Intelligent Systems (IS Fall 2016)

Assignment 1: Neural networks
Deadline: Wed 5 Oct 2015, 17:00

In this assignment you will write code for running and training two neural networks. The assignment can be implemented in either Python *(recommended)* or MATLAB. For each of the tasks below it should be clear that you have to implement the algorithms yourself, i.e. you cannot use a neural network library. Unless otherwise mentioned use the *mean squared error* as your loss function.

The solutions and code must be archived in file named `<surname>.<name>` and uploaded to the iCorsi website **before** the deadline expires. Please make sure that upon delivery your code is readable, commented and can be executed.

1. **(40 points in total)** Write code to train the single-layer neural network shown in Figure 1.

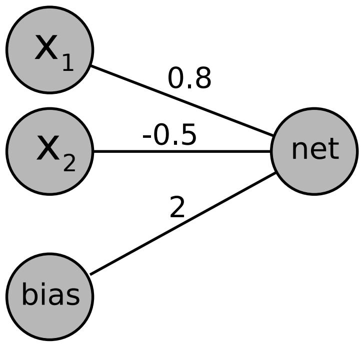

Figure 1: The perceptron network with the initial weight values to be used.

The training set consists of the following points:

x_1	1	6	3	4	3	1	6	7	6	10	4
x_2	8	2	6	4	1	6	10	7	11	5	11
z	1	1	1	1	1	1	-1	-1	-1	-1	-1

(a) **(3 points)** Plot the training set with the initial separation line where the two classes ($z = 1$ and $z = -1$) are displayed in different colours.

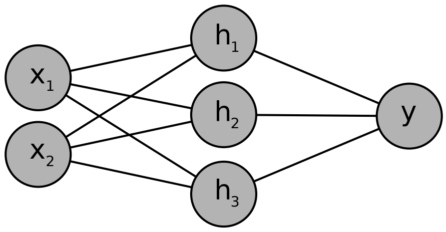

Figure 2: The perceptron network with the initial weight values to be used.

(b) **(10 points)** Derive the delta rule for this network.

(c) **(10 points)** Implement the delta rule and apply it for all points from the test set, with a learning rate of $\eta = \frac{1}{50}$.

HINT: your weights should now be (0.4396), (−0.7511) *and* (1.9522) *for the connections (x_1, net), (x_2, net) and (bias, net) respectively.*

(d) **(3 points)** Plot the new separation line.

(e) **(10 points)** Train the perceptron until **for the first time** all the points are correctly classified, and plot the final decision boundary line.

(f) **(4 points)** Is it a good solution? Discuss potential problems that may arise.

2. **(60 + 5 points in total)** Write code to train the two-layer feed-forward neural network with the architecture depicted in Figure 2.

The training set consists of the following data points:

x_1	4	4	5	5	7	1	2	3	6	3	6	4	7
x_2	2	4	3	1	2	2	1	1	5	6	7	6	6
z	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1

The accompanying test set is:

x_1	4	5	3	5	6	7	3	8	4	7	2	2
x_2	1	2	4	4	1	1	2	7	7	5	3	5
z	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1

This network uses a *sigmoidal* activation function (i.e. the logistic function) for the hidden layer and a *linear* activation for the output layer. Also, don't forget the biases (even though they are not shown in the image). The weights (and biases) should be randomly initialised from a uniform distribution in the range $[-0.1, 0.1]$.

(a) **(5 points)** Plot the data. Would the network be able to solve this task if it had linear neurons only? Explain why.

(b) **(10 points)** Give the formulas for the forward pass and derive the formulas for the backward pass for that network.

(c) **(5 points)** Express the formulas using matrix / vector operations.

(d) **(15 points)** Implement the forward and backward passes and train the network using gradient descent with a learning rate of $\eta = 0.1$ until all training examples are correctly classified. (This might take a few thousand epochs)

(e) **(10 points)** Perform 10 runs with different random initializations of the weights, and plot the training and test error for each epoch averaged across the 10 runs. How many epochs does it take, on average, to correctly classify all training points. What do you notice?

(f) **(15 points)** Try different learning rates $\eta = [10, 1, 1/10, 1/100, 1/1000]$ and plot the training error over 1000 epochs. What is the effect of varying the learning rate?

(g) **(Bonus: 5 points)** Vary the number of hidden units (1, 2, 4, 8, 16) and run each network with 10 different random initializations. Plot the average train and test errors. Explain the effect of varying the number of hidden units.

If you do not understand how to properly do any of the exercises you may discuss with your colleagues but you have to provide your own solution.

Feel free to ask your questions:

(a) At the Q&A forum

(b) By email: sjoerd@idsia.ch

(c) In any TA session