UiT

THE ARCTIC UNIVERSITY OF NORWAY

EXAMINATION PAPER

Portfolio Assignment 2

Home exam in:	**FYS-2021 - Machine Learning**
Hand-out:	**Friday October 16, 2020, 10:00**
Hand-in:	**Friday October 30, 2020, 13:00**

The exam contains 9 pages including this cover page

Contact person:	**Karl Øyvind Mikalsen**
Email:	**karl.o.mikalsen@uit.no**

UiT

THE ARCTIC UNIVERSITY OF NORWAY

Before You Start

Module examination

This is the second portfolio assignment for FYS-2021 Machine Learning. Portfolio assessment of project assignments counts about 40 % of the final grade in the course. All modules in the portfolio are assessed as a whole and one combined grade is given. Note that access to the final examination requires submission and approval of project assignments. Please familiarize yourself with the Regulations for examinations at UiT: https://en.uit.no/exams/art?p_document_id=523936

The report and code should be your own individual work. Remember to cite all sources. More information on source use, plagiarism, and cheating can be found here: https://en.uit.no/sensor/art?p_document_id=684332

Portfolio instructions

Read carefully: Failure to follow the instructions below may have a negative impact on the grade of your submission, or even cause your submission be deemed invalid.

Learning to write a scientific report is an important skill that many of the courses at the Faculty of Science and Technology, including this one, aim to improve. Therefore any *question* that you answer should be contained within the report of this portfolio assignment. Answers outside of the written report, (e.g. in the comment of the code, or within a Jupyter Notebook), will not be considered as a part of your answer of the problem. You can structure your report by having a separate (sub)section with the answer for each question. Remember to cite any sources you use - be advised that your submission will be checked for plagiarism.

Make sure your report shows that you understand what you are doing. More specifically, it is important to elaborate your answers such that essential theory, equations, and intuition is included in your answers. However, your answers should still remain concise and stay focused on the core problem, e.g. there is no need to derive or prove an equation unless the problem asks you to.

Problems that ask for numeric values or plots should include these in the answer of the report.

The code should be commented in such a way that any person with programming knowledge should be able to understand how the program works. Like your report, the code must be your own individual work.

You are permitted to use standard built-in functions and/or packages (e.g. *numpy* and *matplotlib* in

Python) for reading the data and basic calculations. However: make sure that the packages you use do not over simplify your implementation! Of course, all implementations asked for in the problems should be your own work.

Hand-in format

Your report must be a single file as in portable document format (`.pdf`). The file name *has to* follow the format `portfolio2_candidate`*XX*`.pdf` (replace *XX* with your candidate number obtained from StudWeb) for anonymity. Do not put your name in the report or code. Failure to do this may compromise your anonymity. Be advised that the name of the files are visible to the reviewers.

The code you write for this assignment should be included *both* in the appendix of the report *and* submitted as separate files in WISEflow.

Follow the hand-in instruction in WISEflow and **make sure to submit before the WISEflow room closes.**

Resources

All datasets required to answer the exercises can be found in the Canvas room for the course.

Seal Classification

The following is a current problem of great importance to the Institute of Marine Research (IMR - Havforskningsinstituttet). It is very important to be able to estimate the population of seals, and in particular seal pups. The most effective way to do this is from aerial images. However, this is a very challenging problem. Fig. 1 (left) shows an example image. One particular challenge is to distinguish between harp seal pups and hooded seal pups. We have access to a dataset consisting of several thousand aerial RGB images acquired during surveys in the West Ice east of Greenland in 2007 and 2012 and east of New Foundland, Canada, in 2012. The images are acquired from approximately 300m altitude, and the pixel spacing is about 3cm (depending on the exact flight altitude). A typical image size is 11500×7500 pixels. From these images potential locations of seals are identified, Fig. 1 (right), and then crops are extracted.

