EXAMINATION QUESTIONS

Faculty: **Science and Technology**

Examination in: **DAT200** (*Course code*) **Applied Machine Learning** (*Course name*)

Time for exams: Monday, 27.05.2019 (*Day and date*) 14:00 – 17:30 (3.5 hours) (*As from – to and duration of examinations (hours)*)

Course responsible: Oliver Tomic and Ulf Indahl (*Name*)

Permissible aids:

A1: no calculator, no other aids

The exams papers includes: 6 (*Number of pages incl. attachment*)

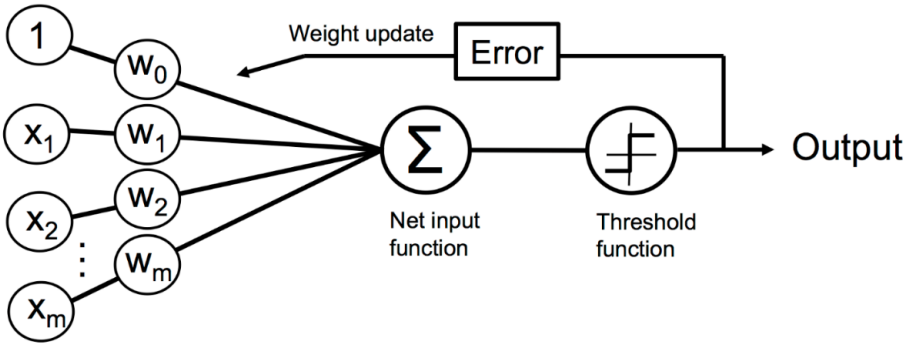
If the examination consists of several parts, information must be given as to how much each part will count toward the grade

Course responsible: Oliver Tomic (9574 6167) and Ulf Indahl

External examiner: Tormod Næs

Exercise 1 (12 points in total)

The perceptron algorithm.



Base your answers on the figure above.

a) (4 points)
Explain briefly how the Perceptron works when classifying a sample (no explanation on error computation or weight update needed here).

b) (4 points)
How are the ”net input function” and ”threshold function” formulated (explain and/or provide formulae)?

c) (4 points)
How are the weights updated?

Exercise 2 (16 points in total)

a) (6 points)
What are unsupervised and supervised learning? When should you use them?

b) (6 points)
PCA, PCR and PLSR are related methods. In which way are they unsupervised/supervised?

c) (4 points)
In machine learning, PCA is sometimes included in pipelines for regression or classification. What is the role of PCA in the pipeline? What does the hyperparameter for PCA control?

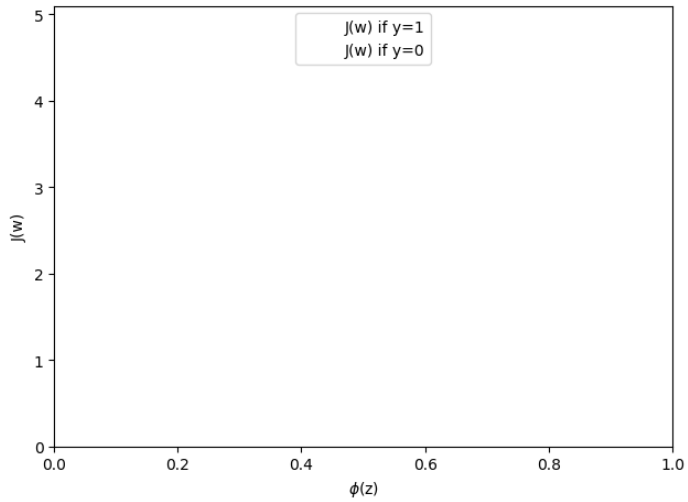
Exercise 3 (7 points in total)

Below the cost function of the logistic regression algorithm for a single sample is given.

$$J(\boldsymbol{w}) = -y \log\big(\phi(z)\big) - (1-y) \log\left(1 - \phi(z)\right)$$

a) (4 points)
What does the logistic regression cost function above simplify to when a sample belongs to either class 0 or 1?

b) (3 points)
Sketch the logistic regression cost function in a figure as shown below. Indicate in the plot which part of the cost function represents class 0 and class 1 respectively.
Do not sketch it on this sheet, but on the sheets where you provide your answers!



Exercise 4 (10 points in total)

a) (5 points)
Regularization/penalization is often used in machine learning models. Why would you regularize a model? Explain briefly.

b) (5 points)
Which are the two most common types of regularization in machine learning and how/what do they shrink?

