# Exam in
# Neural Networks and Learning Systems
# - TBMI26

| | |
|---|---|
| Time: | 2013-03-14, 14-18 |
| Teacher: | Anette Karlsson, Phone: 0731807722 |
| Allowed additional material: | Calculator, Tefyma, Beta, Physics handbook, English dictionary |

The exam consists of three parts:

Part 1 Consists of ten questions. The questions test general knowledge and understanding of central concepts in the course. The answers should be short and given on the blank space after each question. Any calculations does **not** have to be presented. Maximum one point per question.

Part 2 Consists of five questions. These questions can require a more detailed knowledge. Also here, the answers should be short and given on the blank space after each question. Only requested calculations have to be presented. Maximum two points per question.

Part 3 Consists of four questions. All assumptions and calculations made should be presented. Reasonable simplifications may be done in the calculations. All calculations and answers should be on separate papers (not in the exam). Each question gives maximum five points.)

The maximum sum of points is 40 and to pass the exam (grade 3) normally 18 points are required. There is no requirement of a certain number of points in the different parts of the exam. The answers may be given in English or Swedish.

The result will be reported at 2013-03-28 at the latest. The exams will then be available at IMT.

GOOD LUCK!

| AID: | Exam Date: 2013-03-14 |
| --- | --- |
| Course Code: TBMI26 | Exam Code: TEN1 |

Part 1

1. Write the mathematical expression of how a linear classifier determines a class label.

2. What is the difference between regression and classification?

3. If you have 100 training examples and perfom a 5-fold cross-validation to evaluate the performance of a classifier, how many times must you train the classifier?

4. Mention one advantage and one disadvantage of the k-nearest neighbor classifier.

5. What is the difference between *gradient descent* and *gradient ascent*?

| AID: | Exam Date: 2013-03-14 |
| --- | --- |
| Course Code: TBMI26 | Exam Code: TEN1 |

6. You have manually labeled training data for training an AdaBoost classifier. You happen to give one training example the wrong class. Why could this be a particular problem with the AdaBoost classifier?

7. In k-means clustering, the user must set one paramter. What does this parameter control?

8. How is a policy in the context of reinforcement learning usually represented?

9. Write down the cost function that is optimized in Principal Component Analysis and any constraints there may be.

10. How is the accuracy of a classifier calculated?

| AID: | Exam Date: 2013-03-14 |
|---|---|
| Course Code: TBMI26 | Exam Code: TEN1 |

Part 2

11. For a medical diagnosis problem of a certain disease, there are measurements of different medical parameters (blood pressure, blood values, etc.) that are thought to be relevant for the disease. Measurements from both patients with the disease and from healthy persons are available. The medical doctors have asked you to help them analyse and use this data to improve and simplify the diagnosis of new patients.

    a) Explain briefly what you could do with *supervised learning* techniques.

    b) Explaing briefly what you could do with *unsupervised learning* techniques.

12. Illustrate the concept of *overtraining* by drawing a feature space and the classification boundary of a classifier that has overtrained. Why do linear classifiers, in general, not overtrain?

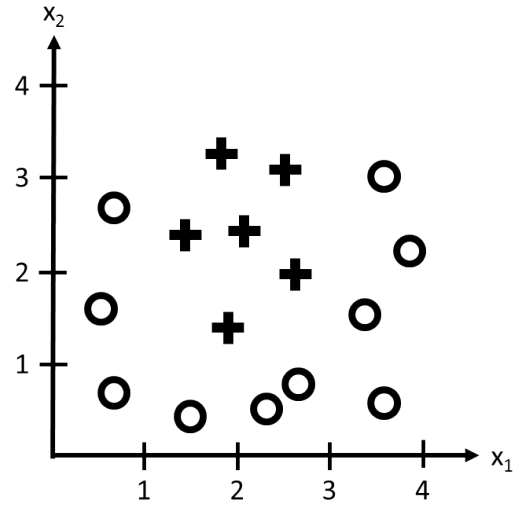
13. Consider a Gaussian kernel function $\kappa(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{4}}$. What is the distance between two feature vectors

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ and } \mathbf{x}_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

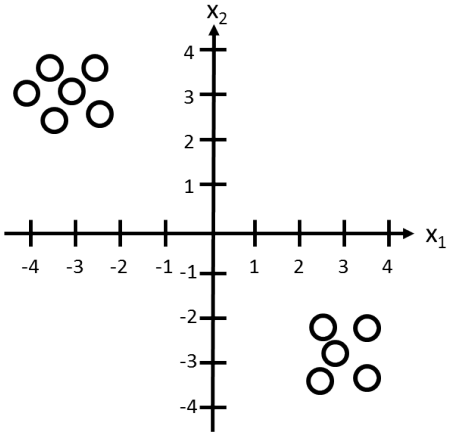
    in the new feature space defined by this kernel function?

| AID: | Exam Date: 2013-03-14 |
|---|---|
| Course Code: TBMI26 | Exam Code: TEN1 |

14. The figure below shows a feature space with two classes. Draw a Classification and Regression Tree (CART) that classifies the two classes and draw the classification boundaries in the figure.




15. The figure below shows 2-dimensional training examples. Sketch and draw what happens if you first apply PCA to reduce the dimension of the training data to 1, and then apply k-Means clustering to the result. Also give approximate numbers of what would be the result of the k-Means algorithm.



| AID: | Exam Date: 2013-03-14 |
|---|---|
| Course Code: TBMI26 | Exam Code: TEN1 |

Part 3

16. Assume two signals $s_1(t)$ and $s_2(t)$. Let $n_i(t)$, $i = 1 \dots 6$ be independent realizations of white normal noise with equal variance. We now want to analyse the three different pairs of signal mixes (a-c) using CCA. Please answer the following three questions for each pair of signal mixes.

