Friedrich-Alexander-Universität Erlangen-Nürnberg
Faculty of Engineering
Chair of Multimedia Communications and Signal Processing
Prof. Dr. Vasileios Belagiannis

Machine Learning for Signal Processing
Test (Not-graded), Winter Semester 2022/23

Duration: 60 Minutes Date: 20.12.2022

Name: _____

Matriculation Number : _____

Study Program: _____

Degree (Bachelor/Master): _____

Information

- Only the provided exam shall be used for completing the tasks. If you need more blank sheets, please contact the examiner.
- Please include your matriculation number on each sheet.
- Please reply to each question in a separate answer box. Boxes are provided below the questions.
- Please write cleanly and legibly. Sections that we cannot read will be scored zero.
- Results without any comprehensible justification or without calculations for arriving at the answers can not be graded.
- Allowed items: only a permanent pen (not of red colour).
- The total number of pages is 8.

Point Distribution

Task	1	2	3	Total
Points:	18	15	15	48
Obtained Points:				

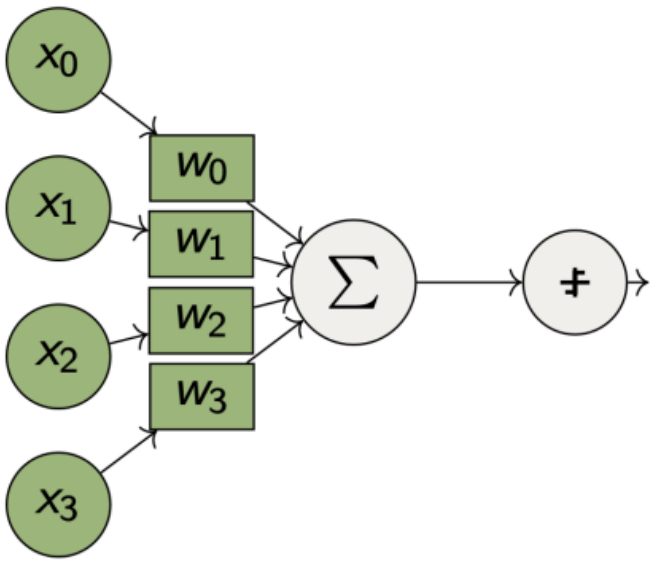
Grade: _____

Task 1 Multi-Topic Questions

1. Reply the following questions using the empty boxes.

   (a) (3 points) Define the threshold function of the McCulloch-Pitts Neuron (A). Draw the McCulloch-Pitts Neuron (B). What is the major difference between the McCulloch-Pitts neuron and Perceptron (C)?
   Reply to each question in a separate answer box.

$$f(x) = \begin{cases} 0, if\ \boldsymbol{w}x \leq T \\ 1, otherwise \end{cases}$$



The major difference is in their learning abilities. McCulloch-Pitts neurons lack learning capabilities, its weights and thresholds remain constant. In contrast, Perceptron can learn through supervised learning. They adjust their weights based on prediction errors.

(b) (4 points) Does the generalization error refer to the training or test error? (A) What is the i.i.d assumption in the data generation process (with respect to the train and test set) (B)?
Reply to each question in a separate answer box.

The generalization error typically refers to the test error. It measures how well a machine learning model performs on unseen data that it hasn't been trained on. It helps assess how well the model has learned to generalize patterns from the training data to new, unseen examples.

The i.i.d (independent and identically distributed) assumption in the data generation process means that the data points in both the training and test sets are assumed to be independent of each other and drawn from the same probability distribution. In other words, each data point is not influenced by or dependent on the others, and they all come from the same underlying data distribution.

(c) (3 points) What is a major disadvantage of the Empirical Risk Minimization? (A) List two gradient-based optimizers (B).

A major disadvantage of Empirical Risk Minimization (ERM) is its susceptibility to overfitting. ERM aims to minimize the training error, which can lead to models that perform poorly on unseen data because they have learned noise or specific characteristics of the training data.

Stochastic Gradient Descent (SGD): It updates model parameters using the gradient of the loss function with respect to a single randomly chosen training example at each iteration.
Adam (Adaptive Moment Estimation): Adam is an adaptive learning rate optimization algorithm. It adjusts the learning rates for each parameter individually and keeps a running estimate of the second moment of the gradients.

(d) (2 points) What are the two major limitations of the Linear Discriminant Analysis?
Reply to each question in a separate answer box.

Assumption of Gaussian Distributions: This assumption may not hold in real-world datasets, and if the data significantly deviates from this assumption, LDA's performance can be compromised.

Linearity Assumption: This linearity assumption may not capture complex relationships in the data, especially when the true decision boundary is nonlinear.

(e) (3 points) What does the model capacity determine? Which two phenomena can help to avoid?
Reply to each question in a separate answer box.