Exercise 5 (13 points in total)

a) (3 points)
Explain the concept of majority voting in classification.

b) (5 points)
How is majority voting applied in bagging? Why may bagging be more accurate than using a single classification model?

c) (5 points)
How is majority voting applied in K-nearest neighbours (KNN) classification? Why is KNN usually too resource demanding when the training data set is very large?

Exercise 6 (7 points in total)

One-vs-Rest (OvR) (also called One-vs-All (OvA)) is an often used strategy when working with classifiers.

a) (5 points)
Explain briefly the concept of OvR / OvA.

b) (2 points)
Explain briefly how new samples are classified with OvR / OvA applied to the Perceptron algorithm.

Exercise 7 (10 points in total)

Feature importance is an important tool for determination of which variables are important to a model. Assume you have a regression model and you would like to know the importance of each feature of the model. The feature importance methods available to you are dropout/drop column feature importance and feature importance permutation.

a) (6 points)
Explain how feature importance is computed using dropout/drop column feature importance.

b) (5 points)
Explain how feature importance is computed using feature importance permutation.

c) (4 points)
Given a situation where you have only limited hardware resources (it would take a long time to re-train the model), which of the methods mentioned in sub-exercise a)/b) would you use? Explain why you would make that choice.

Exercise 8 (20 points in total)

You work as a data scientist at hospital A. Your department leader wants you to analyse data of 500 patients that were treated for myocardial infarction (heart attack) at hospital A. You are supposed to build a model (based on 15 clinical features) that can be used to predict whether a patient will be dead (class 1) or alive (class 0) after the treatment. Note that about 85% of the patients survive the treatment. Your department leader asks you to train a K-Nearest Neigbours model using a `scikit-learn` `pipeline`. Here is what you are supposed to implement in a script using `pandas` and `scikit-learn`:

1. Use the data provided to you in a comma-separated CSV-file named "infarction.csv". The first column contains integer class labels, the remaining 15 columns are clinical features of type float. The data has no header.
2. Use pandas to load the data.
3. Reduce the dimension of the data from 15 features to three features using PCA.
4. Train a KNN classifier with those three extracted features using grid search (5-fold cross validation). Search across the following KNN parameters:
    a. 1, 2, 3, 4 and 5 neighbours (`'kneighborsclassifier__n_neighbors'`)
    b. Values 1 and 2 for p (the parameter of the distance metric 'Minkowski', `'kneighborsclassifier__p'`)
    c. Make sure to use all processor cores on your computer
5. Use 20% of the data to test the model
6. After completed grid search, print the best score and the best parameters of the KNN

Use the list of arbitrary commands provided in **scikit-learn / pandas code** on the last page for support. Make sure that you insert your choices of parameters in those places highlighted with '*YOUR_INPUT*' or '*YOUR_ESTIMATOR'*. NOTE: you don't need to use all commands on this list to make your script work. Regular Python code (also needed in your script) is not provided.

scikit-learn / pandas code

```
X_train, X_test, y_train, y_test = train_test_split(YOUR_INPUT, YOUR_INPUT, test_size=YOUR_INPUT,
      stratify=YOUR_INPUT, random_state=YOUR_INPUT)
from sklearn.ensemble import RandomForestClassifier
YOUR_ESTIMATOR.best_params_
from sklearn.metrics import accuracy_score
from sklearn.decomposition import PCA
from sklearn.metrics import precision_score
np.hstack((YOUR_INPUT, YOUR_INPUT))
'kneighborsclassifier__p'
PCA(n_components=YOUR_INPUT)
from sklearn.neighbors import KNeighborsClassifier
YOUR_ESTIMATOR.fit(YOUR_INPUT, YOUR_INPUT)
from sklearn import datasets
GridSearchCV(estimator=YOUR_INPUT, param_grid=YOUR_INPUT, scoring=YOUR_INPUT, cv=YOUR_INPUT)
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
pandas.read_csv(YOUR_INPUT, sep= YOUR_INPUT, header=YOUR_INPUT)
from sklearn.metrics import accuracy_score
X, y = someName.iloc[YOUR_INPUT, YOUR_INPUT].values, someName.iloc[YOUR_INPUT, YOUR_INPUT].values
from sklearn.preprocessing import StandardScaler
YOUR_ESTIMATOR.best_score_
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import recall_score
from sklearn.pipeline import make_pipeline
YOUR_ESTIMATOR.transform(YOUR_INPUT)
from sklearn.metrics import f1_score
ax.axhline(y=YOUR_INPUT, linewidth=YOUR_INPUT, color=YOUR_INPUT, linestyle=YOUR_INPUT)
from sklearn.datasets import make_moons
accuracy_score(YOUR_INPUT, YOUR_INPUT)
'kneighborsclassifier__n_neighbors'
import pandas as pd
from sklearn.metrics import roc_auc_score
```