Prof. Dr. Thomas Schultz
Shekoufeh Gorgi Zadeh (gorgi@cs.uni-bonn.de)

B-IT
Bonn-Aachen International
Center for Information Technology
June 18, 2018

Summer term 2018

Visual Computing in the Life Sciences

Assignment Sheet 6

If you have questions concerning the exercises, please write to our mailing list:
vl-bioinf@lists.iai.uni-bonn.de.

This exercise can be submitted in **small groups** of 2-3 students. Please submit each solution only once, but clearly indicate who contributed to it and remember that all team members have to be able to explain it.

Solutions have to be sent until ***July 02, 2018, 10:30am*** *to gorgi@cs.uni-bonn.de. Please bundle the results as PDF and scripts in a single ZIP file. Include your names explicitly in the PDF. Use the following format for naming the files "vclsi-x-lastName.pdf" and "vclsi-x-lastName.ipynb", where x is the ID of assignment sheet. For example, for Alice Smith, the file name must be vclsi-6-smith.pdf.*

Exercise 1 (Neural Networks, *35 Points*)

a) Download the data set `mnist.pkl.gz` from the lecture homepage. It contains 70000 images of handwritten digits as well as the labels for each image. Each label is a 10-dimensional vector, that is 0 everywhere except at the index that corresponds to the number in the image. Each image has a fixed size of 28×28 pixels. In this exercise you will use deep learning to read in an image of a hand written digit, to output that digit. For this, use `network.py`, `mnist_loader.py`, and `digit_classifier.py` from the lecture homepage to do the following tasks.

   Hint: More information on the data set and the original code of the framework can be found in `http://neuralnetworksanddeeplearning.com/chap1.html`. As part of turning it into this exercise, we converted it from Python 2 to 3.

b) Use *load_data_wrapper()* function from `mnist_loader.py` to read in the data into three different sets, training, validation and test. (2P) Write a function to visualize the first image in the training set and print out the corresponding label. (2P)

c) `digit_classifier.py` file shows how you could define a 2 layer network of input size 784, hidden layer of size 30 and output layer of size 10. You could train the network by calling `SGD(...)` function. Read through `network.py` and modify the code to extract classification accuracy for each epoch. Draw a diagram that shows classification accuracy over the validation data set at every epoch for 30 epochs. (6P)

d) Does multiplying all the weights and biases to a constant positive number change network's performance? Briefly explain your answer. (3P)

e) Write a function to feed in only one image (instead of a mini batch) into the network and to print out the values of the output layer of the network. (5P)

f) Briefly explain how the network accuracy is computed in `network.py`. (3P)

g) Perform network training on the training set using three different learning rates $\{0.001, 3, 100\}$ for 30 epochs each time. Compute networks performance on the validation set and draw them in one diagram. (3P) Which learning rate leads to best performance on validation set? (1P)

h) If the network is trained for a long time over the training set, it's performance on the validation set first increases but then it starts to gradually decrease. Briefly explain why it happens? (3P)

i) Increase number of the neurons in the second layer to 100. How does this affect classification accuracy? (4P)

j) Data shuffling is commonly used on training data before each training iteration. Here as well, in `SGD(...)` function, the input data is shuffled. Briefly explain why is it important to shuffle the data? (3P)

Good Luck!