Heinrich-Heine-Universität Düsseldorf
Computational Cell Biology
Alexander Kroll

Applications of Transformer Networks in Bio- and Cheminformatics

Worksheet 1

Submission Deadline:	22. April 2024, 11:59 pm
Discussion of solutions:	23. April 2024, 4:30 - 6:00 pm

Submission Instructions:

- Create a single Jupyter notebook for Exercises 1 and 2. This Jupyter notebook should contain your code and the results for these exercises.
- If you want to submit multiple files, zip them together. The uploaded file should have the following filename"lastname_studentID_worksheet1".
- Each submission must include an answer to Exercise 1.4. You can write the answer in your Jupyter notebook or in a text file.
- Submit your solutions by uploading the Jupyter notebook to https://uni-duesseldorf.sciebo.de/s/hCt1rTP23EeWmUC.

Excercise 1.1 *Data Preprocessing* (20 Points)

Each enzyme has a preferred pH at which it does its job most efficiently. We refer to the ideal pH as the enzyme's optimal pH. Above or below this pH, the activity of the enzyme may decrease, making it less effective. In this worksheet, we want to predict experimentally obtained optimal pH values.

(a) Download the datasets *phopt_training.fasta*, *phopt_validation.fasta*, and *phopt_testing.fasta* from https://zenodo.org/records/8011249. Load the datasets into Python: Create three Pandas DataFrames *train_df*, *val_df* and *test_df*, each with two columns "Sequence" and "optimal pH".
The "Sequence" column should contain the protein amino acid sequences as strings; the "optimal pH" column should contain the optimal pH as a float value.

(b) Plot histograms showing the distributions of sequence length and optimal pH in the training set.

Excercise 1.2 *Fitting and validating prediction models* (60 Points)

(a) Define a function to list all possible k-mers for a given k for the amino acids 'A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'V', 'W', 'Y'. The function should return all k-mer strings as a list.

(b) Given a list of k-mers, define a function that, for a protein sequence, returns an array containing the number of occurrences of each k-mer. The frequencies should be listed in the same order as the k-mers are stored in the list given as input.

(c) Calculate all 1-mers for the proteins in *train_df*, *val_df* and *test_df* from Exercise 1.1. Convert the absolute occurrences of these 1-mers to frequencies, i.e., divide by the total number of 1-mers in a sequence. Store the resulting vectors in a new column column "1-mer" for all dataframes.

(d) Repeat (c) for 2-mers and 3-mers.

(e) Fit linear regression models on the training set using k-mer frequencies as model inputs (k=1,2,3) to predict the optimal pH of a protein. Validate the models on the validation set using the mean squared error and the coefficient of determination R^2. Report the best model based on the performance on the validation set.

(f) Before training the linear regression models, write a function that considers only the n most frequent k-mers in the training set.

(g) Train one model for each combination of $k = 1, 2, 3$ and $n = 10, 20, 50, 400$. Select the model with the best R^2 on the validation set and evaluate the model's performance on the test set.

Excercise 1.3 *Using HPC* (20 Points)

For this exercise, follow the instructions on the "2_3 Using_HPC.pdf" slides. Submit all Python scripts and Bash scripts required to perform the following tasks, and any log files resulting from performing the tasks on the HPC.

(a) Submit a bash script that installs torch (version 2.0.0 with CUDA 11.7 enabled) for your Python version 3.8.3 on the HPC.

(b) Submit a bash script that runs a Python script that imports torch and prints out its version.

(c) Check if torch can find a GPU: Change the requirements in your bash script from *#PBS -l select=1:ncpus=1:mem=8GB* to
#PBS -l select=1:ncpus=1:ngpus=1:mem=8GB,
and add the command *module load CUDA/11.7.1* right after *module load Python/3.8.3* in your bash script. Using this bash script, run a Python script that checks whether a GPU is available.

Excercise 1.4 *Use of LLMs* (0 Points)

State for which exercise you have used LLMs (large language models) such as ChatGPT or GitHub Copilot. State which tools you have used and for which steps. This answer does not influence how many points you receive for your submission.