# Backpropation learning: Exercise 5

This exercise is based on C.M.Bishop: *Neural Networks for Pattern Recognition*, Chapter 4.8, 7.1 and 7.5.

The objective of this exercise is to get a feel for the issues related to initialization, the parameter tuning and stop criteria for gradient descent learning of neural networks.

Print and comment on the figures produced by the software `main5a.m` and `main5c.m` as outlined below at the three **Checkpoints**.

## Neural network model

Let $\mathbf{y}(\mathbf{x})$ be a function of the vector $\mathbf{x}$, where $\mathbf{x} = (x_1, \ldots, x_d)^\top$. The functional form of $\mathbf{y}(\mathbf{x})$ is a neural network (NN) with one hidden layer as shown in figure 1. We have a data-set, $\mathcal{D} = \{(\mathbf{x}^n, \mathbf{t}^n)\}$, $n = 1, \ldots, N$ of $N$ corresponding values of $\mathbf{x}$ and $y(\mathbf{x})$, and we wish to learn the optimal parameters the training data.

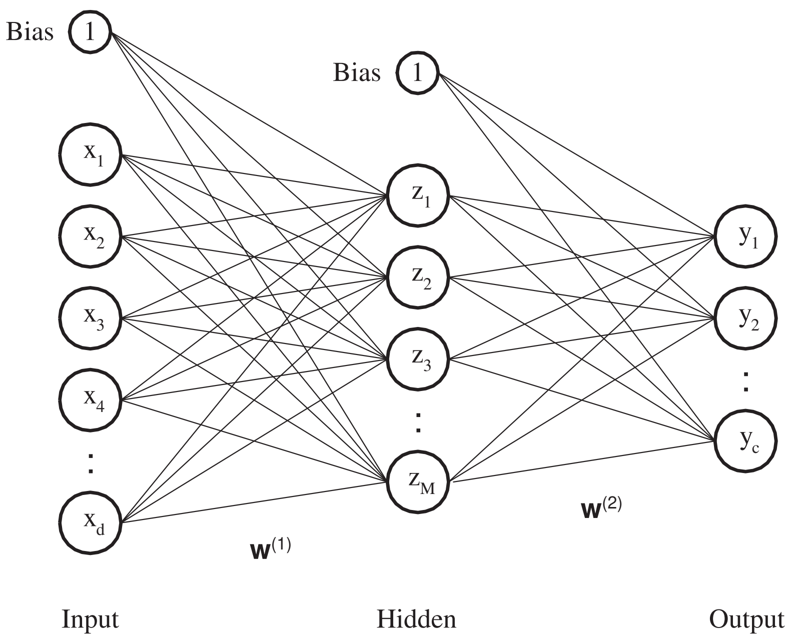


Figure 1: Neural network with one hidden layer.

The function is given by

$$y_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j \quad , \quad z_0 \equiv 1 \tag{1}$$

$$z_j = g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right) \quad , \quad x_0 \equiv 1 \tag{2}$$

where $y_k$ is the $k$'th output of the model, $z_j$ is the the $j$'th output of hidden units and $g(\cdot)$ is a nonlinear sigmoid function. The parameters that we a going to estimate are usually called weights. They are given as $\mathbf{w}^{(1)}$ for the hidden layer and $\mathbf{w}^{(2)}$ for the output layer. It should be noted that a bias is included in the output and hidden layer by having $z_0$ and $x_0$ set to one. In matrix notation this can also be expressed as

$$\mathbf{y} = (\mathbf{w}^{(2)})^\top \mathbf{z} \ , \ \ z_0 \equiv 1 \tag{3}$$
$$\mathbf{z} = g\left((\mathbf{w}^{(1)})^\top \mathbf{x}\right) \ , \ \ x_0 \equiv 1 \tag{4}$$

The optimal weights **w**are found through minimizing an error function. Here we shall use the sum-of-squares error function augmented by a weight-decay term

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y(\mathbf{x}^n;\mathbf{w})_k - t_k^n\}^2 + \frac{1}{2}\alpha\mathbf{w}^2 \tag{5}$$

The weight decay $\alpha$ is a control parameter that controls the value of the weights and can contribute to hinder overfit to the noise in the data set.

