Exercise 4 - Classification

Winter term 2019/2020

student1, student2, student3

General information

The assignments are voluntary. All students who choose to participate have to form groups comprising three to four students (not more and not less). The groups do not have to be static, you may form new groups for each assignment. You have **two weeks** to answer the questions and to submit your work. The solutions are going to be presented and discussed after the submission deadline. Sample solutions will **not** be uploaded. However, you are free to share correct solutions with your colleagues **after they have been graded.**

Formal requirements for submissions

Please submit your solutions via Moodle (as a .zip file) as well as in printed form. The .zip file must contain one .pdf file for the pen-and-paper tasks as well as one .py file per programming task. Only pen-and-paper tasks have to be printed, you do not have to print the source code. Only one member of the group has to submit the solutions. Please make sure to specify the matriculation numbers (**not the names!**) of all group members so that all participants receive the points they deserve!

Please refrain from submitting hand-written solutions or images of solutions (*.png* / *.jpg* files). Rather use proper type-setting software like LaTeX or other comparable programs. If you choose to use LaTeX, you may want to use the template files provided.

Code assignments have to be done in Python. Please submit *.py* files (**no jupyter notebooks**). The following packages are allowed for code submissions: `numpy`, `pandas` and `scipy`. Please ask **beforehand**, if you want to use a specific package not mentioned here. Finally, do not use already implemented models (e.g. from `scikit-learn`).

Grading details

Your homework is going to be corrected and given back to you. Correct solutions are rewarded with a bonus for the exam which amounts to at most ten percent of the exam, if all solutions submitted by you are correct (this corresponds to at most six points in the exam). It is still possible to achieve full points in the exam, even if you choose not to participate in the assignments (it is additional). The function which is used to compute the bonus is given by:

$$b(a) = \min\left(B, \left\lceil \frac{B}{A^2} \cdot a^2 \right\rceil\right) \tag{1}$$

- b denotes the number of bonus points you get for the exam (this is up to you)
- B refers to the maximum attainable bonus points for the exam (six points)
- A denotes the maximum attainable points in the assignments (40 points)
- a is the score you achieved in the assignments (this is up to you)

Please note: You have to pass the exam **without the bonus points!** This means that it is not possible to turn a failing grade ($= 5.0$) into a passing grade (≤ 4.0). The bonus points will be taken into account in case you have to repeat the exam (i. e. they do not expire if you fail the first attempt).

Important!

The solutions have to be your own work. If you plagiarize, you will lose all bonus points!

1 Decision Trees

a) ID3 Decision Tree Construction (3 points)

You are given the following labeled data set. Construct a decision tree classifier using pen and paper. Apply the information gain splitting heuristic. Constructing the first two levels of the tree is sufficient. Draw the tree and indicate each splitting attribute. Show your calculations.

Outlook	Temperature	Humidity	Wind	Sport
sunny	cold	high	weak	soccer
cloudy	cold	low	strong	soccer
sunny	warm	low	weak	soccer
rainy	cold	high	weak	squash
sunny	cold	high	weak	squash
rainy	warm	high	strong	squash
cloudy	cold	high	weak	squash
rainy	warm	high	weak	squash
cloudy	warm	high	weak	tennis
cloudy	cold	low	strong	tennis
sunny	cold	low	strong	tennis
cloudy	cold	high	weak	tennis

Solution:

2 Neural Networks

a) Hyperparameter exploration (2 points)

On the *TensorFlow Playground* webpage[1] try varying the hyper-parameters of an MLP (# hidden layers, # neurons per layer) using the *'Circle'* classification data set. Does it work better to ❶ use more neurons near the input layer, ❷ more neurons towards the last hidden layer or ❸ use the same number of neurons in each hidden layer? Provide a justification for why ❶, ❷ or ❸ might work better. Report the best configuration which you found. Can a perceptron separate the circular dataset? Why or why not?

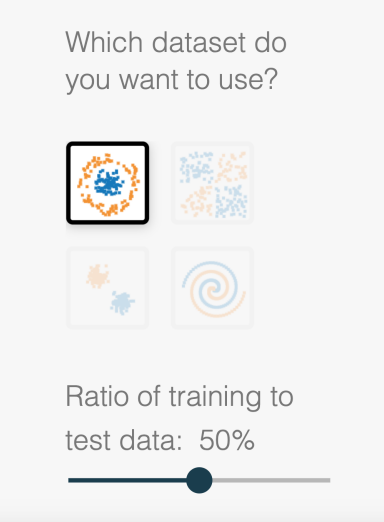

Figure 1: Mandatory settings on TensorFlow Playground for this exercise

Solution:

[1] `https://playground.tensorflow.org`

b) Multi-Layer Perceptron for Sentiment Analysis (5 points)

Implement an MLP to classify movie reviews into either positive or negative sentiment using the deep learning library `PyTorch`.[2] The data is stored in the folder `/data`. Each line i in `labels.txt` contains the label of the i-th movie review in `reviews.txt`. You have to map each of the movie review texts to a fixed-size embedding vector, which you can use as input to your MLP. You can do this by using the `flair` library.[3] Install it by running `pip install flair`. Perform a 3-fold cross-validation / random subsample validation and report precision and recall. Also, report your hyper-parameter configuration (learning rate, batch size, network structure, etc.).

Solution:

c) Bonus Question: Contextualized Word Embeddings (1 point)

Read the paper *'Deep contextualized word representations'*[4] and answer the following questions: What is the most important difference between word2vec embeddings and the ELMo model proposed in the paper? Why does this difference have a large effect on the quality of the resulting word embeddings?

Solution:

[2] `https://pytorch.org/`
[3] `https://github.com/flairNLP/flair`
[4] Peters et al., `https://arxiv.org/pdf/1802.05365.pdf`