The model capacity determines the ability of a machine learning model to fit complex patterns in data.

The two phenomena that model capacity helps to avoid is
1. overfitting
2. underfitting.

(f) (3 points) Consider a convolutional network trained for a classification task on the ImageNet dataset. The input to the network is RGB images of size 224×224. How many parameters are present in the first convolution layer of the network that uses 16 filters of size 5×5 with a stride of 1, and each filter has a bias term? Assuming no padding is used, what will be the dimensions of the convolution layer output? Express the result in the form: feature maps $\times$ height $\times$ width.

Parameters = (filter_height x filter_width x depth + bias) * total_filters
Parameters = (5 x 5 x 3 + 1) * 16 = 1216

Output size = [(Input size - Filter size + 2*Padding) / Stride] + 1
Output size = [(224 - 5 + 2*0) / 1] + 1 = 220

Ans: 16 x 220 x 220

The number of parameters in a convolutional layer is given by the formula:

Parameters = (Shape + bias) * no. of parameters
In this case, (filter_height x filter_width x filter_depth(RGB) + bias) * no. of parameters

Here, the filter size is 5x5, number of input channels is 3 (RGB image), bias added, total filter is 16.
So, the total number of parameters in the first convolutional layer would be:

Parameters = (5 x 5 x 3 +1) x 16 = 1216

The output dimensions of a convolutional layer are given by the formula:
Output size = [(Input size - Filter size + 2*Padding) / Stride] + 1

Here, the input size is 224x224, the filter size is 5x5, stride is 1 and no padding is used. So, the output dimensions would be:
Output size = (224 - 5 + 2*0) / 1 + 1 = 220

So, the output dimensions in the form: feature maps × height × width would be:
16 x 220 x 220.

Task 2 Linear Regression

2. Reply the following questions using the empty boxes.

   (a) (5 points) Consider the target value y and the linear model $f(x)$ with x as input. (A) Define the mean absolute error for m training samples as the loss function (L1 loss). (B) Draw the derivative of the loss function.
   Reply to each question in a separate answer box.

$$L1 = \frac{1}{m}\sum_{i=1}^{m} |y_i - f(x_i)|$$

The derivative of the L1 loss function is not as straightforward as the L2 loss function because the absolute value function is not differentiable at zero. However, we can express it using the sign function:
If y_i f(x_i) > 0, then the derivative is -1.
If y_i f(x_i) < 0, then the derivative is 1.
If y_i f(x_i) = 0, then the derivative is undefined.

   (b) (5 points) Consider the linear regression problem with the mean square error loss function. Write down the loss function given that the model with parameterised by $\mathbf{w}$, the input is given by $\mathbf{x}$ the ground-truth by $\mathbf{y}$ (A). Explain the two approaches to obtain the optimal parameters $\mathbf{w}^*$ (B).
   Reply to each question in a separate answer box.

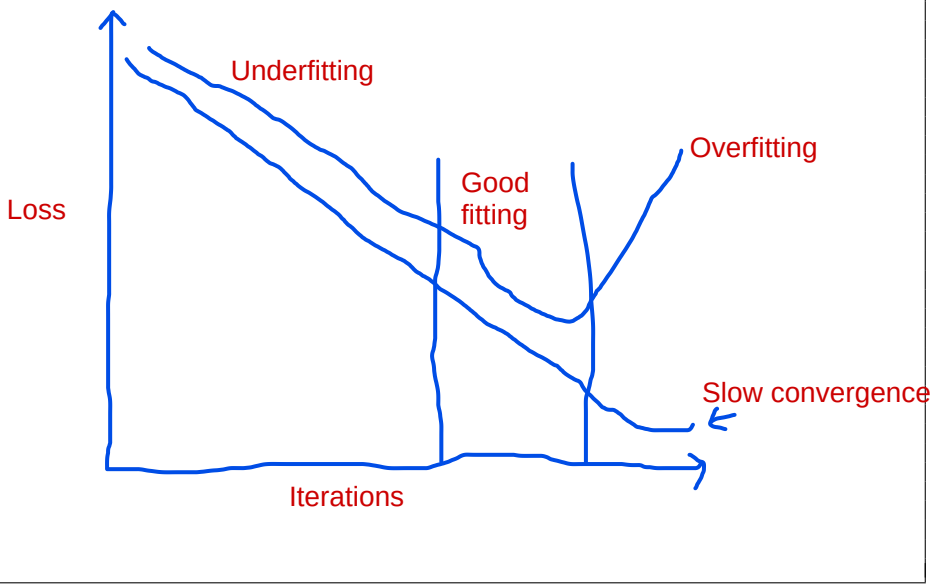
$$L(w) = \frac{1}{m}\sum_{i=1}^{m} (y_i - w^T x_i)^2$$

1. Gradient Descent: This is an iterative optimization algorithm for finding the minimum of a function. To find a local minimum, the function takes steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point. The learning rate determines how big these steps are.

