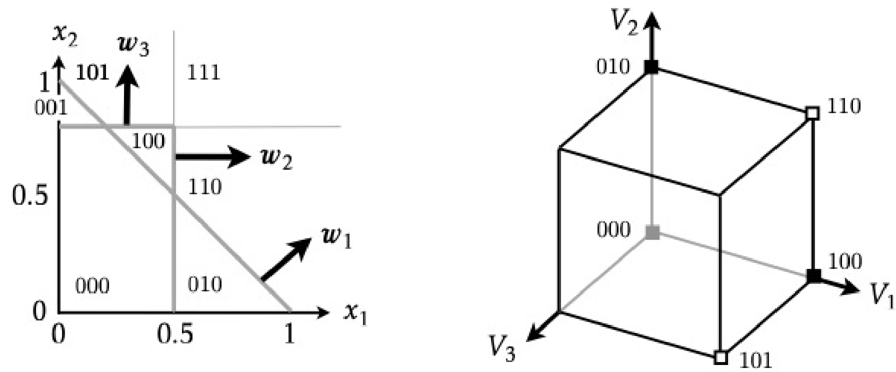

Figure 1: Left: weights and decision boundaries in the input plane, Exercise 5.8. Right: output problem.

Solutions to exercises in Chapter 5

5.8 Multilayer perceptron. Weight vectors for the three decision boundaries in Figure **5.23** are shown in Figure 1. From Equation (**5.13**) we infer

$$\boldsymbol{w}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \theta_1 = 1\,, \quad \boldsymbol{w}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \theta_2 = \tfrac{1}{2}\,, \quad \boldsymbol{w}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \theta_3 = \tfrac{4}{5}\,. \tag{1}$$

The resulting output problem is shown on the right in 1. It can be solved by a decision boundary that contains the points $\boldsymbol{x}_1 = [\frac{1}{2}, 1, 0]^{\mathsf{T}}$, $\boldsymbol{x}_2 = [1, \frac{1}{2},, 0]^{\mathsf{T}}$, and $\boldsymbol{x}_3 = [0, 0, 1]^{\mathsf{T}}$. Equation (**5.13**) gives three conditions for these three points:

$$W_1 + \tfrac{1}{2} W_2 = \Theta\,, \quad W_2 + \tfrac{1}{2} W_1 = \Theta\,, \quad \text{and} \quad W_3 = \Theta\,. \tag{2}$$

The solution is $\boldsymbol{W} = [\frac{2}{3}\Theta, \frac{2}{3}\Theta, \Theta]^{\mathsf{T}}$. To map the origin $\boldsymbol{V} = [0, 0, 0]^{\mathsf{T}}$ to output $O = 1$, we must choose a negative threshold, for example $\Theta = -1$. In this case, the output neuron calculates $O = \theta_{\mathrm{H}}(-\frac{2}{3} V_1 - \frac{2}{3} V_2 - V_3 - 1)$.

5.11 Non-linear activation function. A *linear* unit can solve a classification problem $O_i^{(\mu)} = t_i^{(\mu)}$ ($i = 1, \ldots, N$ and $\mu = 1, \ldots, p$) if the inverse of the overlap matrix (**5.22**) exists, if its columns are linearly independent. This requires linearly independent patterns $\boldsymbol{x}^{(\mu)}$, and therefore $p \leq N$. Introducing a nonlinear, monotonically increasing activation function $g(b)$, such as the sigmoid function, does not help. Since the activation function is monotonically increasing, it can be inverted to map the targets $g^{-1}(t_i^{(\mu)})$. Applying g^{-1} to the network output results in a linear function. So solving the classification problem requires that there are at most N patterns.

