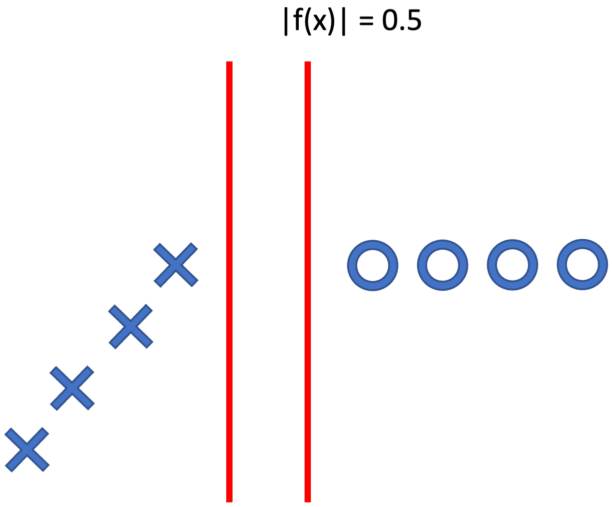
Solutions to the exam in
Neural Networks and Learning Systems - TBMI26 / 732A55
Exam 2022-06-09

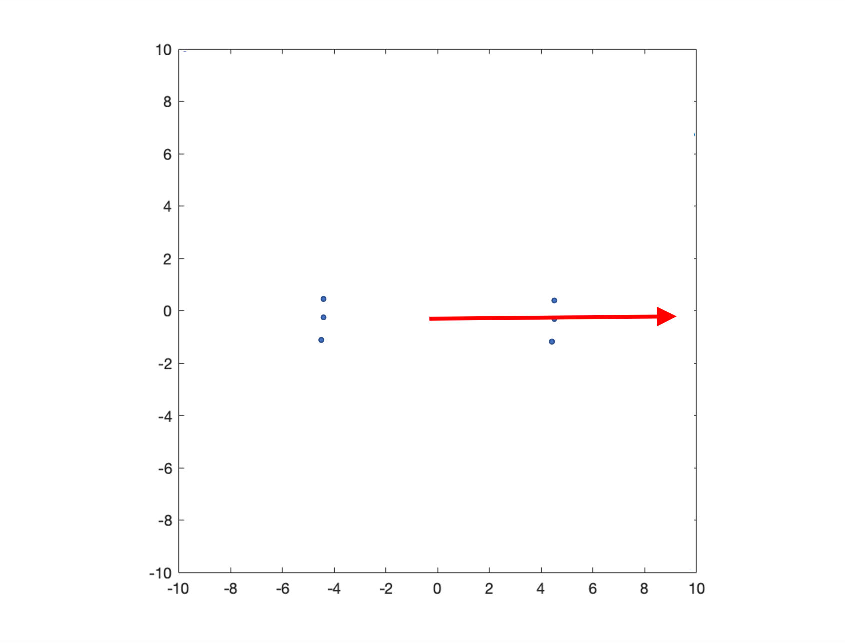
Part 1

1. - Q-leaerning - Reinforcement learning
   - k-means - Unsupervised learning
   - kNN (k nearest neighbors) - Supervised learning
2. See figure:



3. For example $\mathbf{w} = \begin{pmatrix} -2 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$, where -3 is the bias weight.

4. See figure. (The opposite direction is also correct.)



5. The horizontal "concatenation connections" (shortcuts) are missing.

6. Wee need three mean vectors (each 2 parameters) and three covariance matrices (4 parameters each) which gives $6 + 12 = 18$ parameters in total. (Actually, since the covariance matrices are symmetric we only need to estimate 3 parameters for each covariance matrix, which gives 15 in total.)

7. Pooling.

8. The general definition of a kernel function is $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})$, where $\phi((\mathbf{x})$ is a non-linear function.

9. Beacause this leads to over-fitting to the training data and reduced performance on new data.

10. The value of the V function for is the maximum Q-value over all possible actions.

11. This illustrates the ”vanishing gradient” problem refers to problem of propagating the error back in deep neural networks when using sigmoid activation functions, which has a small derivative almost everywhere. To avoid this, we can us alternative activation functions such as the ReLU function.

12. In image segmentation, the size of the output layer is the same as the size of the input layer, since the output is an image. In image classification, the output layer is typically much smaller, with one output for each class.