2. Normal Equation: This is an analytical approach to linear regression with a least square cost function. We can directly find out the value of w where our cost function reaches its minimum value. The normal equation formula is given by:

$$w=(X^T X)^{-1} X^T y$$

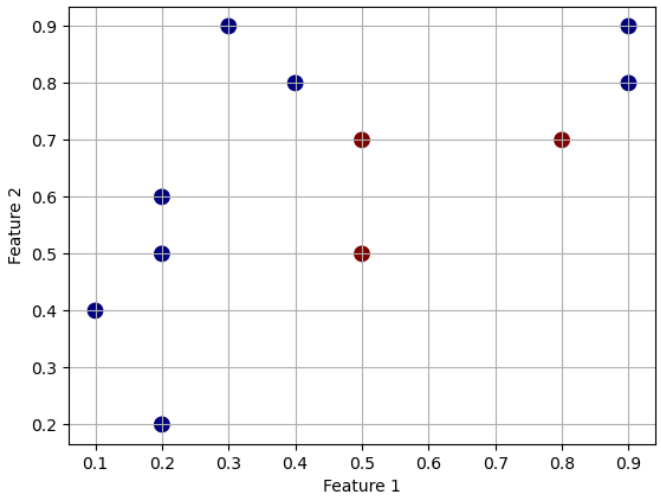
(c) (5 points) Consider the L1-loss function for a number of iterations. Draw the loss vs iteration figure and illustrate the following training situations: good-fitting, under-fitting, over-fitting, and slow convergence



Task 3 Decision Trees

3. Reply the following questions using the empty boxes.

   (a) (5 points) In decision trees, evaluating the possible split is based on the purity of a node. One purity measure is the Gini score that is defined as: $g(\mathbf{p}) = \sum_{i=1}^{k} p_i(1-p_i)$ with $\mathbf{p} = \{p_1, \ldots, p_k\}$ and p_i the fraction of samples that correspond to class i of total k classes. The score is 0 if all samples are from the same class and it increases as the class mix becomes uniform. Calculate the Gini score for the samples drawn in the figure. There are 11 samples and 2 classes in total.



Fraction of sample of class A "Blue"(p_a) = 8/11
Fraction of sample of class B "Red"(p_b) = 3/11

Calculate Gini score/impurity = 1 - [(8/11)^2 + (3/11)^2]

Gini Impurity = 48/121 = 0.3966

   (b) (5 points) The decision tree formation is an recursive process that splits the training samples into subsets. The criterion to split the node data is to minimise the Gini impurity. The goodness-of-split can be defined as: $F(s,n) = g(\mathbf{p}) - S_l * g(\mathbf{p}_l) - S_r * g(\mathbf{p}_r)$ where s is the split function for node n. S_l corresponds to the fraction of samples at the left node and S_r to the right one accordingly. The higher the value of $F(s,n)$, the better the split is. Grow a decision tree until

you reach leaves (end nodes) with samples of the same class. The goal is to seek for the best split function per node. Report for each node the goodness-of-split and the selected split function.

For X1 = 0.4; pl = x1 <= 0.4
pr = x1 > 0.4

pl:-
Total = 6
Red = 0
Blue = 6
Sl(Gini Impurity) = 6/11
g(pl) = 1-[(0/6)^2 + (6/6)^2]
g(pl) = 0

pr:-
Total = 5
Red = 3
Blue = 2
Sr(Gini Impurity) = 5/11
g(pr) = 1-[(3/5)^2 + (2/5)^2]
g(pr) = 0.48

F(s,n) = g(parent_node) - Sl*g(pl) - Sr*g(pr)
= 0.3966 - (6/11)*0 - (5/11)*0.48
=0.1785 (Which is higher then other value of x1)

For X2 = 0.7; pl= x2 <= 0.7
pr = x2 > 0.7

pl:-
Total = 3
Red = 3
Blue = 0
Sl(Gini Impurity) = 3/5
g(pl) = 1-[(3/3)^2 + (0/3)^2]
g(pl) = 0

pr:-
Total = 2
Red = 0
Blue = 2
Sr(Gini Impurity) = 2/5
g(pr) = 1-[(0/2)^2 + (2/2)^2]
g(pr) = 0

F(s,n) = 0.48 - (3/5)*0 - (2/5)*0
=0.48 (Which is higher than any other value for x2)

(c) (5 points) Based on the constructed tree classify the following samples: (0.3, 0.2), (0.6,0.5), (0.3, 0.8), (0.6,0.7).

1) (0.3, 0.2) -> BLUE
2) (0.6, 0.5) -> RED
3) (0.3, 0.8) -> BLUE
4) (0.6, 0.7) -> RED