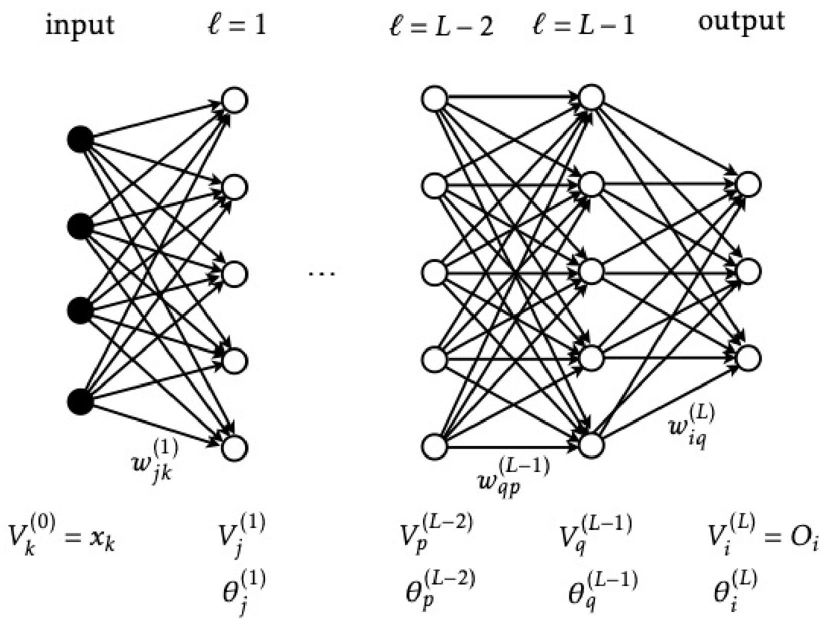

Figure 2: Network layout for Exercise .

Solutions to exercises in Chapter 6

6.7 Multi-layer perceptron. The network is drawn in Figure 2. First, to compute the recursion for the derivatives of $V_i^{(\ell)}$ with respect to $w_{mn}^{\ell'}$ for $\ell' < \ell$, one uses the chain rule

$$\frac{\partial V_i^{(\ell)}}{\partial w_{mn}^{(\ell')}} = \frac{\partial}{\partial w_{mn}^{(\ell')}} g\Big(\sum_j w_{ij}^{(\ell)} V_j^{(\ell-1)} - \theta_i^{(\ell)}\Big) = g'(b_i^{(\ell)}) \sum_j w_{ij}^{(\ell)} \frac{\partial V_j^{(\ell-1)}}{\partial w_{mn}^{(\ell')}}\,. \tag{3}$$

Second, for $\ell' = \ell$ the result is different. Note that $V_j^{(\ell-1)}$ does not depend on $w_{mn}^{(\ell)}$ because of the feed-forward layout of the network (Figure 2). Therefore

$$\frac{\partial V_i^{(\ell)}}{\partial w_{mn}^{(\ell)}} = g'(b_i^{(\ell)}) \sum_j \frac{\partial w_{ij}^{(\ell)}}{\partial w_{mn}^{(\ell)}} V_j^{(\ell-1)} = g'(b_i^{(\ell)}) \delta_{im} V_n^{(\ell-1)}\,. \tag{4}$$

This is analogous to Equation (**6.7d**). Third, put these results together to derive the learning rule for layer $L-2$. We feed pattern $\boldsymbol{x}^{(\mu)}$ and minimise $H = \frac{1}{2}\sum_i (t_i^{(\mu)} - O_i^{(\mu)})^2$, dropping the index μ in the following.

$$\delta w_{mn}^{(L-2)} = \eta \sum_i (t_i - V_i^{(L)}) \frac{\partial V_i^{(L)}}{\partial w_{mn}^{(L-2)}} \tag{5}$$

Iterating twice with the recursion (3) and then using (4) gives

$$\delta w_{mn}^{(L-2)} = \eta \sum_i (t_i - V_i^{(L)}) g'(b_i^{(L)}) \sum_j w_{ij}^{(L)} g'(b_j^{(L-1)}) w_{jm}^{(L-1)} g'(b_m^{(L-2)}) V_n^{(L-3)} . \tag{6}$$

