Solutions to the exam in
Neural Networks and Learning Systems - TBMI26
Exam 2018-03-12

Part 1

1. - Q-leaerning - Reinforcement learning
   - k-means - Unsupervised learning
   - kNN (k nearest neighbors) - Supervised learning
2. Support vector machines (SVM).
3. The requirement is that it is differentiable.
4. PCA
5. The exploration-exploitation dilemma refers to the conflict between utilizing the current knowledge to maximize the reward and trying new policies to explore new potentially better solutions.
6. 10 × 10.
7. The quotient between the difference between the cluster means and the variance of the projections of the clusters.
8. For example, using a momentum term.
9. It will improve the accuracy when the distribution of the classes are very different, since the boundary will be in the middle between the class centroids. (Slide 16 on lecture 3).
10. One benefit is that there are fewer parameters to train, which reduces the risk of over-fitting and improves the generalization properties.

Part 2

11. K-means and Mixture of Gaussians are two examples of the expectation maximization (EM) algorithm. The difference to gradient search is that there are latent variables (e.g. class labels) that are discrete and the algoritm iterates between estimating the model parameters (e.g. mean and/or covariance matrices for the class distributions), and estimating the latent variables (class labels).

12. The data is divided into a training set, a validation set and test set. the training set is used for updating the parameters. The validation set is used for checking the generalization error during training. The test set is used after trining for evaluating the final performance.

13. The original cost function is:

$$\min \|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$$
$$\text{subject to the constraint } y_i\left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \geq 1 \text{ for all } i.$$

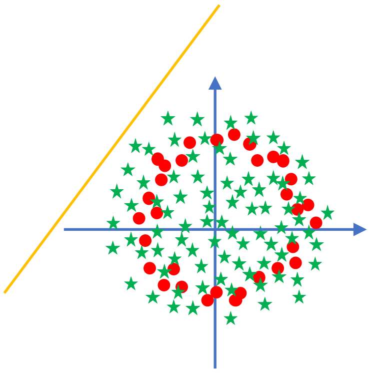
The dual cost function can be derived for problems for which it can be shown that the optimal solution $\mathbf{w}^*$ lies in the span of the input training data, i.e., $\mathbf{w}^* = \sum_{i=1}^N \alpha_i \mathbf{x}_i$, where α_i are some number, $\mathbf{x}_i$ is a traing data example and N is the number of training examples. Inserting this relationship in the original cost function yields $\mathbf{w}^T\mathbf{w} = \sum_{i=1}^N \sum_{j=1}^N \alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j$ (which can also be written in a vector form $\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}$ using a kernel matrix $\mathbf{K}$). The entire dual cost function is

$$\min \sum_{i=1}^N \sum_{j=1}^N \alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j$$

$$\text{subject to the constraint } y_i\left(\sum_{j=1}^N \alpha_j\mathbf{x}_j^T\mathbf{x}_i + \alpha_0\right) \geq 1 \text{ for all } i.$$

In non-linear kernel methods, the inner products $\mathbf{x}_i^T\mathbf{x}_j$ is replaced with a non-linear kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$.

14. Obviously, there is no good linear solution for this problem, but due to the imbalanced data, any discriminant function that classifies all sampels as healthy would have an accuracy of 68 %!



15. The sigmoid function leads to vanishing gradients. If the activation is large at one layer, gradients become small. Since the gradients from all layers are multiplied in the chain rule, deep networks suffer exponentially from this effect. This problem can be avoided by using a ReLU activation function instead of the sigmoid.

Part 3

16. a)

C =

3	2	4	2
4	9	2	4
6	4	6	0

b)

AA =

0	0	0	0	0	0
0	**0**	1	0	1	0
0	2	0	1	0	0
0	0	2	0	0	0
0	0	0	0	0	0

c) A filter at the n-th layer, is $3 * 2^{n-1}$ cells wide in the original image.

17. a) In figure 1a the classification problem is sketched along with the initial weights. In figure 1b the best threshold (yielding minimum error) is drawn. We get $\alpha_1 = \frac{1}{2}\ln\frac{1-\epsilon_1}{\epsilon_1} = \frac{1}{2}\ln\frac{5/6}{1/6} = \frac{\ln 5}{2}$. Now we update (decreasing) the weights of the correctly classified samples with $e^{-\alpha_1} = \frac{1}{\sqrt{5}}$. The weight of the erroneous classified sample (upper left in the solution example) is instead increased with $e^{\alpha_1} = \sqrt{5}$. The sum of the weights are now $\frac{5}{6\sqrt{5}} + \frac{\sqrt{5}}{6} = \frac{\sqrt{5}}{3}$. After normalizing the weights (dividing with the sum) we get the resulting weights as shown in figure 1c.

b) In figure 1d the classification problem is sketched with the new best threshold. The smallest error possible give the error $\epsilon = \frac{2}{10} = \frac{1}{5}$. We get $\alpha_2 = \frac{\ln 4}{2} = \ln 2$. Now we update (decreasing) the weights of the correctly classified samples with $e^{-\alpha_1} = \frac{1}{2}$. The weight of the erroneous classified sample (upper left in the solution example) is instead increased with $e^{\alpha_1} = 2$. The sum of the weights are now $\frac{2}{5} + \frac{1}{4} + \frac{3}{20} = \frac{4}{5}$. After normalizing the weights (dividing with the sum) we get the resulting weights as shown in figure 1e.

