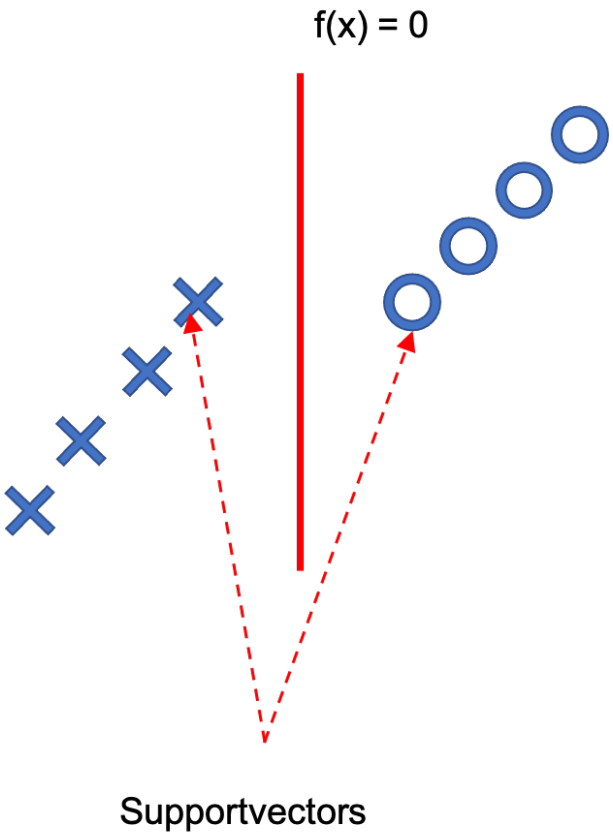
Solutions to the exam in
Neural Networks and Learning Systems - TBMI26 / 732A55
Exam 2021-03-19

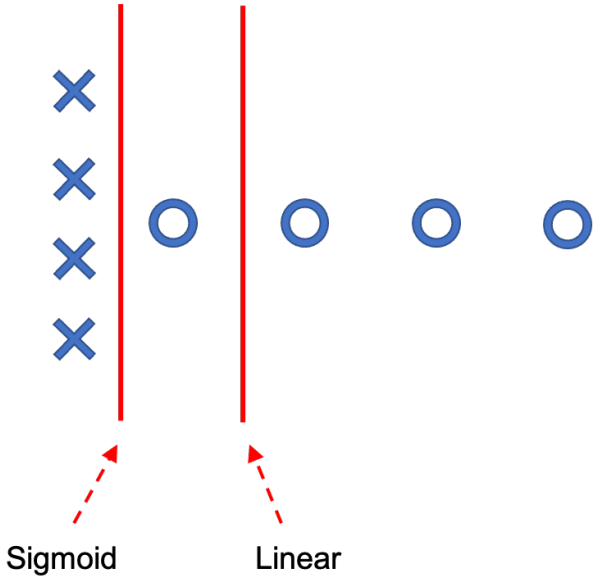
Part 1

1. In supervised learning the training data consist of pairs of input an correct output, e.g. labels. In unsupervised learning, there are no correct output in the training data.

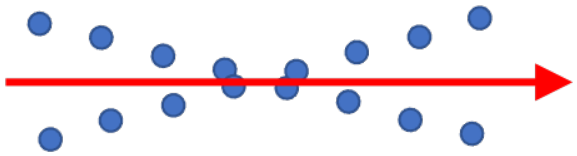
2. See figure:



3. See figure:



4. See figure:

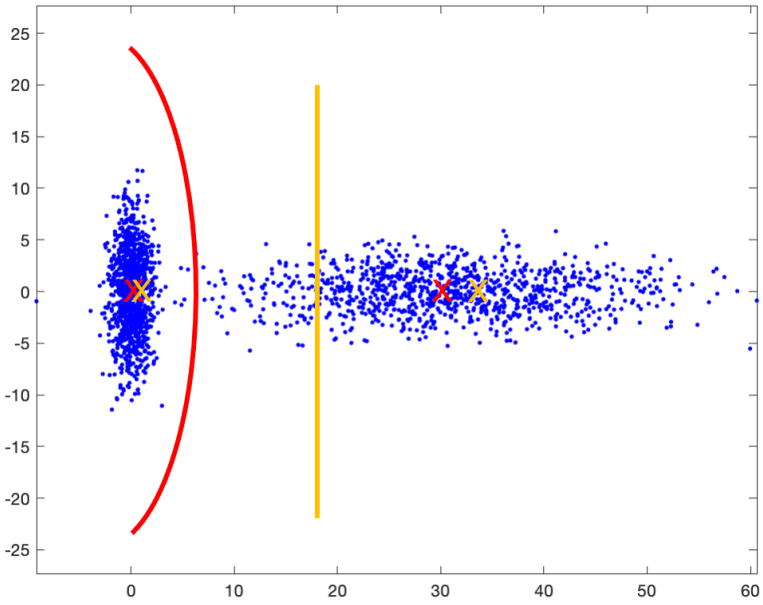


5. $f(s) = \frac{s}{\|s\|}$ has a zero derivative everywhere, except in zero where it is undefined. Therefore it is not useful in gradient search.
6. Residual networks
7. The *momentum* term.
8. Overfitting.
9. $k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} e^{-(x_i^2 + y_i^2)}$
10. For $k = 1$, x belongs to the "cricle" class and for $k = 3$, x belongs to the "squares".

11. For example this:

-1	-1	-1
2	2	2
-1	-1	-1

12. k-means in yesllow, MoG in red:



13. The ideal discriminant function is a discrete radial basis function with center at $\mathbf{k}$ and radius ρ. Another radial basis function, which is differentiable, is for example a Gaussian:

$$f(\mathbf{x}) = e^{-||\mathbf{x}-\mathbf{k}||^2/\rho}$$

The update function is

$$\Delta\mathbf{k} = -\frac{\partial f}{\partial \mathbf{k}} = -\frac{2(\mathbf{x}-\mathbf{k})}{\rho} e^{-||\mathbf{x}-\mathbf{k}||^2/\rho}$$

and

$$\Delta\rho = -\frac{\partial f}{\partial \rho} = -\frac{||\mathbf{x}-\mathbf{k}||^2}{\rho^2} e^{-||\mathbf{x}-\mathbf{k}||^2/\rho}$$

14. (a) can be used for image segmentation as it has the same size of the output layer as the input layer. (b) can be used for image classification as the output is of lower dimensionality than the input

15. The loss function in LDA is given by

$$\epsilon(\mathbf{w}) = \frac{\mathbf{w^T M w}}{\mathbf{w C_{tot} w}}$$

where $\mathbf{C}_{tot} = \mathbf{C}_1 + \mathbf{C}_2$.

Let us write

$$\mathbf{C}_1 = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}.$$

Then, due to the 90 degree rotation,

$$\mathbf{C}_2 = \begin{pmatrix} b & 0 \\ 0 & a \end{pmatrix}.$$

This means that

$$\mathbf{C}_{tot} = \mathbf{C}_1 + \mathbf{C}_2 = \begin{pmatrix} c & 0 \\ 0 & c \end{pmatrix}$$

where $c = a + b$, is the covariance matrix of an isotropic distribution!

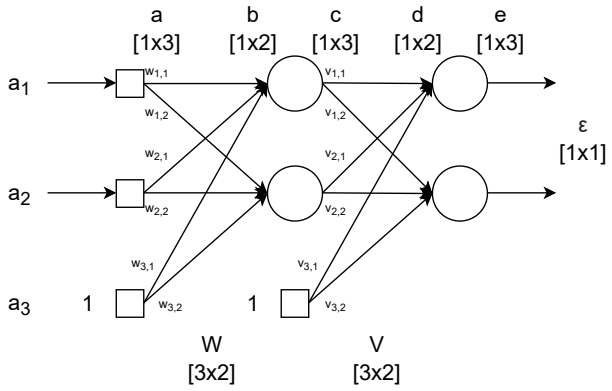


Figure 1: Neural network diagram.

