Heinrich-Heine-Universität Düsseldorf
Computational Cell Biology
Alexander Kroll

Applications of Transformer Networks in Bio- and Cheminformatics

Worksheet 6

Submission Deadline:	03. June 2024, 11:59 pm
Discussion of solutions:	04. June 2024, 4:30 - 6:00 pm

Submission Instructions:

- Upload a Jupyter notebook with your solutions to the exercises that you solved on your own PC. Upload the Python scripts and log files for any code that could not be executed in Jupyter Notebooks or was executed on the HPC.
- If you are submitting multiple files, zip them together.
- Submit your solutions by uploading the file to https://uni-duesseldorf.sciebo.de/s/hCt1rTP23EeWmUC. The uploaded file should have the following filename: "lastname_studentID_worksheet6".

Excercise 6.1 *Predicting Lipophilicity for small molecule drugs* (100 Points)

Lipophilicity measures how well a drug can dissolve in fatty substances, such as oils or fats. Drugs with high lipophilicity tend to break down quickly, don't dissolve well in water, are used up quickly by the body, and are not absorbed effectively. In this exercise, we will use information about the small molecule drugs to predict their lipophilicity. For this exercise, you will need to download the .csv files in *data/worksheet6*.

(a) Compute RDKit fingerprints for all small molecule SMILES strings. Then perform hyperparameter optimization as described in exercise 5.1(b) and validate the best model as described in exercise 5.2(c).

(b) The Python package *chemprop* contains an implementation of the D-MPNN and allows an easy training of the model. Use *chemprop* version 1 (e.g. version 1.6) and follow the instructions on this GitHub page https://github.com/chemprop/chemprop-v1-old-branches to train a D-MPNN on the training set using the default hyperparameters.

Validate the trained model on the test set, i.e. use the trained D-MPNN model to make predictions on the test set and calculate the MSE and R^2.

P Please note that you want to train on the whole training set, but by default *chemprop* splits the provided dataset into training, validation and test sets if you don't provide your own validation and test set.

(c) Compute the ChemBERTa-2 representations for all SMILES strings, i.e. compute for each SMILES string the element-wise mean of all token representations from the last layer of the pre-trained ChemBERTa-2 model. You can get all token representations from the last layer for a single SMILES string with the following code:

```
from transformers import AutoTokenizer
from transformers import AutoModelForMaskedLM
tokenizer = AutoTokenizer.from_pretrained("
   DeepChem/ChemBERTa-10M-MTR")
model = AutoModelForMaskedLM.from_pretrained("
   DeepChem/ChemBERTa-10M-MTR")
smiles = "CCO"
inputs = tokenizer(smiles, return_tensors="pt")
outputs = model(**inputs)[0]
```

Before computing the element-wise mean, you need to find out where the special tokens are in the input sequence, because you don't want to include them when computing the mean over all SMILES token representations.

Repeat Exercise 1(a), but with the ChemBERTa-2 representations instead of the RDKit fingerprints.

(d) Fine-tune the ChemBERTa-2 Transformer network for this prediction task; this will be very similar to fine-tuning the ESM-2 model for pH prediction in Exercise 4.3.

- You need to implement a feed-forward neural network (for lipophilicity prediction) on top of the updated classification token. By default, the ChemBERTa-2 model contains a classification token at the first position (index 0) in the input sequence.
- When generating input batches, you must ensure that you truncate input sequences that are too long and pad sequences that are too short. You can achieve this with the following settings for the tokenizer: *tokenizer(batch, padding=True, truncation=True)*
- Fine-tune the ESM-2 model for optimal pH prediction for at least 5 epochs on the training set. Set the loss function to mean squared error, the optimizer to the Adam optimizer, and the learning rate to 10^{-5}.
- After training, validate the prediction model on the test set, i.e., compute the MSE and R^2 for the test set.

Excercise 6.2 *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.