Neural networks FFR135
Extra examples sheet for re-examination

1. Construct a one-dimensional Kohonen network to learn the properties of a distribution that is uniform inside an equilateral triangle with sides of unit length, and zero outside. [Hint: to generate this distribution, generate at least 1000 points uniformly distributed over the smallest square that contains the triangle, and then accept only points that fall inside the triangle. These points are the input to your network.] Your network needs to have at least 100 output nodes. There are two learning phases. In the first phase (*ordering phase*) you should use a time-dependent Gaussian width $\sigma(t)$ of the neighbouring function, as well as a time-dependent learning rate $\eta(t)$. For the former, one uses:

$$\sigma(t) = \sigma_0 \exp(-\frac{t}{\tau_\sigma}) \ . \tag{1}$$

Here, σ_0 is the parameter that is typically set to correspond to the largest distance in the output lattice. In your one-dimensional network with 100 output nodes, you can use $\sigma_0 = 100$. Furthermore, the total learning time in the ordering phase $T_{\rm order}$ should be chosen so that $\sigma(T_{\rm order}) \approx 1$. Thus, the parameter τ_σ is usually chosen to satisfy $\tau_\sigma = T_{\rm order}/\ln[\sigma_0]$. In your simulations, you can use $T_{\rm order} = 10^3$, but you are welcome to experiment with other settings. For the learning rate $\eta(t)$ one uses:

$$\eta(t) = \eta_0 \exp(-\frac{t}{\tau_\sigma}) \ . \tag{2}$$

The parameter η_0 is set by the user. A suggestion is to use $\eta_0 = 0.1$. After the ordering phase, you should perform fine tunning of the input data (*convergence phase*). In this phase, the learning rate $\eta_{\rm conv}$, and the Gaussian width $\sigma_{\rm conv}$ of the neighbouring function are kept constant, and small. A suggestion is to use $\sigma_{\rm conv} = 0.9$, and $\eta_{\rm conv} = 0.01$. The total learning time $T_{\rm conv}$ in the convergence phase is typically long (say, $T_{\rm conv} = 5 \cdot 10^4$ steps).

1a. Using the settings suggested above, plot the weight vectors corresponding to the 100 output nodes that you obtain after the ordering, and after the convergence phase, together with the desired input triangle (show two panels in a single figure). Discuss: does the network recognise the triangle? What are the differences between the ordering and convergence phase? **(1p)**

1b. How does the parameter σ_0 influence the results? To answer this question, you need to perform additional experiments with different values of σ_0. Show in one plot (with two panels) the results obtained after the ordering and convergence phase for $\sigma_0 = 10$ (similarly to the plot in **1a.**). Explain the results obtained. Which setting works better? **(1p)**

2. Learn a two-dimensional Kohonen network with two-dimensional output array with 20×20 units to recognise wine classes in the data set on the

course home page. Download two files from the course home page: a data set, and a text explaining the variables. The data set is adapted from one of the classification problems on the UCI Machine Learning Repository [1]. In the data set, the first column is a classification and should be ignored. [Hint: for each remaining column in the data set, normalise to zero mean and unit variance.]

Show how the winning neurones for the different classes are distributed at the end of the convergence phase. This can be done by displaying lattice positions of winning neurones for each class and by colouring different classes differently. Suggested parameter values: $\sigma_0 = 30$, $\eta_0 = 0.1$, $T_{\text{order}} = 10^3$, $\sigma_{\text{conv}} = 0.9$, $\eta_{\text{conv}} = 0.01$, $T_{\text{conv}} = 2 \cdot 10^4$. Discuss your results. Does the network group wines of the same class together? **(1p)**

3. In this example, you train a network to predict the time-series of the intensity of the output of a laser, exhibiting low-dimensional dynamics. The data is in the form of a time series of the intensity of the laser (download the description file from the course web page for more information). The task is to reconstruct the dynamics underlying the data set.

In order to reconstruct the dynamics, one may use the embedding theorem: assuming there is an n-th-order differential equation describing the dynamics one may find a mapping from the intensity values $I(t-\mathrm{d}t), \ldots, I(t-n\mathrm{d}t)$ to $I(t)$, where t is the time and $\mathrm{d}t$ is the sampling period.

3a. Discuss under which circumstances and to which extent it may be possible (and thus meaningful) to predict a time series. To which extent the present example allows to predict the time series? To answer these questions, you need first to visualise the data given. Assuming $\mathrm{d}t$ corresponds to the time between two consecutive samples in the training data file (available from the course web page), plot $I(t)$ versus $I(t-\mathrm{d}t)$ in one panel (as points), and $I(t)$ versus $I(t-\mathrm{d}t)$ and $I(t-2\mathrm{d}t)$ in another panel. Discuss the results obtained. **(1p)**

3b. Construct a network which takes $I(t)$ and $I(t-\mathrm{d}t)$ and tries to predict $I(t+\mathrm{d}t)$. In this case, it is suitable to use a linear output unit and *tanh* units in the hidden layers. Why? Use one or two hidden layers with 5 units in each. Train the network 10 times and show how the training and validation energy depend on time. Make two panels, one for one hidden layer, and the other for two hidden layers. Experiment with parameter values and show the results obtained for the parameter values that seemed to work best in your experiments. Discuss how changing the parameter values influences the results. **(1p)**

3c. During training, measure and plot how well the networks predict $I(t+k\mathrm{d}t)$ for $k = 1, ...20$ (you can measure the relative error). Discuss your results. How do you expect the error to change as k increases? Does this expectation agree with your results? **(1p)**

References

[1] The UCI Machine Learning Repository, University of California, Irvine, USA, https://archive.ics.uci.edu/ml