Part 2

1. a) Figure 1 shows a labeled diagram of the network. We define the variables at each stage of the network:

$$a_i = i\text{-th input}, \quad b_j = \sum_{i=1}^{3} a_i w_{i,j},$$

$$c_j = \begin{cases} \max(0, b_j) & j = \{1, 2\}, \\ 1, & j = 3 \end{cases} \qquad d_k = \sum_{j=1}^{3} c_j v_{j,k},$$

$$e_k = d_k(\text{no activation}), \quad \epsilon = \sum_{k=1}^{2} (e_k - y_k)^2,$$

where y_k represents the target value for the k-th output.

b) From the above definitions, we calculate all partial derivatives between contiguous stages of the network:

$$\frac{\partial \epsilon}{\partial e_k} = \frac{\partial}{\partial e_k} \sum_{k=1}^{2} (e_k - y_k)^2 = 2(e_k - y_k),$$

$$\frac{\partial e_k}{\partial d_k} = \frac{\partial}{\partial d_k} d_k = 1,$$

$$\frac{\partial d_k}{\partial c_j} = \frac{\partial}{\partial c_j} \sum_{j=1}^{3} c_j v_{j,k} = v_{j,k}, \quad \frac{\partial d_k}{\partial v_{j,k}} = \frac{\partial}{\partial v_{j,k}} \sum_{j=1}^{3} c_j v_{j,k} = c_j,$$

$$\frac{\partial c_j}{\partial b_j} = \frac{\partial}{\partial b_j} \max(0, b_j) = \text{step}(b_j), \quad j = 1, 2$$

$$\frac{\partial b_j}{\partial a_i} = \frac{\partial}{\partial a_i} \sum_{i=1}^{3} a_i w_{i,j} = w_{i,j}, \quad \frac{\partial b_j}{\partial w_{i,j}} = \frac{\partial}{\partial w_{i,j}} \sum_{i=1}^{3} a_i w_{i,j} = a_i.$$

Using these, we can find the gradient of the loss with respect to the weight matrices:

$$\frac{\partial \epsilon}{\partial v_{j,k}} = \frac{\partial \epsilon}{\partial e_k} \frac{\partial e_k}{\partial d_k} \frac{\partial d_k}{\partial v_{j,k}} = 2(e_k - y_k) c_j,$$

$$\frac{\partial \epsilon}{\partial w_{i,j}} = \sum_{k=1}^{2} \frac{\partial \epsilon}{\partial e_k} \frac{\partial e_k}{\partial d_k} \frac{\partial d_k}{\partial c_j} \frac{\partial c_j}{\partial b_j} \frac{\partial b_j}{\partial w_{i,j}} = \sum_{k=1}^{2} 2(e_k - y_k) v_{j,k} \text{step}(b_j) a_i,$$

and the weight update rules are

$$v_{j,k} \leftarrow v_{j,k} - \eta \frac{\partial \epsilon}{\partial v_{j,k}},$$
$$w_{i,j} \leftarrow w_{i,j} - \eta \frac{\partial \epsilon}{\partial w_{i,j}}.$$

c) and d) We are asked to find update rules for the input that maximize specific outputs. We can directly reuse the calculations from b) to find this. To maximize the output of the hidden layer we have

$$\frac{\partial c_j}{\partial a_i} = \frac{\partial c_j}{\partial b_j}\frac{\partial b_j}{\partial a_i} = \text{step}(b_j) w_{i,j},$$
$$a_i \leftarrow a_i + \eta \frac{\partial c_j}{\partial a_i},$$

and to maximize the output we have

$$\frac{\partial e_k}{\partial a_i} = \frac{\partial e_k}{\partial d_k} \sum_{j=1}^{2} \frac{\partial d_k}{\partial c_j}\frac{\partial c_j}{\partial b_j}\frac{\partial b_j}{\partial a_i} = \sum_{j=1}^{2} v_{j,k} \text{step}(b_j) w_{i,j},$$
$$a_i \leftarrow a_i + \eta \frac{\partial e_k}{\partial a_i}.$$

Note that, since we are maximizing a function, we follow the positive direction of the gradient. Therefore, the gradient is added rather than subtracted from a_i. Note also that we need to sum over j because of the multiple ways in which each e_k depends on each a_i.

