Solutions to the exam in Neural Networks and Learning Systems - TBMI26 / 732A55 Exam 2020-06-10

Part 1

1. The following methods are supervised learning methods:

   - Back-propagation
   - k-NN
   - LDA
   - SVM

2. The cost function for SVM is

   $$\min \|\mathbf{w}\|^2$$

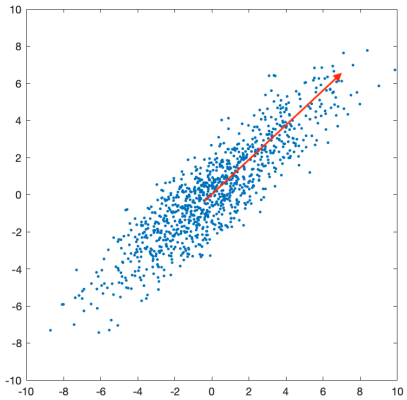
   under the constraint

   $$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \leq 1$$

3. The following functions can be used in the hidden layers of a back-prop network?

   - $y = \frac{1}{e^{-s}+1}$ (A sigmoid function)
   - $y = \begin{cases} 0, \text{for } s \leq 0 \\ s, \text{for } s > 0 \end{cases}$ (The Re-LU function)
   - $y = e^{(-s^2)}$ (A Gaussian radial basis function)

4. The first principal component:



5. There are two parameters to train. One input weight and one bias weight.

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. The rectified linear unit (ReLU) is used to avoid the vanishing gradient problem in deep multi-layer neural networks.

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

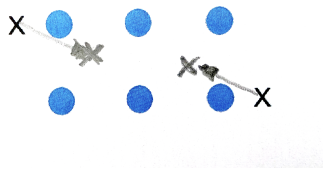
9. k is the number of stored data vectors that vote for the decision.

10. In order to have the possibility to find an even better policy. (The Bias-variance dilemma)

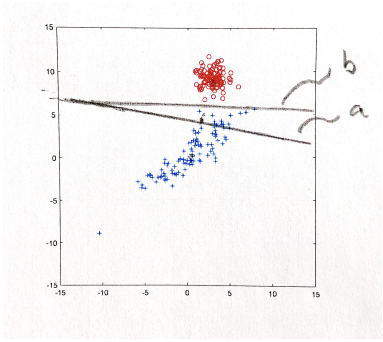
11.

$$\varphi(\mathbf{x})^T\varphi(\mathbf{y}) = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 = (x_1y_1 + x_2y_2)^2 = \left(\mathbf{x^T y}\right)^2 \tag{1}$$

12. The algorithm has converged already after one step:



13. See figure:



14. One possible implementation of the decision tree:

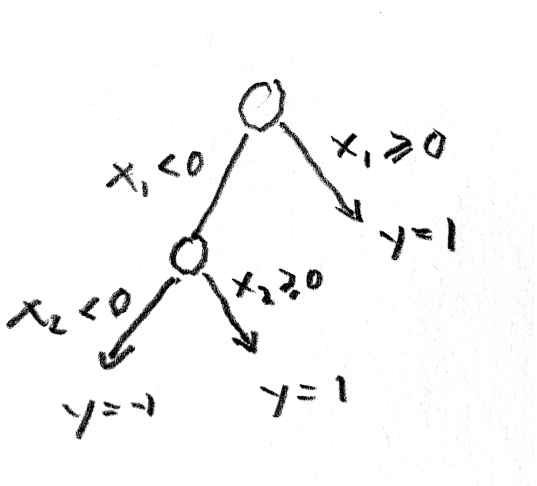


Figure 1: Decision tree

15. The values are $V(S_1) = 4$, $V(S_2) = 4$, $V(S_3) = 3$, and $V(S_4) = 2$. The system will take action A_1.

Part 2

1.

a) The update should be as

$$\Delta w_{ji} = -\eta \frac{\partial \varepsilon}{\partial w_{ji}}$$

where ε is defined as

$$\varepsilon = \frac{1}{2}\sum_k e_k^2 = \frac{1}{2}\sum_k (d_k - y_k)^2$$

where k is the number of output neurons.