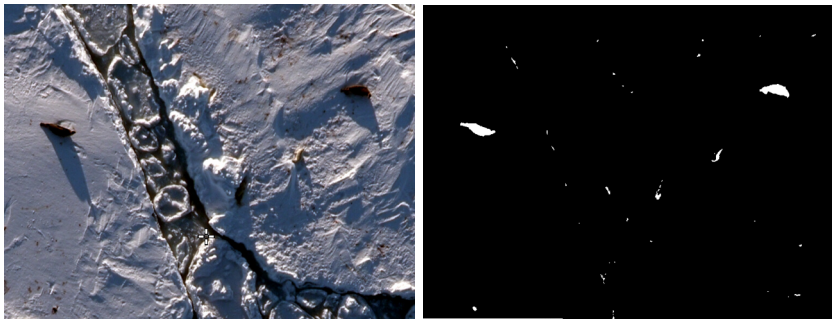

Figure 1: *Aerial images (left) and potential locations of seals (right).*

A crop is shown in Fig. 2. The Institute of Marine Research, Norway, and the Northwest Atlantic Fisheries Centre, Canada, have provided the images and the ground truth information about the type and location of the seal pups in the images.

Your challenge is to classify images into the two categories *harp seal pups* (Class 0) or *hooded seal pups* (Class 1).

In the following exercises, you are going to write machine learning algorithms to classify which type of seal there is in a given image.

The grayscale images used to create the dataset are located in `seals_images_*.csv`. The indices in both files matches themselves. (E.g. row 4 from `seals_*.csv` corresponds to image 4 in `seals_images_*.csv`). To plot the images, they would have to be reshaped to 64×64, e.g.

```
// load the file
images=load_file("seals_images_*.csv")
// Extract the first row, that contains 4096=64 × 64 points
one_image=images[1]
// Reshape it to be 64 × 64
reshaped_image=Reshape(one_image, 64,64)
// Show the image
imshow(reshaped_image)
```

Figure 2: *An image crop. A classifier needs to determine if this a harp seal pup or a hooded seal pup!*

Preprocessing of the Seal Data

You have a very nice colleague who has preprocessed the images and extracted 128 features (see below in the preprocessing section for information on how these features were obtained) from a subset of 1420 images. The preprocessed images are contained in the data files, `seals_train.csv` and `seals_test.csv`. The label of each image is in the first column, and the 128 features in the subsequent columns. Each row contains extracted features from one image. The training set, `seals_train.csv` should be used to train your algorithms. The testing set, `seals_test.csv` should be used *only* to evaluate the performance of your classifier. Hence, using the testing set for any type of experimentation is *not* allowed.

Convolutional Neural Networks

Deep convolutional neural networks are specifically designed for various types of image recognition. Imagine e.g. a self-driving car. The car need to scan its surroundings and take decisions on the type of objects in its vicinity. Deep convolutional neural networks excel at this, and have had profound impact. The convolutional aspect ensures a certain robustness to image object translations and rotations. Convolutional networks by-pass the hand-crafted feature engineering stage which is often a pre-cursor to classification algorithms in other domains of machine learning. It has been shown that hidden layers in the network pick up properties of images in an increasing order of complexity. See Fig. 3 for an illustration.

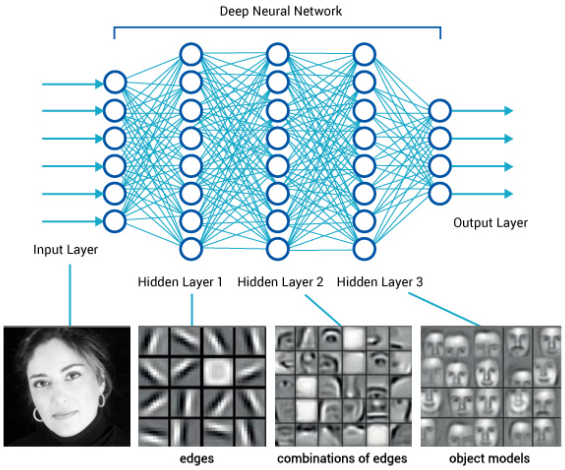

Figure 3: *Illustration of a deep neural networks. Figure from* https://www.saagie.com/blog/object-detection-part1.

Feature selection

As mentioned above, the data consists of 128 extracted features. The images are processed by a convolutional neural network, which generates 4096 numbers per image.

The features are further reduced in dimension by first applying Principal Component Analysis (PCA), and then extracting the hidden representations from an autoencoder. (These preprocessing steps are not important for the report you should write for the assignment).

Problem 1

Note: You are not supposed to do any programming in exercise 1a and 2a.