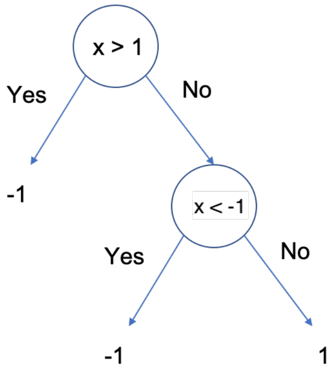
13. A kernel function defines the inner product $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1)^T\Phi(\mathbf{x}_2)$ in the new feature space. Thus, the distance between $\mathbf{x}_1$ and $\mathbf{x}_2$ in the new feature space is

$$\begin{aligned}\|\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2)\| &= \sqrt{(\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2))^T\,(\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2))} \\ &= \sqrt{\Phi(\mathbf{x}_1)^T\Phi(\mathbf{x}_1) - 2\Phi(\mathbf{x}_1)^T\Phi(\mathbf{x}_2) + \Phi(\mathbf{x}_2)^T\Phi(\mathbf{x}_2)} \\ &= \sqrt{\kappa(\mathbf{x}_1, \mathbf{x}_1) - 2\,\kappa(\mathbf{x}_1, \mathbf{x}_2) + \kappa(\mathbf{x}_2, \mathbf{x}_2)} \\ &= \sqrt{1 - 2/9 + 1} = 4/3\end{aligned}$$

14. For example this:

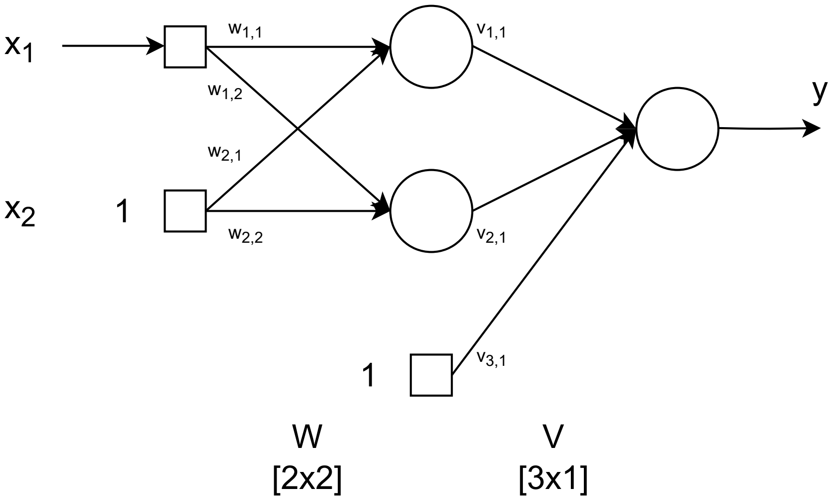
-1	-1	-1
-1	8	-1
-1	-1	-1

15.



Part 2

1. a) One possible solution is using the network illustrated below.



where

$$\mathbf{W} = \begin{bmatrix} -1 & 1 \\ -0.2 & -1.1 \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

After forward propagation through the network with *sign* as activation function in the hidden layer and in the output layer, $\mathbf{Y}$ will be:

$$\mathbf{y} = \begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \end{bmatrix}^T.$$

$\mathbf{y} = \mathbf{d}$, giving an accuracy of 100%.

b) The weights need to be initialized randomly. The data is forward propagated and the error (using a suitable error function) is calculated. The minimum error is found using gradient search:$\frac{\delta\epsilon}{\delta\mathbf{V}}$ and $\frac{\delta\epsilon}{\delta\mathbf{W}}$. The weights are after that updated using gradient descent: $\mathbf{V}_{i+1} = \mathbf{V}_i - \eta\frac{\delta\epsilon}{\delta\mathbf{V}_i}$ and $\mathbf{W}_{i+1} = \mathbf{W}_i - \eta\frac{\delta\epsilon}{\delta\mathbf{W}_i}$. *sign* is not differentiable so it must be changed to another suitable activation function, e.g. $tanh()$.