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. We get

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

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 1f. One sample is still misclassified, but will be correctly classified after one more iteration.

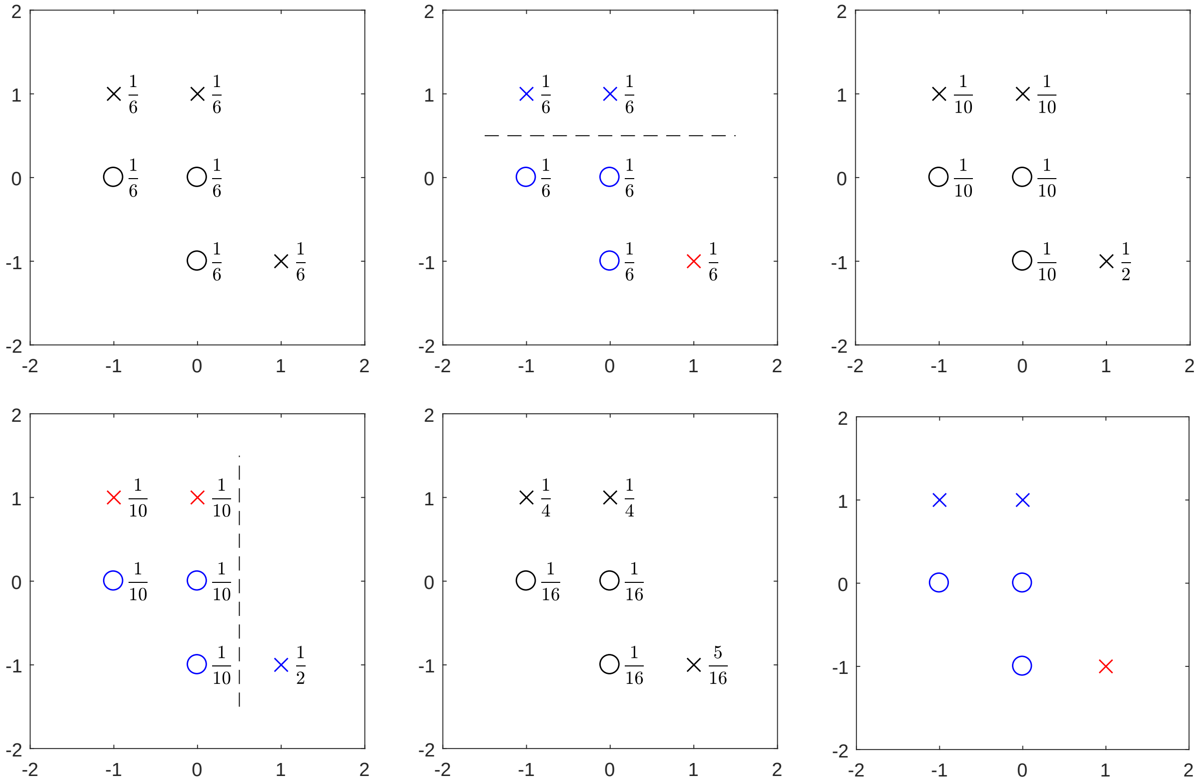
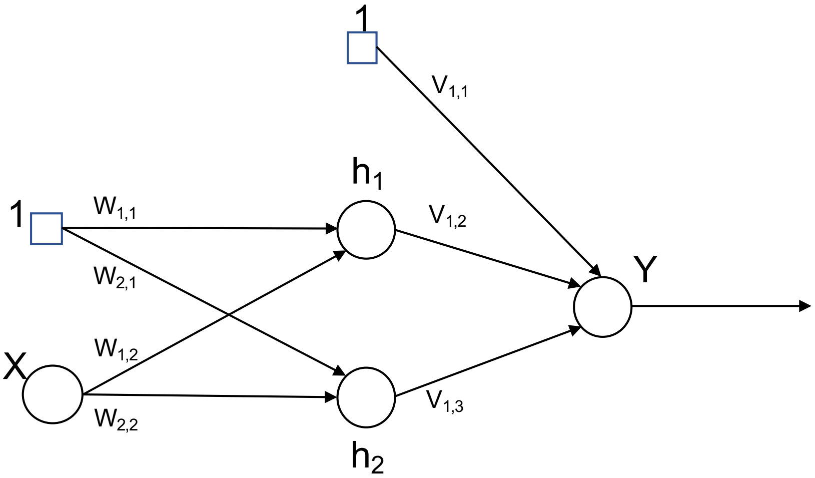


Figure 1: a) The initial state. Crosses marks the positive class, circles the negative. b) The first threshold, and the resulting classifications. Blue indicates correct classifications, and red indicates incorrect classifications. c) Weights after the first AdaBoost iteration. d) Second threshold and resulting classifications. e) Weights after the second AdaBoost iteration. f) Final strong classification. One sample is misclassified.

