Artificial neural networks FFR135
Examples sheet 3

1. Boolean functions with three inputs. How many of the 2^{2^3} Boolean functions with three inputs can be represented by a single-layer, single-unit perceptron with bias? Describe how you arrive at the answer. Hint: visualise the function as a $2 \times 2 \times 2$ cube and look for linear separability. Use symmetries to reduce the number of cases. **(1p)**

2. Back-propagation of multilayer perceptron. The task is to solve a classification problem of a data set by training a multi-layer perceptron with back-propagation. The goal is to achieve a classification error on a given validation set as small as possible by training the network on the corresponding training set. The classification error C_{v} of the validation set is defined as

$$C_{\text{v}} = \frac{1}{p} \sum_{\mu=1}^{p} |\zeta^{(\mu)} - \text{sgn}(O^{(\mu)})|$$

where $|x|$ stands for the absolute value of x, p denotes the total number of patterns in the validation set, $\zeta^{(\mu)}$ is the target output for pattern μ, and $O^{(\mu)}$ is the network output for pattern μ. The classification error for the training set is defined similarly. Download from the web page two files containing training data ('train_data1') and the corresponding validation data ('valid_data1'). Each data set is to be read as follows. Patterns that you are asked to classify are given by the first two columns (each row being a different pattern). The element in the third column in each row represents the target output for the pattern in this row. To improve the training performance, normalise the input data to zero mean and unit variance.

2a. Train the network without hidden layers. Use asynchronous back- propagation with parameters: learning rate $\eta = 0.01$ activation-function parameter $\beta = 1/2$, weights initialised randomly with uniform distribution in $[-0.2, 0.2]$, biases initialised randomly with uniform distribution in $[-1, 1]$. Take activation functions to be *tanh*. Iterate for at least 10^6 iterations (one iteration corresponds to feeding one randomly chosen pattern). Perform ten independent training runs. For each training, plot the energy in the training and in the validation set obtained during training. Discuss the trends observed. Explain how the energy is expected to change over time according to the gradient-descent rule. Discuss the effect of η: how do you expect it to change the trends obtained. (You can perform additional trainings with different values of η to support your conclusions.) **(1p)**

2b. Plot the classification errors obtained during these ten training experiments both for the training and for the validation set. Discuss how well the network performs. Is there a noticeable difference between the results obtained in the different runs you made? Explain why this is the case. **(1p)**

2c. Now train a network with one hidden layer that has three neurones. As in task **2a**, perform ten independent runs. Plot the energy during training in both the validation and the training set. Discuss the results in comparison to those obtained in **2a**. Is the performance better? **(1p)**

2d. Make a table showing the classification error at the end of each training for both the training and the validation set. Discuss your results. **(1p)**

2e. Now train a network with an additional hidden layer (thus, two hidden layers in total). Use two hidden neurones in the second hidden layer. Perform training ten times. Plot the classification error obtained during each training in both the training and the validation set. Discuss your results. Is the performance better than with one hidden layer? Explain why this is so. **(1p)**