(1a) Consider the following training set: $\{(\mathbf{x}^i, y^i)\}_{i=1}^6$, where

$$\mathbf{x}^1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \mathbf{x}^2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \mathbf{x}^3 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \mathbf{x}^4 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \mathbf{x}^5 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \mathbf{x}^6 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

and $y^1 = 1$, $y^2 = 1$, $y^3 = 1$, $y^4 = 0$, $y^5 = 0$, $y^6 = 0$.

Create a plot in the 2-dimensional plane that illustrates the decision boundary of a logistic discrimination classifier that gets 0 errors on the training set. Write down an equation for the decision boundary where you use the plot to find approximate estimates for the corresponding weights.

What is the geometrical interpretation of the weights of the classifier? Illustrate this in the figure. What is the distance between the origin and the decision boundary?

Consider a test point

$$\mathbf{x}^t = \begin{bmatrix} x_1^t \\ x_2^t \end{bmatrix}$$

Write down a decision rule that describes how your logistic discrimination classifier would classify the datapoint $\mathbf{x}^t$.

(1b) Implement your own logistic discrimination classifier and use the `seals_train.csv` data to train you classifier. Test your trained classifier on the `seals_test.csv` data. Report the confusion matrix and the accuracy on the test set.

(1c) Briefly explain what the Receiver Operating Characteristic (ROC) curve represents. How could you easily change the true-positive and false-positive rates for the logistic discrimination classifier? Plot the ROC curve and compute the Area Under the Curve (AUC) score for the test set (it is okay to use a built-in function to compute AUC).

(1d) Create a function that plots the seal images corresponding to a given an index. Use the function to plot 5 examples of images that are correctly classified and 5 examples of images that are misclassified by your classifier. Discuss possible reasons as to why these photos are misclassified.

Problem 2

(2a) Consider the same training set as in (1a).

Create a plot in the 2-dimensional plane that illustrates the decision boundary of a decision tree for classification that get 0 errors on the training set. Also create an illustration of the corresponding decision tree (remember to describe what is happening in each of the nodes in the tree).

Consider a test point

$$\mathbf{x}^t = \begin{bmatrix} x_1^t \\ \\ x_2^t \end{bmatrix}$$

Find an IF-THEN rule that describes how the decision tree would classify the datapoint $\mathbf{x}^t$.

(2b) Implement your own decision-tree classifier and use the `seals_train.csv` data to train you classifier. Test your trained classifier on the `seals_test.csv` data. Report the confusion matrix and accuracy on the test set.

Note: Your answer should include an explanation of key design choices, including, but not limited to:

- How you select the maximum depth of the tree.
- The choice of impurity measure.
- How you determine the threshold when spitting a node.

(2c) Explain how you can obtain soft (probabilistic) predictions from the decision-tree classifier. Modify your implementation accordingly, and use these to plot the ROC curve for the decision-tree classifier on the test set. Calculate the AUC score. Compare the performance of the decision tree to the performance of the logistic discrimination classifier.

Problem 3

Missing data are common in many real-world datasets, and might appear for various reasons. For example, a sensor may stop working after some time and therefore the measurements are censored, or a participant in a survey may forget to answer a question in a questionnaire. This leads to *missing values* in the data. In this exercise, we will study different methods for handling missing values using the dataset `censored_data.csv`

(3a) One common technique to handle missing data is to replace or fill in the missing value with some estimated value. This is called imputation. We will now study different imputation methods.

Load the dataset `censored_data.csv` and identify which of the variables (features) that contain missing values. Report the the mean, median and the maximum of the available (observed) data for this variable.

Create three imputed datasets by replacing the all missing values with the following three estimates: 1. mean, 2. median, 3. max.

Load the dataset `uncensored_data.csv` that contains the true values for the missing data and compute the MSE (mean squared error) for the three imputed datasets. (The MSE should be computed only for the datapoints that were imputed and not for the entire dataset). Which of the three methods give the lowest error ?

(3b) Imputation can also be done by regression. In this exercise, we are going to predict the value of the missing values in `censored_data.csv` using linear regression.

First, you should investigate if the variables are correlated by computing the correlation matrix (3x3 matrix that can be computed using built-in functions). Let the variable that is most correlated to the variable with missing values be the independent variable and let the variable with missing values be the dependent variable. Train a linear regression model and predict the missing values. Compute the MSE.

(3c) Implement a decision tree for regression and do the imputation using this model. Report the MSE. Which of the imputation methods give the lowest error ?