Tik-61.3030 Principles of Neural Computing

Raivio, Venna

Exercise 6

1. Construct a MLP network which is able to separate the two classes illustrated in Figure 1. Use two neurons both in the input and output layer and an arbitrary number of hidden layer neurons. The output of the network should be vector $[1, 0]^T$ if the input vector belongs to class $\mathcal{C}_1$ and $[0, 1]^T$ if it belongs to class $\mathcal{C}_2$. Use nonlinear activation functions, namely McCulloch-Pitts model, for all the neurons and determine their weights by hand without using any specific learning algorithm.

   (a) What is the minimum amount of neurons in the hidden layer required for a perfect separation of the classes?

   (b) What is the maximum amount of neurons in the hidden layer?

2. The function
$$t(x) = x^2, \quad x \in [1, 2]$$
is approximated with a neural network. The activation functions of all the neurons are linear functions of the input signals and a constant bias term. The number neurons and the network architecture can be chosen freely. The approximation performance of the network is measured with the following error function:
$$\mathcal{E} = \int_1^2 [t(\mathbf{x}) - y(\mathbf{x})]^2 d\mathbf{x},$$
where $\mathbf{x}$ is the input vector of the network and $y(\mathbf{x})$ is the corresponding response.

   (a) Construct a single-layer network which minimizes the error function.

   (b) Does the approximation performance of the network improve if additional hidden layers are included?

3. The MLP network of Figure 2 is trained for classifying two-dimensional input vectors into two separate classes. Draw the corresponding class boundaries in the (x_1, x_2)-plane assuming that the activation function of the neurons is (a) sign, and (b) tanh.

4. Show that (a)
$$\Delta w_{ij}(n) = \eta \sum_{t=0}^{n} \alpha^{n-t} \delta_j(t) y_i(t)$$
is the solution of the following difference equation:
$$\Delta w_{ij}(n) = \alpha \Delta w_{ij}(n-1) + \eta \delta_j(n) y_i(n),$$
where α is a positive momentum constant. (b) Justify the claims 1-3 made on the effects of the momentum term in Haykin pp. 170-171.

5. Consider the simple example of a network involving a single weight, for which the cost function is
$$\mathcal{E}(w) = k_1(w - w_0)^2 + k_2,$$
where w_0, k_1, and k_2 are constants. A back-propagation algorithm with momentum is used to minimize $\mathcal{E}(w)$. Explore the way in which the inclusion of the momentum constant α influences the learning process, with particular reference to the number of epochs required for convergence versus α.

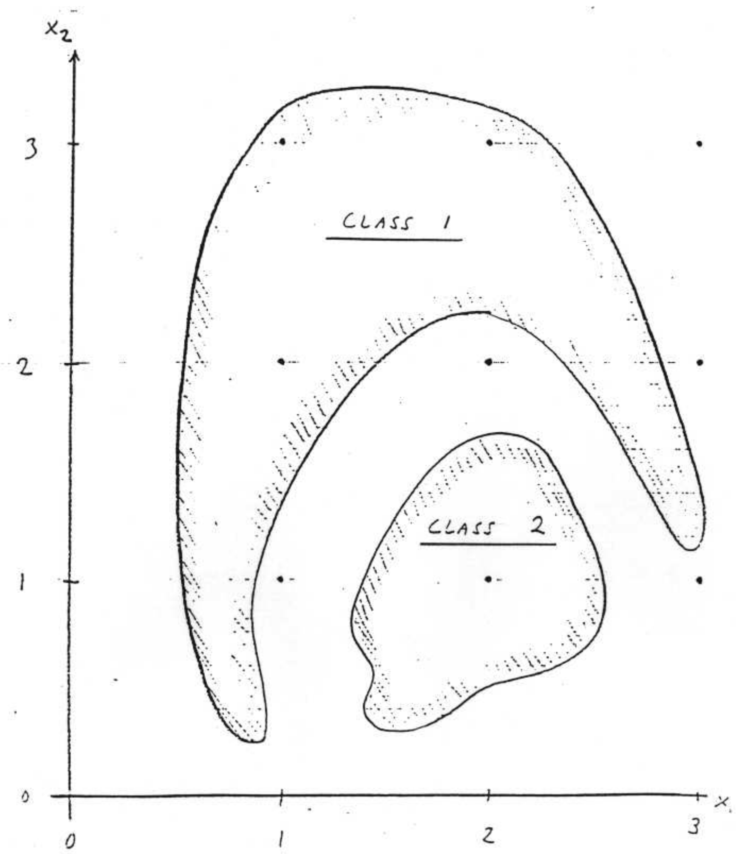

Figure 1: Classes $\mathcal{C}_1$ and $\mathcal{C}_2$.

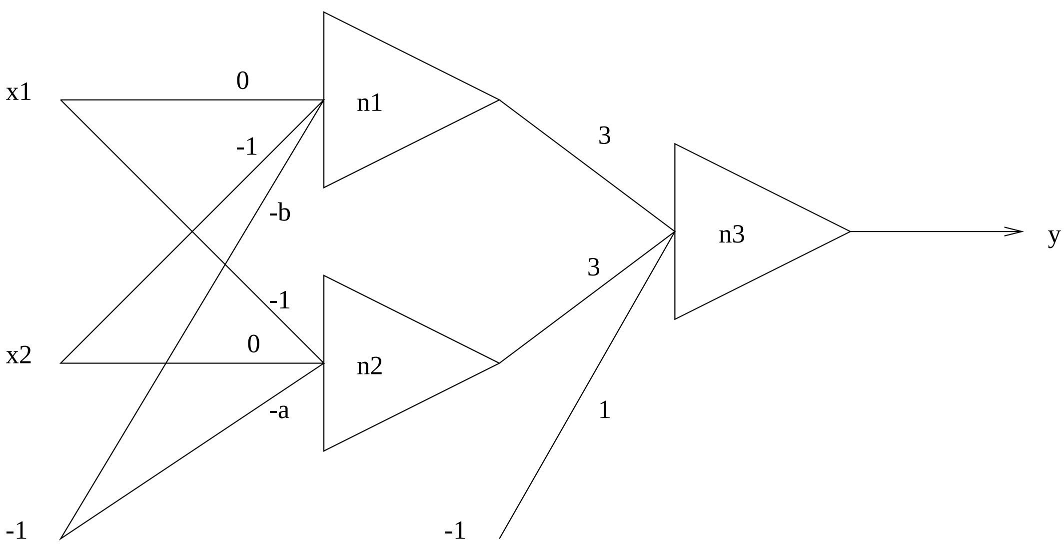

Figure 2: The MLP network.