, 50, 1, 50, 50, 2\}$, as well as an architecture implementing PCA (up to a linear transformation).

(a) Provide PyTorch code defining all three architectures. Pay particular attention to any non-linearities and explain your choices. (3 pts)

(b) Use the pytorch lightning framework to train your models (use the provided code blocks in the jupyter notebook). Plot the loss curves for each model. Encode the input data to obtain a latent embedding, then, for each model, visualize the embeddings as colors in the input data space (use a colorbar). Discuss your results. (3 pts)

(c) If we sample points from an interval in the latent space, the decoder can attempt to map them to a curve in the ambient/input space. Guess what these curves may look like: i) After random initialization of the MLP parameters, and ii) After training the respective architecture.
Answer for all three architectures. (3 pts)

(d) Verify your hypotheses experimentally and discuss your observations. (3 pts)

(e) Assume you have a dataset of n points in $\mathbb{R}^p$. Can an MLP autoencoder with a bottleneck dimension of 1 perfectly reconstruct all n points, given a suitable architecture and training? Argue. (2 pts)

(f) Assume you have a fully trained autoencoder (trained on a dataset of n points in $\mathbb{R}^p$). Discuss what exactly happens if you fix all parameters in the decoder but retrain the encoder from scratch. (2 pts)

(g) Verify your hypotheses using the large autoencoder architecture above. Create a meaningful visualization, e.g., using arrows. *Hint: Set the requires_grad attribute of all parameters that you do not want to update during training to False.* (3 pts)

(h) Study the learning curves for the above problem for gradient descent vs. stochastic gradient descent. For a fair comparison, plot the learning curves over the number of training examples the model has seen during training. (2 pts)

2 Bonus: Training of an MLP

In this exercise, we use http://playground.tensorflow.org to gain some intuition on what happens throughout the training of a neural network and how the different pieces interact and work together. For each part submit a screenshot and a short discussion of what you observed.

(a) Fitting a Neural Net. Consider the spiral data set and the first two features (X_1, X_2). Come up with an architecture that can learn to classify the pattern well. You are free to use any number layers/neurons/activation functions/regularization/... (2 pts)

(b) Exploring Regularization. Pick the largest network size (i.e. 6 layers of 8 neurons each) and one of the data sets. Train it first without regularization, observing the behavior. Retrain it with L1 and L2 regularization and observe how the weight structure changes. What kind of behavior do you expect and does it fit with what you observe? (2 pts)

(c) Breaking Things. As discussed in the lecture, a net with enough parameters can fit any kind of pattern, even if there is none. Try to replicate this observation. Have your net learn a pattern "perfectly" (i.e. very low training error), but without having predictive power (i.e. the test error stays larger than random (which would be 0.5)). *Hint: If you consider the spiral data set with the minimal amount of training data and the maximum amount of noise, you get a collection of points with most structure removed.* (2 pts)