Exercise 4 - Classification

Winter term 2019/2020

Important

Please solve the assignments in groups of 3 to 4 students. 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**. Please submit your solutions via Moodle **and** in printed form. Only one member of the group has to submit the solutions. Therefore, make sure to specify the names of all group members. Please do not submit hand-written solutions, rather use proper type-setting software like LaTeX or other comparable programs.

Your homework will be corrected and given back to you. Correct solutions are rewarded with a bonus for the exam (max. 10 percent, if all solutions submitted are correct). **Please note:** You have to pass the exam **without the bonus points**! *(i.e. it is not possible to turn 5.0 into 4.0)* The solutions have to be your own work. If you plagiarize, you will lose all bonus points!

Further remarks:

- Code assignments have to be done in `Python`
- The following packages are allowed: `numpy`, `pandas`
  (please ask, if you want to use a specific package not mentioned here)
- **Do not use already implemented models** (e. g. from `scikit-learn`)

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	Which 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?

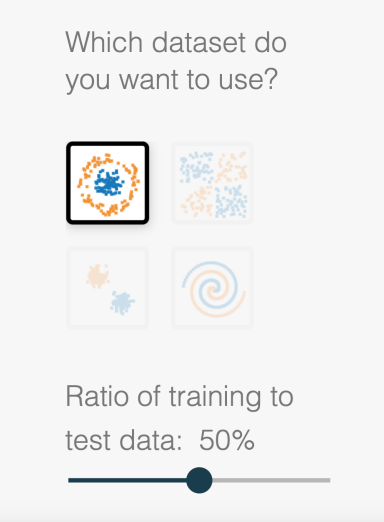

Figure 1: Mandatory settings on TensorFlow Playground for this exercise

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 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`