2. a) In Figure 2(a) the classification problem is sketched along with the initial weights. Figure 2(b) shows the results of the first AdaBoost iteration. After selecting the threshold yielding minimum error we get $\epsilon_1 = \frac{1}{6}$ and $\alpha_1 = \frac{1}{2}\ln\frac{1-\epsilon_1}{\epsilon_1} = \frac{1}{2}\ln\frac{5/6}{1/6} = \frac{\ln 5}{2}$. The weights of the correctly classified samples are multiplied by $e^{-\alpha_1} = \frac{1}{\sqrt{5}}$, while the weight of the single misclassified sample is multiplied by $e^{\alpha_1} = \sqrt{5}$. The sum of the weights is $\frac{5}{6\sqrt{5}} + \frac{\sqrt{5}}{6} = \frac{\sqrt{5}}{3}$, and after normalizing they take the values shown in Figure 2(b).

   b) Figure 2(c) shows the results of the second AdaBoost iteration. After selecting the threshold yielding minimum error we get $\epsilon_2 = 2\frac{1}{10} = \frac{1}{5}$ and $\alpha_2 = \frac{1}{2}\ln\frac{1-\epsilon_2}{\epsilon_2} = \frac{1}{2}\ln\frac{4/5}{1/5} = \frac{\ln 4}{2} = \ln 2$. The weights of the correctly classified samples are multiplied by $e^{-\alpha_2} = \frac{1}{2}$, while the weight of the two misclassified samples are multiplied by $e^{\alpha_2} = 2$. The sum of the weights is $\frac{2}{5} + \frac{1}{4} + \frac{3}{20} = \frac{4}{5}$, and after normalizing they take the values shown in Figure 2(c).

   c) The strong classifier results are shown in Figure 2(c). The strong classifier is defined as $sign\left(\sum_i \alpha_i h_i\right)$ where h_i are the classifications by the weak classifiers.

$$\begin{aligned}
\sum_i \alpha_i h_i &= \frac{\ln 5}{2} \times \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 \end{bmatrix} \\
&+ \ln 2 \times \begin{bmatrix} -1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{\ln 5}{2} - \ln 2 \\ \frac{\ln 5}{2} + \ln 2 \\ \frac{\ln 5}{2} + \ln 2 \\ \frac{\ln 5}{2} + \ln 2 \\ \frac{-\ln 5}{2} + \ln 2 \\ \frac{-\ln 5}{2} + \ln 2 \end{bmatrix}^T
\end{aligned}$$

   Since $\frac{\ln 5}{2} > \ln 2$ we get $sign\left(\sum_i \alpha_i h_i\right) = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 \end{bmatrix}$ as the final strong classification, which is shown in figure 2c (right). One sample is still misclassified.

