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

# Exercise Sheet 10

Initialization and Convolution

**Deadline: 25.01.2024 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 10.1 - Initialization (0.5 + 0.25 points)

a) Why is it a bad idea to initialize all weights to zero when using ReLU activation functions? (max. 2 sentences)

b) We often initialize biases to zero. Name an example where it is better to initialize them to a different value. (1 sentence)

## Exercise 10.2 - Convolutional Neural Networks (0.5 + 1.0 + 0.5 + 0.75 points)

a) Why are CNNs well suited for images? Does it make sense to apply CNNs to text data as well? Why or why not? (2 sentences)

b) Since a convolutional layer applies a linear operation, it can be represented as a matrix multiplication. Represent the following convolutional layer as a matrix-vector multiplication:

$$\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} * \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}$$

Hint: Since the image is currently a 3×3 matrix, you need to flatten it to a 9-dimensional vector first.
Hint: The resulting matrix should be $4 \times 9$.

c) The effective receptive field $R_\ell$ at layer $\ell$ is the number of input units that produce a hidden unit in the feature map. For a fully convolutional network with kernel size $k$ and stride $s$, it can be calculated recursively as follows:

$$R_\ell = R_{\ell-1} + (k-1)\prod_{i=1}^{\ell-1} s_i,$$

We encourage you to think about why this is the case. By solving the recursion, show that this can be written in closed form as

$$R_\ell = 1 + (k-1)\sum_{j=1}^{\ell-1}\prod_{i=1}^{j} s_i.$$

You may assume that $R_0 = 1$, $s_i$ denotes the stride at layer $i$.

d) Consider a convolutional layer with 10 filters of size $k = 5$, i.e. the filters are $5 \times 5$ matrices, and stride $s = 1$. If the input is an image of size $32 \times 32 \times 3$, what will be:

a) the output dimensions?

b) the number of parameters?

c) the receptive field at layers $1, 2, 3, 4$?

**Exercise 10.3 - Residual Networks** (1.0 + 1.0 points)

Read the paper Deep Residual Learning for Image Recognition and answer the following questions:

(a) What is the problem with very deep networks? (2-3 sentences)

(b) How does introducing skip connections help? (2-3 sentences)

**Exercise 10.4 - Training & Fine-Tuning CNNs** (2+2.5 points)

In this exercise you build two CNN models to classify the labels in the german traffic sign recognition benchmark dataset. The training of the models should be done on the cluster. If you are unsure how convolutional layers can be used in PyTorch, checkout this tutorial.

a) First implement your own CNN, you are free to choose how many layers to use, which activation functions etc. Make sure to output the correct number of classes. Your model should reach at least 85% test set accuracy.

b) Finetune a pretrained model ResNet18 model on the dataset. Note that you need to modify the classifier of the model for the task at hand.

Train each model for at most 20 epochs. While training report the cross-entropy and accuracy for each epoch on the training and test set. **In addition to your source code submit the output file that is created on the cluster.**
If you have not yet set up the cluster checkout the tutorial [1].

[1] There also have been some slight improvements to the `run.sub` file, since the last exercise sheet.

**Exercise 10.5 - Bonus: SAM** (0.5 + 1.0 + 0.5 points)

Read the paper Sharpness-Aware Minimization for Efficiently Improving Generalization and answer the following questions:

a) What is the problem with sharp minima? (1-2 sentences)

b) What is the main idea of SAM? (2-3 sentences)

c) What form would $\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})$ take in the case of $L_2$-regularization? State the formula.