    - How many CCA solutions exist? (1p)
    - Specify $\mathbf{w}_x$ and $\mathbf{w}_y$ for each solution. (2p)
    - Define which of the following alternatives that are correct for the canonical correlation $\rho$:
      $|\rho| = 0$; $0 < |\rho| < 1$ or $|\rho| = 1$. (2p)

    a)
    $$\mathbf{x}(t) = \begin{pmatrix} s_1 + n_1 \\ s_2 \end{pmatrix} \text{ and } \mathbf{y}(t) = \begin{pmatrix} s_1 + s_2 \\ n_2 \end{pmatrix}$$

    b)
    $$\mathbf{x}(t) = \begin{pmatrix} s_1 + n_1 \\ s_2 + n_2 \end{pmatrix} \text{ and } \mathbf{y}(t) = \begin{pmatrix} n_3 \\ n_4 \\ s_1 + n_5 \end{pmatrix}$$

    c)
    $$\mathbf{x}(t) = \begin{pmatrix} n_1 \\ n_2 \\ 0.25s_1 + n_3 \end{pmatrix} \text{ and } \mathbf{y}(t) = \begin{pmatrix} n_4 \\ s_1 + n_5 \\ 0.5s_1 + n_6 \end{pmatrix}$$

17. You have the following data:

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

    where $\mathbf{X}$ contains five 2d-samples (one per column), and $\mathbf{Y}$ contains the classification labels for the corresponding samples.

    a) Perform the first AdaBoost iteration on the data $\mathbf{X}$. Sketch the classification problem. Use 'decision stumps' as weak classifiers. (2p)

    b) Perform the second AdaBoost iteration on the data $\mathbf{X}$. Sketch the classification problem. Use 'decision stumps' as weak classifiers. (2p).

    c) Sketch the final strong classifier $H(\mathbf{X})$. Does AdaBoost work well in this setting? (1p)

    *Hint 1:* The standard way of updating the weights in the standard AdaBoost method is $d_{t+1}(i) \propto d_t(i)e^{-\alpha_t y_i h_t(\mathbf{x})}$, where $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$.
    *Hint 2:* The final strong classifier: $H(\mathbf{X}) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

| AID: | Exam Date: 2013-03-14 |
|---|---|
| Course Code: TBMI26 | Exam Code: TEN1 |

18. The figure below shows a state model of a system were the task is to get from state "1" to an end node with maximal reward over time. The allowed actions are "up","down" and "right", see figure. When trying to moving right from state "1" there is a chance $p$ of ending up in state "3", and a chance $1-p$ of ending up in state "4" ($0 \leq p \leq 1$).

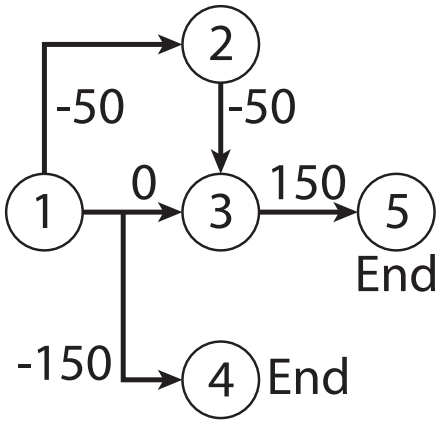



Figure 1: The state model. The numbers next to the arrows are the rewards for moving from one state to another.

a) Calculate the estimated Q-function after the system has trained by moving $1 \rightarrow 3$, $3 \rightarrow 5$, $1 \rightarrow 4$, $1 \rightarrow 4$, $1 \rightarrow 3$, $3 \rightarrow 5$. Use the discount factor $\gamma = 1$ and repeat the learning using the learning rates $\alpha = 0.5$ and $\alpha = 1$. Initiate the Q-function with all zeros and update it after each move. (2p)

b) Calculate the expected Q and V function after a very large number of iterations using $p = 0.5$, $\alpha = 0.5$ and the discount factor $0 < \gamma \leq 1$. (1+2p)

19. Consider a *single*-layered neural network (i.e., no hidden layer) with the sigmoid activation function $\sigma(z) = 1/(1+e^{-z})$.

a) Derive the backpropagation batch update rule for training the network using the error function $\epsilon = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{M}(y_i^n - u_i^n)^2$. $N$ is the number of training samples, $M$ is the number of output nodes, $y_i^n$ the expected output and $u_i^n$ is the output of a neuron with the activation function $\sigma$ from above. Draw the network and declare all indices, matrices etc that are used. (2p)

b) Sometimes it is desired that the sum of output nodes equals 1, so that each output can be interpreted as a probability. One way of accomplishing this is to normalize the output from each output node to be $h(z_i) = \frac{\sigma(z_i)}{\sum_{k=1}^{M}\sigma(z_k)}$, where $\sigma(z)$ is the activation function from a). Derive the *online* update rule in parameter form for the single layered network with this new normalized activation function. If you want, you can assume two outputs ($M = 2$). (3p)

*Hint:* The derivative of the function $\sigma(z)$ is conveniently found as $\sigma'(z) = \frac{d\sigma}{dz} = \sigma(z)(1-\sigma(z))$.