2. We have the following data:

$$\mathbf{X} = \begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix}$$

a) The corresponding initial weights, d_1 are;

$$\mathbf{d}_1 = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}$$

After performing brute force optmization over the three features we get:

$$\epsilon_1 = \frac{1}{6}$$

$$\alpha_1 = \frac{1}{2}\ln(\frac{1-\epsilon}{\epsilon}) = \frac{1}{2}\ln(\sqrt{5})$$

We update the correctly classified samples with $\exp(-\alpha)$ and the wrongly classified samples with $\exp(\alpha)$.

$$\mathbf{d}_2 = \frac{1}{6}\begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \sqrt{5} & \frac{1}{\sqrt{5}} \end{bmatrix}$$

Normalize $\mathbf{d}_2$:

$$\sum(\mathbf{d}_2) = \frac{1}{6}(\sqrt{5} + \frac{5}{\sqrt{5}}) = \frac{\sqrt{5}}{3}$$

$$\mathbf{d}_2 = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{2} & \frac{1}{10} \end{bmatrix}$$

b) The training accuracy will never reach 100% since the second and fifth samples have the same coordinates but different class.

3. The direction that optimally separates the two classes is given by

$$\mathbf{w} = \mathbf{C}_{\text{tot}}^{-1}\left(\mathbf{m}_{\text{x}} - \mathbf{m}_{\text{o}}\right)$$

where $\mathbf{C}_{\text{tot}}$ is the sum of the individual covariance matrices for the two respective classes, and $\mathbf{m}_{\text{x}}$ and $\mathbf{m}_{\text{o}}$ are the centers of the two classes. We begin with the "crosses" class:

$$\mathbf{X}_{\text{x}} = \begin{pmatrix} 1 & 2 & 2 & 2 & 2 & 3 \\ 1 & 0 & 2 & 3 & -1 & 1 \end{pmatrix}$$

$$\mathbf{C}_{\text{x}} = \frac{1}{N_{\text{x}} - 1}\left(\mathbf{X}_{\text{x}} - \mathbf{m}_{\text{x}}\right)\left(\mathbf{X}_{\text{x}} - \mathbf{m}_{\text{x}}\right)^T = \left[\mathbf{m}_{\text{x}} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}\right] = \frac{2}{5}\begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}$$

Now for the "circles" class:

$$\mathbf{X}_{\text{o}} = \begin{pmatrix} -2 & -2 & -2 & 0 & 0 & 0 \\ -3 & -1 & 2 & -3 & 0 & -1 \end{pmatrix}$$

$$\mathbf{C}_{\text{o}} = \frac{1}{N_{\text{o}} - 1}\left(\mathbf{X}_{\text{o}} - \mathbf{m}_{\text{o}}\right)\left(\mathbf{X}_{\text{o}} - \mathbf{m}_{\text{o}}\right)^T = \left[\mathbf{m}_{\text{o}} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}\right] = \frac{2}{5}\begin{pmatrix} 3 & -1 \\ -1 & 9 \end{pmatrix}$$

We now get:

$$\mathbf{C}_{\text{tot}} = \mathbf{C}_{\text{x}} + \mathbf{C}_{\text{o}} = \frac{2}{5}\begin{pmatrix} 4 & -1 \\ -1 & 14 \end{pmatrix}$$

$$\mathbf{C}_{\text{tot}}^{-1} = \frac{1}{\left(\frac{2}{5}\right)^2 (4 * 14 - 1 * 1)} * \frac{2}{5}\begin{pmatrix} 14 & 1 \\ 1 & 4 \end{pmatrix} = \frac{1}{22}\begin{pmatrix} 14 & 1 \\ 1 & 4 \end{pmatrix}$$

$$\mathbf{w} = \mathbf{C}_{\text{tot}}^{-1}\left(\mathbf{m}_{\text{x}} - \mathbf{m}_{\text{o}}\right) = \frac{1}{2}\begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

Normalize $\mathbf{w}$:

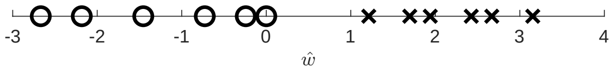
$$\hat{\mathbf{w}} = \frac{\mathbf{w}}{||\mathbf{w}||_2} = \frac{1}{\sqrt{17}}\begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

Project the data on $\hat{\mathbf{w}}$ which gives the following:

$$\mathbf{Y}_{\text{x}} = \hat{\mathbf{w}}^T\mathbf{X}_{\text{x}} = \frac{1}{\sqrt{17}}\begin{pmatrix} 5 & 8 & 10 & 11 & 7 & 13 \end{pmatrix}$$

$$\mathbf{Y}_{\text{o}} = \hat{\mathbf{w}}^T\mathbf{X}_{\text{o}} = \frac{1}{\sqrt{17}}\begin{pmatrix} -11 & -9 & -6 & -3 & 0 & -1 \end{pmatrix}$$

The projected data is shown in the figure below.



