Assignment 2: Neural Networks
MA-INF 2313: Deep Learning for Visual Recognition

Due Date Theoretical: 10.11.2017
Due Date Programming: 17.11.2017
Assistant: Fadime Sener `sener@cs.uni-bonn.de`

1 Theoretical Exercises *(15 pts)*

1. *(6 pts)* Neural networks can be interpreted geometrically as a partitioning of the input feature space $\boldsymbol{x}$ (recall lecture 3 and the formation of the AND/OR units). Now consider the trapezoid in Figure 1, which represents a decision boundary, where the region inside belongs to class A and the region outside belongs to class B. How should one design a two-layered network[1] to represent this trapezoid? Fully specify your network, including the form of input, hidden and output activation functions as well as associated parameters.

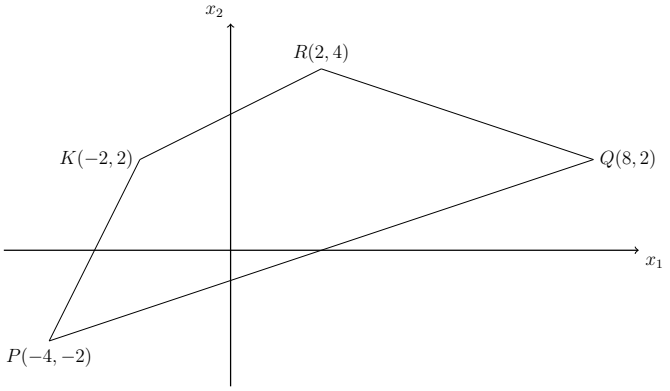

Figure 1: The region in the trapezoid belongs to class A and the other region to class B.

2. Consider the two-layered feed-forward network given in Figure 2 with two inputs x_1, x_2, two hidden units h_1, h_2, two output units o_1, o_2 and additionally, the hidden and output bias units b_1, b_2.

   In this task you are going to work with a single training set: given inputs 0.1 and 0.4, we want the neural network to output 0.1 and 0.9. You will use the Sigmoid function as your activation function for each hidden layer units h_1, h_2 and output units o_1, o_2.

   (a) *(3 pts)* Run a forward pass to see if the neural network with the current weights predicts the outputs correctly. Show your work step by step for the forward pass. Calculate the total error in the network using mean squared error.

[1] as per Bishop's notation, which results in one input layer, one hidden layer and one output layer.

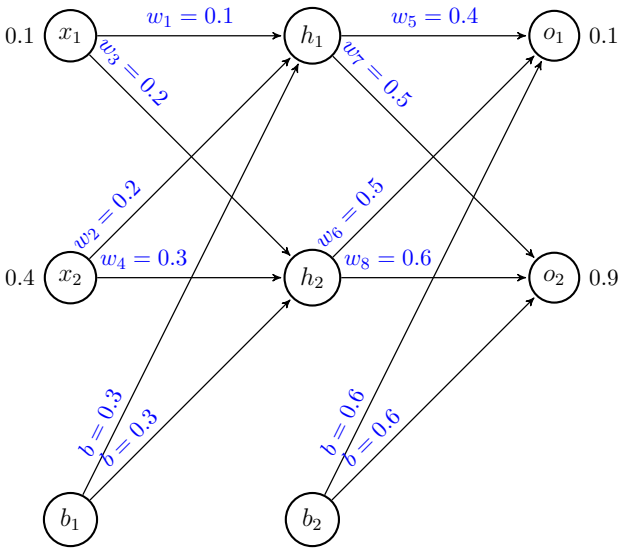

Figure 2: Two-layered feed-forward network

(b) *(6 pts)* Run backpropagation in this network and update each weight in the network. Show your work step by step for the backward pass. Later on, run a forward pass to see if the neural network with the updated weights predicts outputs better than in the first forward pass. (Set your learning rate to 0.5)

2 Programming Exercises *(15 pts)*

In the programming part, you will set up, train and test a multi-layered network for classification using TensorFlow. Write the code for a three-layered network (two hidden layers) and test your classifiers on the following two datasets.

1. *(6 pts)* First, classify hand-written digits on the MNIST dataset, according to the given training and testing splits on the website. Your code should plot the training loss over the iterations and report the final test accuracy.

2. *(9 pts)* For the second part, we would like to train a classifier to identify and distinguish between faces of 20 different people. The dataset is a subset of The ORL Database of Faces and is provided in the file `ORL_faces.npz.zip`. The size of each image is 92×112 pixels, with 256 grey levels per pixel. It is recommended to visualize the dataset as a sanity check. You can use `load_ORL_faces.py` to load the data. Use the given training (trainX, trainY) and test (testX, testY) splits. Your code should plot the training loss over the iterations and report the final test accuracy.