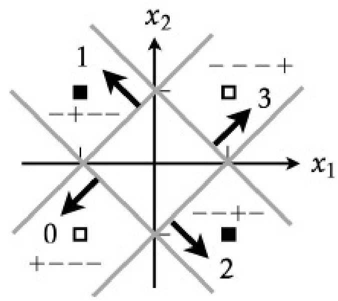

Figure 3: Shows solution of XOR problem. Exercise 7.2

Solutions to exercises in Chapter 7

7.2 Decision boundaries for XOR problem. The solution is shown in Figure 3 (see also Figure 7.6). The four weight vectors $\boldsymbol{w}_j$ for $j = 0, 1, 2, 3$ are obtained from Equation (7.6):

$$\boldsymbol{w}_0 = \begin{bmatrix} -\delta \\ -\delta \end{bmatrix}, \quad \boldsymbol{w}_1 = \begin{bmatrix} -\delta \\ \delta \end{bmatrix}, \quad \boldsymbol{w}_2 = \begin{bmatrix} \delta \\ -\delta \end{bmatrix}, \quad \boldsymbol{w}_3 = \begin{bmatrix} \delta \\ \delta \end{bmatrix}, \tag{7}$$

The thresholds are all the same. The intersections of the decision boundaries with the x_2-axis are determined by Equation (5.13). The 4-digit codes describe the output of the hidden neurons, one verifies that the output layer $O = \mathrm{sgn}(-V_0 + V_1 + V_2 - V_3)$ does the trick.

7.4 Residual network. We start with Equation (7.30),

$$\delta^{(L-1)} = \delta^{(L)} w^{(L,L-1)} g'(b^{(L-1)}). \tag{8}$$

The error $\delta^{(L-2)}$ is obtained using the recursion (7.33):

$$\delta^{(\ell-1)} = \delta^{(\ell)} w^{(\ell,\ell-1)} g'(b^{(\ell-1)}) + \delta^{(\ell+1)} w^{(\ell+1,\ell-1)} g'(b^{(\ell-1)}), \tag{9}$$

valid for $\ell - 1 \leq L - 2$. This recursion reflects that every neuron $\ell - 1 < L - 2$ can be reached backwards directly from ℓ, and also from $\ell + 1$ via a skipping connection. So we have for $\delta^{(L-2)}$:

$$\delta^{(L-2)} = \delta^{(L-1)} w^{(L-1,L-2)} g'(b^{(L-2)}) + \delta^{(L)} w^{(L,L-2)} g'(b^{(L-2)}). \tag{10}$$

Iterating once more yields three terms for $\delta^{(L-3)}$:

$$\begin{aligned} \delta^{(L-3)} = {} & \delta^{(L)} w^{(L,L-1)} g'(b^{(L-1)}) w^{(L-1,L-2)} g'(b^{(L-2)}) w^{(L-2,L-3)} g'(b^{(L-3)}) \\ & + \delta^{(L)} w^{(L,L-2)} g'(b^{(L-2)}) w^{(L-2,L-3)} g'(b^{(L-3)}) \\ & + \delta^{(L)} w^{(L,L-1)} g'(b^{(L-1)}) w^{(L-1,L-3)} g'(b^{(L-3)}). \end{aligned} \tag{11}$$

Each term in this expression corresponds to one of all possible paths from L to $L-3$,

$$\begin{aligned}
&L \to \ell_1 = L-1 \to \ell_2 = L-2 \to L-3\,,\\
&L \to \ell_1 = L-2 \to L-3\,,\\
&L \to \ell_1 = L-1 \to L-3\,.
\end{aligned}\tag{12}$$

The first path visits $n=2$ intermediate neurons, it has no skipping connections. The other two paths have one skipping connection, each of them visits only $n=1$ intermediate neuron. There are no paths that involve two or more skipping connections. We conclude that the error $\delta^{(L-3)}$ can be written as a sum over all paths, as stated in Equation (**7.34**). The general form of Equation (**7.34**) is obtained by iterating backwards using Equation (9).