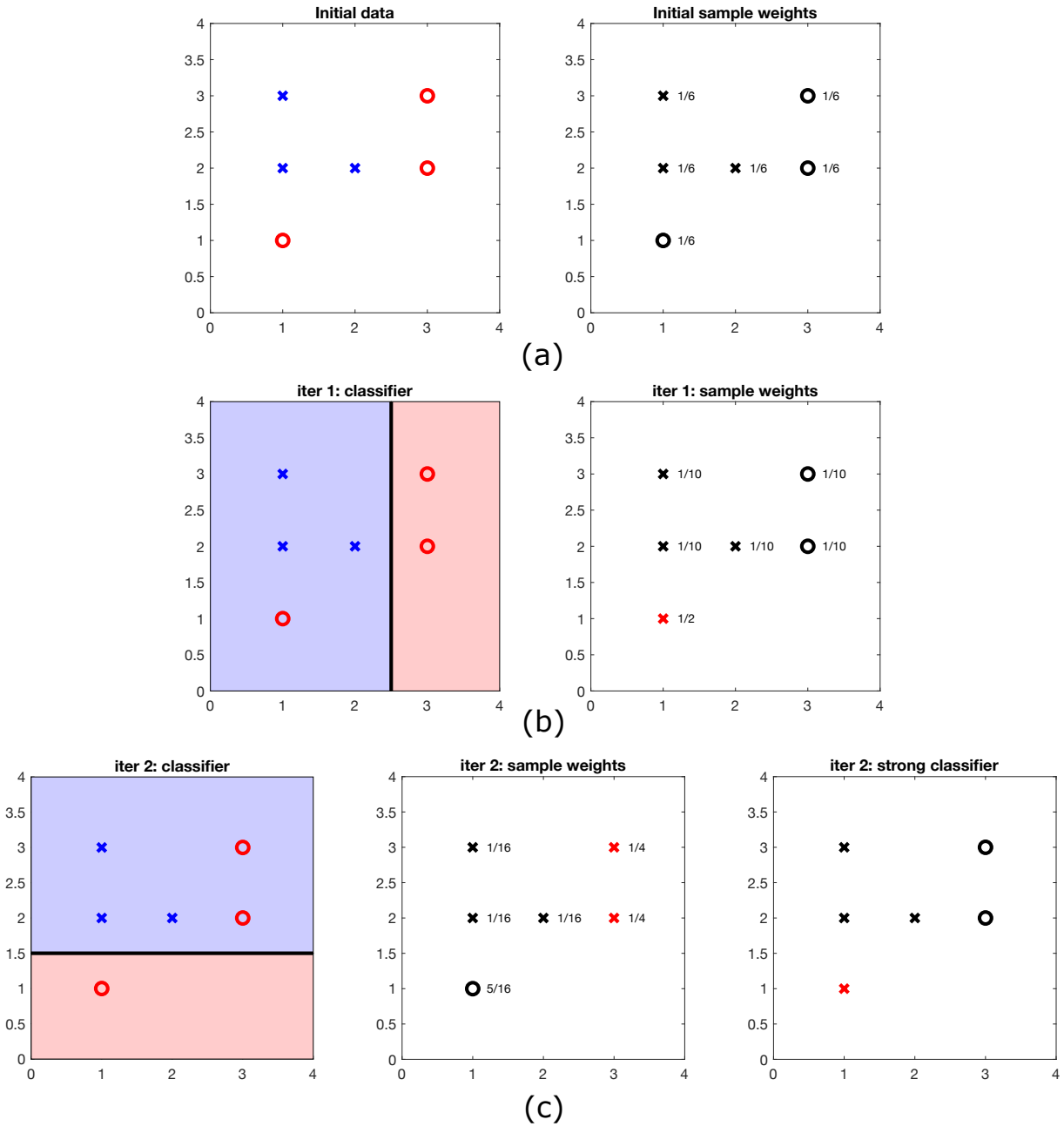


Figure 2: a) Initial state. Blue crosses mark the positive class and red circles mark the negative. b) Left: first threshold. Right: classification and updated weights. Black and red indicate correct and incorrect classifications, respectively. c) Left: second threshold. Middle: classification and updated weights. Black and red indicate correct and incorrect classifications, respectively. Right: strong classifier results. Black and red indicate correct and incorrect classifications, respectively.

3. a) The kernel function $K(\mathbf{x}, \mathbf{y})$ is the scalar product of the mapped feature vectors $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$:

$$\begin{aligned}
\Phi(\mathbf{x})^T\Phi(\mathbf{y}) &= \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ 1 \end{pmatrix}^T \begin{pmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2}y_1y_2 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ 1 \end{pmatrix} = \\
&= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 + 2x_1y_1 + 2x_2y_2 + 1 = \\
&= (x_1y_1 + x_2y_2 + 1)^2 = \\
&= (\mathbf{x}^T\mathbf{y} + 1)^2
\end{aligned}$$

b) The scalar product $\mathbf{x}^T\mathbf{y}$ in N dimensions has N terms, so $(\mathbf{x}^T\mathbf{y}+1)$ has $N+1$ terms. The square of $N+1$ terms yields $(N+1)^2$ terms, but due to symmetry $N*(N+1)/2$ terms are duplicates and thus results in no new dimensions in the feature space. One way to visualize this is to think of all the unique elements of a symmetric matrix of size $(N+1)\times(N+1)$, representing all products in a quadratic expansion of $N+1$ terms. We therefore get $(N+1)^2 - N(N+1)/2$ unique dimensions. Simplifying gives:

$$\begin{aligned}
\dim\left(K(\mathbf{x},\mathbf{y})\right) &= (N+1)^2 - \frac{N(N+1)}{2} \\
&= N^2 + 2N + 1 - \frac{N^2+N}{2} \\
&= \frac{N^2}{2} + \frac{3N}{2} + 1 \\
&= \frac{1}{2}(N^2 + 3N + 2) \\
&= \frac{1}{2}(N+1)(N+2) \\
&= \frac{(N+2)!}{2!\,N!} \\
&= \binom{N+2}{2}
\end{aligned}$$