For the output layer we have

$$\frac{\partial \varepsilon}{\partial w_{ji}} = \frac{\partial \varepsilon}{\partial e_j}\frac{\partial e_j}{\partial y_j}\frac{\partial y_j}{\partial w_{ji}} = -e_j v_i$$

and for the hidden layer we have

$$\frac{\partial \varepsilon}{\partial u_{ji}} = \sum_k \frac{\partial \varepsilon}{\partial e_k}\frac{\partial e_k}{\partial y_k}\frac{\partial y_k}{\partial v_j}\frac{\partial v_j}{\partial s_j}\frac{\partial s_j}{\partial u_{ji}} = -\sum_k e_k w_{kj} \acute{\sigma}\left(s_j\right) x_i$$

So the update of the output layer becomes:

$$\Delta w_{ji} = \eta e_j v_i$$

and for the hidden layer:

$$\Delta u_{ji} = \eta \sum_k e_k w_{kj} \acute{\sigma}\left(s_j\right) x_i$$

b) Without the activation function the two layers could be combined to a single layer. In practice we would have a single layer network and therefore only capable of linear decision boundaries.

c) While training, the output data are calculated for a set of training samples with known correct classifications. The output of the network is compared to the known correct output and the weights of the network are updated according to the expressions above. While classifying, the output data are calculated for one or several unknown samples. The class of the unknown sample is decided by the output neuron emitting the largest value.

2. a)

$$
\begin{array}{rcl}
g_{00} & = & \begin{array}{|c|c|c|c|}\hline \mathbf{1} & 2 & 2 & 2 \\ \hline \end{array} \\
g_{01} & = & \begin{array}{|c|c|c|c|}\hline \mathbf{3} & 2 & 3 & 2 \\ \hline \end{array} \\
g_{10} & = & \begin{array}{|c|c|c|c|}\hline \mathbf{2} & 2 & 1 & 2 \\ \hline \end{array} \\
g_{11} & = & \begin{array}{|c|c|c|c|}\hline \mathbf{0} & 3 & 0 & 1 \\ \hline \end{array}
\end{array}
$$

b) 90 parameters.

c) One such kernel is:

1	2	1
0	0	0
−1	2	−1

3. We have the following data:

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

a) The corresponding initial weights, d_1 are;

$$d_1 = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix}$$

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 wronfly classified samples with $\exp(\alpha)$.

$$d_2 = \frac{1}{6}\begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \sqrt{5} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix}$$

Normalize d_2:

$$\sum(d_2) = \frac{1}{6}(\sqrt{5} + \frac{5}{\sqrt{5}}) = \frac{\sqrt{5}}{3}$$

$$d_2 = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} & \frac{1}{2} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \end{bmatrix}$$

b) The training accuracy will never reach 100% since the third and sixth sample has the same coordinates but different class.

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

$$\begin{aligned} Q^*(x,a) &= r(x,a) + \gamma V^*(f(x,a)) && (2)\\ &= r(x,a) + \gamma Q^*(f(x,a), \mu^*(f(x,a))) && (3)\\ &= r(x,a) + \gamma \max_b Q^*(f(x,a),b) && (4) \end{aligned}$$

and for the V-function:

$$V^*(x) = \max_b Q^*(x,b) \quad (5)$$

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

Exercise A) According to the state model we have :

$$\begin{aligned} V^*(3) &= 1 + \gamma V^*(2) && (6)\\ V^*(2) &= 2 + \gamma V^*(3) && (7)\\ V^*(1) &= 1 + \gamma V^*(2) && (8)\\ V^*(3) &= \frac{1+2\gamma}{1-\gamma^2} = Q^*(3, down) && (9)\\ V^*(2) &= \frac{2+\gamma}{1-\gamma^2} = Q^*(2, up) && (10)\\ V^*(1) &= \frac{1+2\gamma}{1-\gamma^2} = Q^*(1, up) && (11)\\ & && (12) \end{aligned}$$

Exercise B)

$$\begin{aligned} V^*(3) &= 0 && (13)\\ V^*(4) &= 3 + \gamma V^*(3) = 3 = Q^*(4, left) && (14)\\ Q^*(2, up) &= 2 + \gamma V^*(3) = 2 && (15)\\ Q^*(2, right) &= 1 + \gamma V^*(4) = 1 + 3\gamma && (16)\\ V^*(2) &= \max(2, 1+3\gamma) = \begin{cases} 2 & \text{if } \gamma \leq \frac{1}{3} \\ 1+3\gamma & \text{if } \gamma > \frac{1}{3} \end{cases} && (17)\\ V^*(1) &= Q^*(1, up) = 1 + \gamma V^*(2) = \begin{cases} 1+2\gamma & \text{if } \gamma \leq \frac{1}{3} \\ 1+\gamma+3\gamma^2 & \text{if } \gamma > \frac{1}{3} \end{cases} && (18) \end{aligned}$$