Universität des Saarlandes
Prof. Dr. Dietrich Klakow
Lehrstuhl für Signalverarbeitung
NNIA Winter Term 2019/2020

Exercise Sheet 6

Backpropagation and Geting to know PyTorch

(Proposed Solutions)

Deadline: 05.01.2021, 23:59

Instructions

The file *example.pdf* contains a sample solution for the following exercises performed on a simple linear regression. In the exercises you have to implement the same, but for a more complex function (defined below). Please, use the example as a guideline. Structure well your responses, use different colors for visualization purposes: this will make the grading process a lot easier.

You can perform the exercises by hand or use any applications and tools of your choice. The requirement is that it is comprehensible.

As you need to make drawings, there is no need to use this latex script for this exercise, you can create the pdf in any way you want. Make sure that you also attach the answer for question 6.5.

Exercises

In this exercise sheet we will continue working with the Titanic dataset. This time we will predict **price of the ticket** based on the *class of a passenger*, their *age* and whether they *survived* or not by training a neural network with one hidden layer.

Here is the **architecture** of the network:

- Number of hidden layers: 1
- Size of the hidden layers: 4
- Bias hidden: true
- Bias output: true
- Activation function on hidden layers: ReLU
- Activation function on the output layer: None
- Loss function: MSE

Exercise 6.1 - Network scheme (1 points)

Draw the scheme of the network defined above. Define all the variables and their dimensions (use real values when known).

Exercise 6.2 - Computation graph (2 points)

Draw the computation graph of the network. Relate the nodes of the network scheme to the nodes of the graph.

Potential issues:

- MSE representation: copy it from the *example*. Rename the intermediate nodes according to your graph if needed.

Points will be subtracted for unmarked (or wrongly marked) shape (dimensions) of each term in your graph.

Exercise 6.3 - Backpropagate (5 points)

Now we will compute the gradients of the weights and biases of out network.

a) Write down the paths for chained partial derivatives

b) Compute the derivatives based on the graph

c) Write the dimensions of each term

d) Reorder and transpose accordingly

e) Rewrite with the terms from the network scheme as the final formula

Potential issues:

- Activation functions are applied element-wise, so the derivative of activation function has the shape as the original input to the activation function. E.g. if we have $\sigma(a)$, where $a \in \mathbb{R}^{n*m}$, then $\sigma(a) \in \mathbb{R}^{n*m}$ and $\nabla_a \sigma(a) \in \mathbb{R}^{n*m}$. Hence, the derivative of the activation function is also applied **element-wise**.

Exercise 6.4 - Implement with numpy (we'll do during tutorial) (no points)

During the tutorial we will together implement the training of this neural network using plain numpy. The numerical example in the *example.pdf* illustrates one training pass. During the tutorial we will train our network for 10 epochs (passes).

For this, we will create a class Network. You can refresh how classes and objects in python work here. What attributes and methods should class Network have, so that we could implement the network training?

Exercise 6.5 - Implement in PyTorch (2 points)

You have seen that even such a small network requires defining solving a lot of equations. Imagine what happens with deep neural networks?

Read this article to understand how the gradients are computed in PyTorch. Summarize the procedure in 3-4 sentences.

Bonus (2 points):
Following this tutorial, custom a module in PyTorch for our network and train it.

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.
- Make sure to write the student teams e-mail 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.