Universität des Saarlandes
Prof. Dr. Dietrich Klakow
Lehrstuhl für Signalverarbeitung
NNTI Winter Term 2023/2024

Exercise Sheet 7

Neural Networks Basics

Deadline: 22.12.2023 23:59

Guidelines: You are expected to work in a group of 2-3 students. While submitting the assignments, please make sure to include the following information for all our teammates in your PDF/python script:

Name:

Student ID (matriculation number):

Email:

Your submissions should be zipped as **Name1_id1_Name2_id2_Name3_id3.zip** when you have multiple files. For assignments where you are submitting a single file, use the **same naming convention** without creating a zip. For any clarification, please reach out to us on the **CMS Forum**.
Note that the above instructions are mandatory. If you are not following them, tutors can decide not to correct your exercise.

Exercise 7.1 - Maximum Likelihood Estimation and Cross-Entropy (0.25 + 0.25 + 0.5 + 1 = 2 points)

Given a set of m i.i.d. samples $\mathcal{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$ drawn from a data-generating distribution $p_{data}(\mathbf{x})$ and a parametric family of probability distributions over the same space $p_{model}(\mathbf{x}; \theta)$.

a) Write down the maximum likelihood estimator for θ.

b) Explain the difference between empirical distribution $\hat{p}_{data}(\mathbf{x})$ and the data-generating distribution $p_{data}(\mathbf{x})$?

c) Rewrite the expression derived in *a)* as an expectation using the empirical distribution $\hat{p}_{data}(\mathbf{x})$. Give an argument for why this is possible.

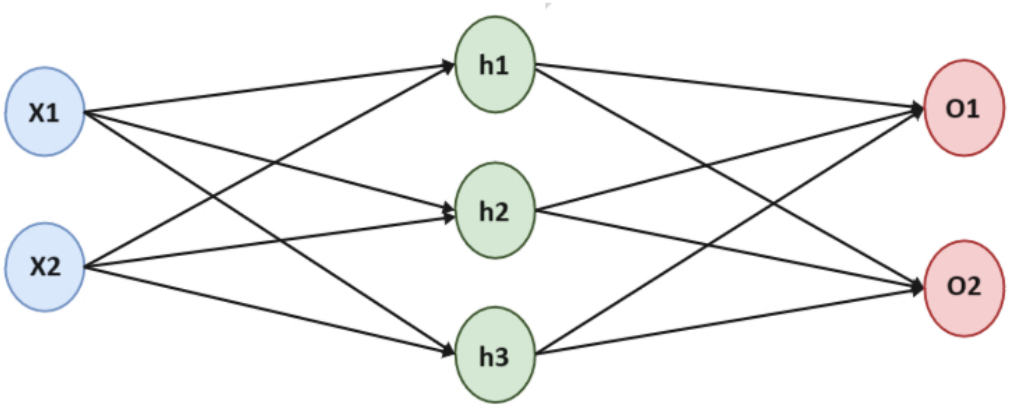
d) Show that minimizing the cross-entropy between $\hat{p}_{data}(\mathbf{x})$ and $p_{model}(\mathbf{x}; \theta)$ is exactly the same as computing the maximum likelihood estimator in *a)*.

Hint: *Note that the definition of KL-divergence is*:

$$D_{KL}(P||Q) = \mathbb{E}_{x \sim P}\ [log\ \frac{P(x)}{Q(x)}]$$

Hint: *For this question, it might be useful to refer to the introduction of the optimization chapter, Chapter 5.5 and chapter 5.10 in Ian Goodfellow's deep learning book, as well as the Loss Function Section in Chapter 5 from the NNTI Lecture*

Exercise 7.2 - Backpropagation (0.5 + 1.5 + 1 points)



We have a Feedforward Neural network with one input layer, one hidden layer and one output layer. The hidden layer uses the leaky ReLU with an $\alpha = 0.01$ as activation function and the output Layer uses the Softmax activation function. Also note that the network minimizes Cross-entropy loss which is given by,

$$L(\theta) = -\mathbb{E}_{x,y\sim\hat{p}_{\text{data}}} \log p_{\text{model}}(y|x;\theta)$$

We consider the true class labels to be binary, i.e. 0 or 1.
For the purpose of computing the derivatives of the loss/cost function consider the numerical values obtained by the network.

The input layer consists of the two nodes x_1 and x_2. For our problem consider the following input:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

with output of:

$$\begin{bmatrix} o_1 \\ o_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

The hidden layer is made up of 3 neurons. There is no bias. The corresponding matrix of weights is given as:

$$W_{hidden} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} = \begin{bmatrix} 0.15 & -0.25 & 0.05 \\ 0.2 & 0.1 & -0.15 \end{bmatrix}$$

The output layer consists of two neurons, i.e., the network generates two outputs. The weight matrix corresponding to the Output layer is given by:

$$W_{out} = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} \\ w_{31}^{(2)} & w_{32}^{(2)} \end{bmatrix} = \begin{bmatrix} 0.2 & 0.5 \\ -0.35 & 0.15 \\ 0.15 & -0.2 \end{bmatrix}$$

Show that Back-propagation reduces the Binary Cross Entropy loss by performing the following steps:

- Perform a Forward-propagation with the given input x and compute the loss $L^{(1)}$.
- Use the chain rule and write down the expressions used for back propagation.
- Compute the Back-propagation and apply Gradient descent with a learning rate of 0.1 to update the weights.
- Perform Forward-propagation again with the updated weights and recompute the loss $L^{(2)}$. Briefly explain your findings.

As always: To get full points, you need to give and explain your intermediate steps explicitly! Because this exercise is quite large, make sure to hand in a structured and understandable solution.

Exercise 7.3 - Backpropagation (5 points)

See the accompanying Jupyter Notebook.