Neural networks
Statistics, Optimisation, and Learning
Examples sheet 1

1. Deterministic Hopfield model. Write a computer program implementing the Hopfield model with synchronous updating (LN[1] pp. 10 and 14) according to the McCulloch-Pitts rule

$$S_i \leftarrow \operatorname{sgn}\left(\sum_j w_{ij} S_j\right) .$$

The weights w_{ij} are given by Hebb's rule for $i \neq j$, and $w_{ii} = 0$. Store p random patterns (LN pp. 16 and 19) and use your computer program to find how the one-step error probability P_{Error} depends on p/N. Parameters: $N = 100,\ 200$ and $p = 10,\ 20,\ 30,\ 40,\ 50,\ 75,\ 100,\ 150,\ 200$. For each data point average over enough independent trials so that you obtain a precise estimate of P_{Error}. Plot your simulation results together with the corresponding theoretical curve as a function of p/N. Discuss. (**1p**).

2. Recognising digits with the deterministic Hopfield model. The task is to recognise distorted versions of the digits shown in Fig. 1 using the deterministic Hopfield model with asynchronous updating. Create binary representations of the digits shown in Fig. 1 and store these patterns. Then distort the patterns by flipping a fraction q of randomly chosen bits. Feed these patterns to the network and use asynchronous updating. For each digit, repeat many times. Determine and plot the probability that the network retrieves the correct pattern as a function of q. Discuss. (**1p**).

3. Stochastic Hopfield model. Write a computer program implementing the Hopfield model with stochastic updating (LN pp. 36). Use Hebb's rule and take $w_{ii} = 0$. Use random patterns (LN pp. 16 and 19).
a. For $\beta = 2$ determine the steady-state order parameter m_1 as a function of $\alpha = p/N$ for $N = 500$. Discard data points corresponding to the initial transient of the stochastic dynamics. Average over independent realisations. Plot the order parameter as a function of α in the range $0 < \alpha \leq 1$. Discuss your results. Refer to the phase diagram (LN p. 60). (**1p**).
b. Repeat the above for $N = 50, 100,$ and 250 choosing p so that α is in the same range as above. Discuss how the value of N influences the order parameter m_1. (**1p**).

4. Backpropagation. The task is to write a computer program that solves a classification task by backpropagation. Train the network on the training set 'train_data_2016', and evaluate its performance using the validation set

[1] LN stands for the lecture notes.

'valid_data_2016' (course home page). Each row corresponds to one pattern. The first two entries in a row give $\xi_1^{(\mu)}$ and $\xi_2^{(\mu)}$, the third entry gives the target value for this pattern, $\zeta^{(\mu)}$.
The aim is is to achieve a classification error that is as small as possible. The classification error C_v for the validation set is defined as

$$C_\text{v} = \frac{1}{2\,p}\sum_{\mu=1}^{p} |\zeta^{(\mu)} - \text{sgn}(O^{(\mu)})|$$

where $|x|$ stands for the absolute value of x, p denotes the number of patterns in the validation set, and $O^{(\mu)}$ is the network output for pattern μ. The classification error for the training set is defined analogously.
a. Normalise the input data to zero mean and unit variance (together for both the validation and the training set). Train the network without hidden layers. Use asynchronous updating with learning rate $\eta = 0.01$, activation function $g(b) = \tanh(\beta b)$ with $\beta = 1/2$. Initialise the weights randomly with uniform distribution in $[-0.2, 0.2]$. Initialise the biases randomly with uniform distribution in $[-1, 1]$. Make 100 independent training experiments, each with $2 \cdot 10^5$ iterations (one iteration corresponds to feeding a randomly chosen pattern and updating weights and thresholds). For each training experiment determine the minimum classification error for the training and the classification sets. Average these errors over the independent training experiments. Discuss your results. (**1p**).
b. Now use back propagation to train a network with one hidden layer that has 2 neurons, same activation function and parameters as in **a**. Perform 100 independent training experiments with up to $2 \cdot 10^5$ iterations. For each experiment find the minimum classification errors for the training and validation sets. Determine the average of the minimum errors for both sets. Repeat for networks with 4, 8, 16, and 32 neurons in the hidden layer. Plot the average errors as a function of the number of hidden neurons. Compare to the results obtained in **a**. Discuss. Does the hidden layer improve the performance? What is the effect of increasing the number of hidden neurons? Do you observe overfitting? (**1p**)

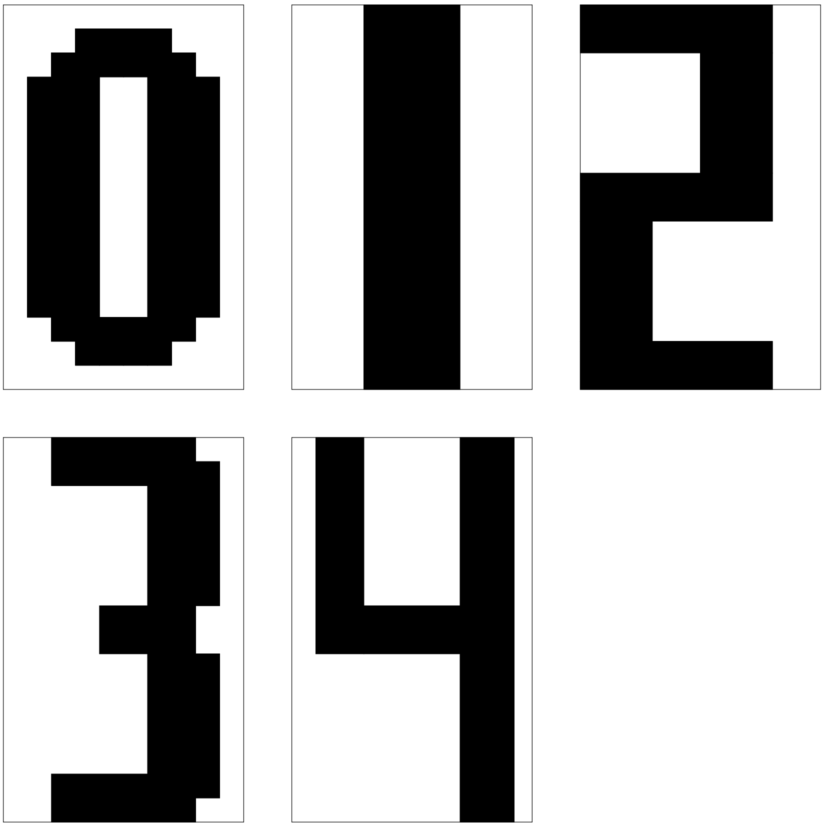

Figure 1: Binary representations of digits (after Fig. 14.17 in S. Haykin, Neural Networks, 2nd edition, Prentice Hall, New Jersey (1999)). Each digit contains 16x10 pixels.