4. The update of Q-value in state S_k and action a_j is given by:

$$\hat{Q}(S_k, a_j) \quad \leftarrow \quad (1-\alpha)\hat{Q}(S_k, a_j) + \alpha(r + \gamma \max_a \hat{Q}(S_{k+1}, a))$$

We can update the values of the Q-function by following the sequence of actions. Starting with *Sequence 1*:

$$\begin{aligned}\hat{Q}(1, down) &= (1-\alpha)\hat{Q}(1, down) + \alpha(r + \gamma \max_a \hat{Q}(3)) \\ &= (1-\alpha)0 + \alpha(0 + \gamma 0) = 0\end{aligned}$$

$$\begin{aligned}\hat{Q}(3, down) &= (1-\alpha)\hat{Q}(3, down) + \alpha(r + \gamma \max_a \hat{Q}(5)) \\ &= (1-\alpha)0 + \alpha(-5 + \gamma 0) = -5\ \alpha\end{aligned}$$

$$\begin{aligned}\hat{Q}(5, down) &= (1-\alpha)\hat{Q}(5, down) + \alpha(r + \gamma \max_a \hat{Q}(7)) \\ &= (1-\alpha)0 + \alpha(0 + \gamma 0) = 0\end{aligned}$$

$$\begin{aligned}\hat{Q}(7, left) &= (1-\alpha)\hat{Q}(7, left) + \alpha(r + \gamma \max_a \hat{Q}(6)) \\ &= (1-\alpha)0 + \alpha(5 + \gamma 0) = 5\alpha\end{aligned}$$

Following now *Sequence 2* of actions:

$$\begin{aligned}\hat{Q}(1, down) &= (1-\alpha)\hat{Q}(1, down) + \alpha(r + \gamma \max_a \hat{Q}(3)) \\ &= (1-\alpha)0 + \alpha(0 + \gamma \max(0, -5\alpha)) = 0\end{aligned}$$

$$\begin{aligned}\hat{Q}(3, left) &= (1-\alpha)\hat{Q}(3, left) + \alpha(r + \gamma \max_a \hat{Q}(2)) \\ &= (1-\alpha)0 + \alpha(-8 + \gamma 0) = -8\alpha\end{aligned}$$

$$\begin{aligned}\hat{Q}(2, down) &= (1-\alpha)\hat{Q}(2, down) + \alpha(r + \gamma \max_a \hat{Q}(4)) \\ &= (1-\alpha)0 + \alpha(4 + \gamma 0) = 4\alpha\end{aligned}$$

$$\begin{aligned}\hat{Q}(4, down) &= (1-\alpha)\hat{Q}(4, down) + \alpha(r + \gamma \max_a \hat{Q}(6)) \\ &= (1-\alpha)0 + \alpha(5 + \gamma 0) = 5\alpha\end{aligned}$$

Following *Sequence 3* of actions:

$$\begin{aligned}\hat{Q}(1, down) &= (1-\alpha)\hat{Q}(1, down) + \alpha(r + \gamma \max_a \hat{Q}(3)) \\ &= (1-\alpha)0 + \alpha(0 + \gamma \max(-8\alpha, -5\alpha)) \\ &= -5\gamma\alpha^2\end{aligned}$$

$$\begin{aligned}
\hat{Q}(3, down) &= (1-\alpha)\hat{Q}(3, down) + \alpha(r + \gamma \max_a \hat{Q}(5)) \\
&= (1-\alpha)(-5\alpha) + \alpha(-5 + \gamma \max(0, -5\alpha)) \\
&= -10\alpha + 5\alpha^2
\end{aligned}$$

$$\begin{aligned}
\hat{Q}(5, left) &= (1-\alpha)\hat{Q}(5, left) + \alpha(r + \gamma \max_a \hat{Q}(4)) \\
&= (1-\alpha)0 + \alpha(4 + \gamma \max(0, 5\alpha)) \\
&= 4\alpha + 5\gamma\alpha^2
\end{aligned}$$

$$\begin{aligned}
\hat{Q}(4, down) &= (1-\alpha)\hat{Q}(4, down) + \alpha(r + \gamma \max_a \hat{Q}(6)) \\
&= (1-\alpha)(5\alpha) + \alpha(5 + \gamma 0) \\
&= 10\alpha - 5\alpha^2
\end{aligned}$$