18. a) One possible solution is using the network illutrated below.



where

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 0.5 & -1 \end{bmatrix}$$
$$\mathbf{V} = \begin{bmatrix} -1 & 1 & 1 \end{bmatrix}$$

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

$$\mathbf{Y} = \begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix}$$

$\mathbf{Y} = \mathbf{D}$, giving an accuracy of 100 %.

b) The weights need to be initialized randomly. The net 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: $\mathbf{V_{i+1}} = \mathbf{V_i} - \eta\frac{\delta\epsilon}{\delta\mathbf{V}}$ and $\mathbf{W_{i+1}} = \mathbf{W_i} - \eta\frac{\delta\epsilon}{\delta\mathbf{W}}$. *sign* is not differentiable so iy must be changed to another suitable activation function, e.g. $tanh()$.

19. A definition of an optimal (deterministic) Q-function:

$$\begin{aligned} Q^*(x,a) &= r(x,a) + \gamma V^*(f(x,a)) && (1)\\ &= r(x,a) + \gamma Q^*(f(x,a), \mu^*(f(x,a))) && (2)\\ &= r(x,a) + \gamma \max_b Q^*(f(x,a), b) && (3) \end{aligned}$$

and for the V-function:

$$V^*(x) = \max_b Q^*(x,b) \qquad (4)$$

We can find a solution by going through the state models recursively.

System A:

$$\begin{aligned} V^*(3) &= 0 && (5)\\ V^*(4) = Q^*(4, \mathit{left}) &= 3 + \gamma V^*(3) = 3 && (6)\\ Q^*(2, \mathit{up}) &= 1 + \gamma V^*(3) = 1 && (7)\\ Q^*(2, \mathit{right}) &= 0 + \gamma V^*(4) = 3\gamma && (8)\\ V^*(2) &= \max(1, 3\gamma) = \begin{cases} 1 & \text{if } \gamma \le 1/3 \\ 3\gamma & \text{if } \gamma > 1/3 \end{cases} && (9)\\ V^*(1) = Q^*(1, \mathit{up}) &= 0 + \gamma V^*(2) = \begin{cases} \gamma & \text{if } \gamma \le 1/3 \\ 3\gamma^2 & \text{if } \gamma > 1/3 \end{cases} && (10) \end{aligned}$$

System B:

$$\begin{aligned} V^*(4) &= 0 && (11)\\ Q^*(3, \mathit{up}) &= 1 + \gamma V^*(4) = 1 && (12)\\ Q^*(3, \mathit{down}) &= 0 + \gamma V^*(2) = \gamma V^*(2) && (13)\\ V^*(2) = Q^*(2, \mathit{up}) &= 2 + \gamma V^*(3) && (14)\\ V^*(1) = Q^*(1, \mathit{up}) &= 0 + \gamma V^*(2) = \gamma V^*(2) && (15) \end{aligned}$$

Assume that $Q^*(3, \mathit{up})$ is larger than $Q^*(3, \mathit{down})$. Then we get

$$V^*(3) = Q^*(3, \mathit{up}) = 1 \qquad (16)$$

If we instead assume that $Q^*(3, \mathit{down})$ is larger than $Q^*(3, \mathit{up})$, then we get

$$\begin{aligned} V^*(3) &= Q^*(3, \mathit{down}) = \gamma V^*(2) = \gamma\,(2 + \gamma V^*(3)) \implies && (17)\\ V^*(3) &= \frac{2\gamma}{1-\gamma^2} && (18) \end{aligned}$$

Since the optimal policy is followed, we go *up* when

$$\begin{aligned} \frac{2\gamma}{1-\gamma^2} &\le 1 \implies && (19)\\ \gamma^2 + 2\gamma - 1 &\le 0 \implies && (20)\\ \gamma &\le \sqrt{2} - 1 \approx 0.414 && (21) \end{aligned}$$

With this and Eq. (14) and (15) we can write

$$V^*(3) = \begin{cases} 1 & \text{if } \gamma \leq \sqrt{2} - 1 \\ \dfrac{2\gamma}{1 - \gamma^2} & \text{if } \gamma > \sqrt{2} - 1 \end{cases} \tag{22}$$

$$V^*(2) = \begin{cases} 2 + \gamma & \text{if } \gamma \leq \sqrt{2} - 1 \\ \dfrac{2}{1 - \gamma^2} & \text{if } \gamma > \sqrt{2} - 1 \end{cases} \tag{23}$$

$$V^*(1) = \begin{cases} 2\gamma + \gamma^2 & \text{if } \gamma \leq \sqrt{2} - 1 \\ \dfrac{2\gamma}{1 - \gamma^2} & \text{if } \gamma > \sqrt{2} - 1 \end{cases} \tag{24}$$