Universität des Saarlandes
Prof. Dr. Dietrich Klakow
Lehrstuhl für Signalverarbeitung
NNIA Winter Term 2018/2019

# Exercise Sheet 10

Optimization II

**Deadline: 28.01.2019, 23:59**

## Exercise 10.1 - SGD with Momentum (3 points)

Familiarize yourself with the SGD with momentum from the lecture slides (Chapter 8 slide 15) and DL book and understand how it works.

a) It is known that the cost function of NNs usually has many saddle points (cf. DL book chapter 8.2.3). How does SGD with momentum help to alleviate the problem of getting stuck at these saddle points when compared to vanilla SGD (Chapter 8 slide 13). Describe a situation in which vanilla SGD will get stuck at one saddle point, while SGD with momentum won't?

## Exercise 10.2 - Parameter Initialization (1 + 1 = 2 points)

a) We normally initialize the parameters in our neural networks using a Gaussian distribution with mean zero and a small variance. Why shouldn't we initialize the parameters with a large variance?

b) Can we initialize the parameters with the same constant (suppose that the constant is small enough s.t. you won't have the problem in a) )? Why/Why not?

## Exercise 10.3 - Batch Normalization (2.5 + 2.5 = 5 points)

In previous assignments/projects, we normally dealt with data whose features have roughly the same scaling, e.g. in a grayscale image each feature (grayscale of a pixel) is between 0 and 255. Now, let's consider data whose features have very different scaling. For an extreme case: consider the situation in which each data point consists of two features: weight ($x_1$) and height($x_2$). However, the unit for weight is kg, while the unit for height is mm (1 mm = $10^{-3}$ m).
Obviously the scales of these two features differ much. Consider a very simple loss function $L(w_1, w_2, b) = (y - (x_1 w_1 + x_2 w_2 + b))^2$, the (2D) contour line will have a very elongated ellipse shape (the condition number of the Hessian is very large), which will harm the convergence speed of gradient descent algorithm.

a) To deal with the problem above, we can easily standardize the input data by subtracting the mean and dividing by the standard deviation. However, in the case that we use a deep neural network, it could still happen that the features in some hidden layers have very different scaling, then the same problem might occur again. Therefore, one can standardize the features in all hidden layers in each iteration. That is, standardize the features before passing it to the next layer. This idea is called batch normalization. Now, consider that in practice we train on mini-batches instead of the entire dataset, we cannot access the mean and sd of the whole dataset [1]. What can we do in this case? Does your solution require a large batch size?

b) Actually, the idea of batch normalization introduced above is only half the story of the true batch normalization method. Let $x$ denote the value of a particular neuron, let $\hat{x}$ denote the value of that neuron after standardization. We will transform $\hat{x}$ to $\gamma\hat{x} + \beta$ before passing it to the next layer, where $\gamma$ and $\beta$ here are learnable parameters. Why do we need these two additional parameters?

## Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

- You have to submit the solutions of this assignment sheet as a team of 2-3 students.
- Hand in a **single** PDF file with your solutions.
- Therefore make sure to write the student ID and the name of each member of your team on your submission.
- Your assignment solution must be uploaded by only **one** of your team members to the course website.
- If you have any trouble with the submission, contact your tutor **before** the deadline.

Plagiarism of any form is not tolerated. If you refer something from the web, you must give proper credit by citing the source. Lack of this would be considered plagiarism. In such a case, the whole sheet would be awarded zero points and a warning is given. If this act is repeated again, then the whole team is excluded from the course.

[1] yes we should be able to compute mean and sd of for the input. However, we are not able to compute the mean and sd of the hidden units if we train on mini-batches