FACHHOCHSCHULE KIEL
University of Applied Sciences

Interdisciplinary Master's Program Data Science

Prof. Dr. Stephan Doerfel WiSe 2022/23

Part I «Portfolio-Exam»

MADS-DL

This is the first part of the portfolio-exam for the Data Science course MADS-DL (Deep Learning), worth 49% of the final grade (98 points).
This exam is homework. Student's are allowed to exchange ideas. However, this is NOT a teamwork exercise. Every student must derive and write up their own solutions in their own words and programming style. To complete this first part of the exam,

- solve ALL the following tasks (3 pages!),
- create a Jupyter Notebook for your (commented) code as well as all textual answers (English or German), and
- upload both files to Moodle **before 23:59 o'clock (German time) October 23, 2022**.

Rules and Hints:

- In your notebooks, please complete the tasks in sequence.
- I will rerun the notebook. Make sure, everything is in the correct order and self-contained.
- Whenever random functions are used, set their seed of 1 (unless stated otherwise) to make the experiments reproducible.
- You are free to reuse code from the lectures and exercises.
- It is well possible, that you will have to look up certain notions before you can answer a specific question. This is intended! Try to find reliable, valid sources.
- **Do NOT FORGET to add textual answers to EACH task. Code alone is not sufficient!**

Exercise 1. (Perceptron. – 5 points)
Given a perceptron with weights $(0.2, 0.3, 0.6, 0.6)$ and bias 0.15, compute the output for the tensor $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0.3 & 0.2 & 0.3 & 0.2 \end{pmatrix}$ of dimensions (dataset, features).

Exercise 2. (Modeling Data. – 17 points)
The online shop Yangtze offers $1,214,467$ different products. Registered users of the system have the option to assign tags to products. For that purpose, a user selects a product and assigns to it one or more tags. Each sequence of characters is eligible as a tag. Users can tag any product, and as many as they like. A user cannot use the same tag twice for the same product. However, different users can assign the same tag to the same product.

The product prices follow a power law distribution. The average price is \$2.55. At the moment, the platform has $1,989,345$ registered users, who have collaboratively added $56,892$ different tags ($1,199,438$ tags in total) to the products. The company wants to model the tagging data as a tensor with binary values.

In the following, consider only the data described above!

1. List the entities that the tensor should contain.
2. Describe a tensor to model the tagging data of Yangtze. Describe its dimensionality.
3. How many entries does the tensor have? How many are 1?
4. How does the tensor change when a user tags a product with a tag that has been used in Yangtze before?
5. How does the tensor change when a user tags a product with a tag that has not previously been used in Yangtze before?

Exercise 3. (SMOTE. – 10 points)

Familiarize yourself with the SMOTE [1] algorithm. In your own words, describe the use-case of SMOTE. Among others, address these points:

1. In which situations can it be useful (explain in general and provide three examples)?
2. What is its fundamental idea?
3. How is SMOTE different from oversampling with replacement?

Exercise 4. (Classification Experiment. – 58 points)

Create a Jupyter Notebook to solve the following machine learning task in Python. In this exercise, use plain Python, numpy, sklearn, etc. but no neural networks libraries yet (no PyTorch).

1. Create a Python class `Perceptron` that implements a perceptron with the following properties:
   - The class can be parametrized with the number of features.
   - The class uses a margin of 1 unless it is overridden.
   - The class has a method for computing the pre-activation values.
   - The class has a method to compute the predictions.
   - The class has a method for training, that takes the training data, the learning rate and the number of epochs as parameters.
   - The class allows controlling all random processes that it handles.
2. Load and arrange the dataset `portfolio_data_wise_2022.csv`. It has two features, `feature_1` and `feature_2`, and a target variable `target`. Process the data such that it is usable with the perceptron.
3. Describe the class distribution.
4. Plot the data.
5. Compare the performance of the perceptron with a margin of 1 in three different settings:
   a) trained directly on the plain data,
   b) trained using SMOTE (default configuration, only the oversampling algorithm, no undersampling of the majority class), and
   c) trained using random oversampling.

   For that purpose

- consider and implement relevant preprocessing steps,
- use a five-fold cross validation setting with shuffled, stratified folds to get a stable performance estimate,
- repeat each run 5 times with different random initializations of the neural network, use seeds 0 through 4 before creating a model, average the results over splits and repetitions,
- use 20 training epochs, and a learning rate of 0.1,
- evaluate performance using accuracy, balanced accuracy, and (averaged) confusion matrixes.

Hint: You must not optimize any hyper-parameters in this experiment.
Comment on your results.

Exercise 5. (Insights. – 8 points **+ 5 points bonus**)
With the results of Exercise 4 in mind:

1. Compare your expectations after Exercise 3 with the results of Exercise 4.
2. Visualize each scenario. For that purpose retrain a classifier on the full dataset (plain or modified as in Exercise 4). Plot the (modified) data together with the line representing the perceptron.
3. **Bonus:** Explain the situation. How did the application of SMOTE or random oversampling lead to such results. What would have to be changed to yield better results?

References

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.