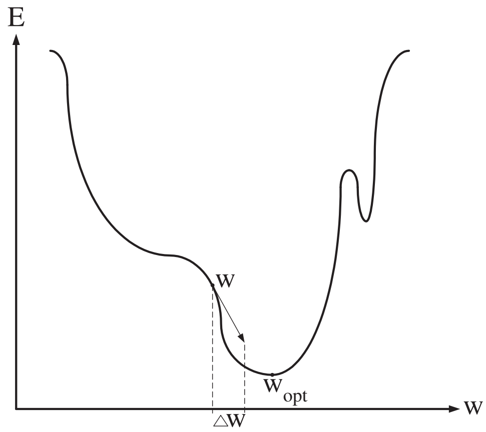


Figure 2: The weight update is done by gradient decent going the opposite direction of the cost functions gradient.

The optimum weights for the model can be found by a iterative scheme taking steps in the weight space that minimizes the error

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \triangle\mathbf{w} \tag{6}$$

,where $\eta$ is the *step size* parameter that determines how long a step $\triangle\mathbf{w}$ should be. One way to determine the update $\triangle\mathbf{w}$, is to move the weights in the opposite direction of the error function gradient, (5),

$$\triangle\mathbf{w} = -\eta\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} \tag{7}$$

This method is called *gradient decent* and is illustrated in figure 2. Differentiating (5) with the respect to the weights in the output and hidden layer gives

$$\triangle w_{kj} = -\eta\sum_n (y_k - t_k)z_j + \alpha w_{kj} \tag{8}$$

$$\triangle w_{ji} = -\eta\sum_n \left((1-z_j^2)\sum_{k=1}^{c} w_{kj}(y_k - t_k)\right) x_i + \alpha w_{ji} \tag{9}$$

using that the nonlinear function $g(a) = \tanh(a)$ shown in figure 3.

In the following sections we want to investigate some of the parameters involved in the optimization.

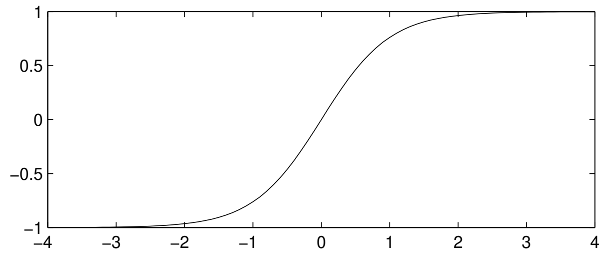


Figure 3: The nonlinear function $g(\cdot) = \tanh(\cdot)$.

## Initialization range

The gradient descent process starts from an initial set of values for the weights. The role of the initialization range is important for the convergence of the parameters. If the range is making the initial weights too small compared to the signal, the nonlinear function $g(\cdot)$ will be close to a linear mapping. On the other hand, setting the range to high forces the nonlinear function is close to step function.

### Checkpoint 5.1:

Use the program main5a.m to create a NN training-set with a 4-dimensional input variable, 5-dimensional hidden variable and a 1-dimensional output variable ($4 \times 5 \times 1$). Observe the effects of setting the range of the initial weights to: 0, 0.5 and 10000.

## Step size parameter

The step size parameter $\eta$ can be chosen in various ways and in this exercise we use fixed value throughout the iteration scheme, this is not optimal, and we shall look at better schemes in exercise 6. When the step size is large the NN can 'overshoot'. To avoid one might choose a small step size value, but this tends to make the NN converge more slowly.

### Checkpoint 5.2:

Use the program main5a.m to create a NN training-set with a 4-dimensional input variable, 5-dimensional hidden variable and a 1-dimensional output variable ($4 \times 5 \times 1$). Set the step size to different values and observe the behavior. What is a good step size for the data-set?.

## Stop criteria

Various methods can be used to stop the iteration process. The stop criterion depends on the application here we will consider two possible candidates.

**Checkpoint 5.3:**

Use the program `main5c.m` to plot the test and train error, the length of the gradient vector, and the training error difference between iterations. Comment on the plots and how they could be used as a stop-criteria.