FACHHOCHSCHULE KIEL
University of Applied Sciences

Interdisciplinary Master's Program Data Science

Prof. Dr. Stephan Doerfel SS 2022

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 (two 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) May 6, 2022**.

Rules and Hints:

- The following exercises guide you through a set of experiments. Tasks build upon the results of previous tasks. Therefore, they must be completed 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 $(0.1, 0.4, 0.6, 0.7)$ and bias 0.2, compute the output for the tensor $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0.1 & 0.2 & 0.1 & 0.2 \end{pmatrix}$ of dimensions (dataset, features).

Exercise 2. (Modelling Data. – 10 points)
The vacation platform JourneyAdvisor wants to apply deep learning in their recommender engine, that recommends points of interest to users based on their user account properties and previously visited places. The catalog of the platform contains $14,467$ points of interest. Users can check-in at such places using their phones. The platform has $1,989,345$ users. When users register, they enter their birthday, a payment method and their home address.

1. Propose a list of features, suitable for the recommendation task. Explain your choice!
2. Describe a tensor to model the data for JourneyAdvisor. Describe its dimensionality.
3. How many entries does the tensor have?

Exercise 3. (SMOTE. – 10 points)
Familiarize yourself with the SMOTE [1] algorithm. In your own words, describe the use-case of the 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. – 73 points)
Create a Jupyter Notebook to solve the following machine learning task in Python, using PyTorch (and other suitable libraries):

1. Load and arrange the dataset `portfolio_data_sose_2022.csv`. It has two features, `feature_1` and `feature_2`, and a target variable `target`.
2. Describe the class distribution.
3. Plot the data.
4. Create a simple (single layer) neural network with two output neurons, one for each of the two classes 0 and 1 (i.e. use a multiclass classification setup).
5. Compare the performance of the neural network in three different settings:
   a) trained directly on the plain training data,
   b) trained on training data modified using SMOTE (default configuration, only the oversampling algorithm, no undersampling of the majority class), and
   c) trained on the plain training data but with appropriate class weights in the loss function – select weights and explain your choice.

   For that purpose
   a) consider and implement relevant preprocessing steps,
   b) use a five-fold cross validation setting with shuffled, stratified folds to get a stable performance estimate,
   c) 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,
   d) use the Adam optimizer, $5,000$ training epochs, and a learning rate of 0.1,
   e) use cross entropy during training (select an appropriate PyTorch setup), and
   f) evaluate performance using balanced accuracy and normalized confusion matrixes.
6. You may reuse code fragments from the Jupyter Notebooks we discussed during the lectures.
7. *Interpret your results, explain your conclusions regarding SMOTE and class weights.*

You must not optimize any hyper-parameters in this experiment.

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.