The final simplification is "$N+2$ choose 2", i.e. a combinatorial operation. In general, a d-degree polynomial kernel $(\mathbf{x}^T\mathbf{y}+1)^d$ maps to "$N+d$ choose d" dimensions. Note that we do not require the full simplification.

4. The formula for the optimal Q-function is given by:

$$\begin{aligned} Q^*(s_t, a) &= r(s_t, a) + \gamma V^*(s_{t+1}) \\ &= r(s_t, a) + \gamma \max_b Q^*(s_{t+1}, b) \end{aligned}$$

where s_{t+1} is the state reached by performing action a in state s_t. $Q(s, a)$ can be updated using the following rule:

$$\begin{aligned} Q(s_t, a) &\leftarrow (1-\alpha)Q(s_t, a) + \alpha\left(r(s_t, a) + \gamma V(s_{t+1})\right) \\ &= (1-\alpha)Q(s_t, a) + \alpha\left(r(s_t, a) + \gamma \max_b Q(s_{t+1}, b)\right) \end{aligned}$$

where $\alpha \in (0, 1]$ is the learning rate and $\gamma \in (0, 1)$ is the discount factor.

a) Counting backwards we get:

$$\begin{aligned} V^*(6) &= 0 \\ V^*(5) = Q^*(5, right) &= 2 + \gamma V^*(6) = 2 \\ V^*(4) = Q^*(4, right) &= 1 + \gamma V^*(5) = 1 + 2\gamma \\ V^*(3) = Q^*(3, down) &= 1 + \gamma V^*(6) = 1 \\ Q^*(2, right) &= 1 + \gamma V^*(3) = 1 + \gamma \\ Q^*(2, down) &= 2 + \gamma V^*(5) = 2 + 2\gamma \\ V^*(2) = \max(Q^*(2, right), Q^*(2, down)) &= 2 + 2\gamma \\ Q^*(1, right) &= 2 + \gamma V^*(2) = 2 + 2\gamma + 2\gamma^2 \\ Q^*(1, down) &= 1 + \gamma V^*(4) = 1 + \gamma + 2\gamma^2 \\ V^*(1) = \max(Q^*(1, right), Q^*(1, down)) &= 2 + 2\gamma + 2\gamma^2 \end{aligned}$$

b) The specified sequence of actions results in the following updates to $Q(s, a)$, using the learning rate α and the discount factor γ.

$$\begin{aligned} Q(3, down) &= \alpha(1 + \gamma V(6)) = \alpha(1 + 0\gamma) &&= \alpha \\ Q(5, right) &= \alpha(2 + \gamma V(6)) = \alpha(2 + 0\gamma) &&= 2\alpha \\ Q(2, right) &= \alpha(1 + \gamma V(3)) = \alpha(1 + \gamma\alpha) &&= \alpha + \gamma\alpha^2 \\ Q(2, down) &= \alpha(2 + \gamma V(5)) = \alpha(2 + 2\gamma\alpha) &&= 2\alpha + 2\gamma\alpha^2 \\ Q(4, right) &= \alpha(1 + \gamma V(5)) = \alpha(1 + 2\gamma\alpha) &&= \alpha + 2\gamma\alpha^2 \\ Q(1, right) &= \alpha(2 + \gamma V(2)) = \alpha(2 + \gamma(2\alpha + 2\gamma\alpha^2)) &&= 2\alpha + 2\gamma\alpha^2 + 2\gamma^2\alpha^3 \\ Q(1, down) &= \alpha(1 + \gamma V(4)) = \alpha(1 + \gamma(\alpha + 2\gamma\alpha^2)) &&= \alpha + \gamma\alpha + 2\gamma^2\alpha^3 \end{aligned}$$