Heinrich-Heine-Universität Düsseldorf
Computational Cell Biology
Alexander Kroll

Applications of Transformer Networks in Bio- and Cheminformatics

Worksheet 2

Submission Deadline:	29. April 2024, 11:59 pm
Discussion of solutions:	30. April 2024, 4:30 - 6:00 pm

Submission Instructions:

- Create a single Jupyter notebook for all exercises. This Jupyter notebook should contain your code and results. Each submission must include an answer to Exercise 2.4.
- Submit your solutions by uploading the Jupyter notebook to https://uni-duesseldorf.sciebo.de/s/hCt1rTP23EeWmUC. The uploaded file should have the following filename "lastname_studentID_worksheet2".

Excercise 2.1 *Sinusoidal positional encodings* (40 Points)

Sinusoidal positional encodings are defined as follows

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right),$$
$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right),$$

where pos is the position of a token in the input sequence, d is the dimension of the position encoding vector, and $2i$ and $2i+1$ are the indices of the entries of the vector. This definition gives you a vector of dimension d for each position pos.

(a) Implement a function that computes sinusoidal positional encodings. This function should take a position pos and a dimension d as input and return a d-dimensional array with the sinusoidal positional encoding.

(b) For $d = 100$, compute sinusoidal positional encodings for positions 1 to 200. Plot the entries of the 200 resulting vectors in a single plot, using different colors for different values. The y-axis should represent the encoding dimensions and the x-axis should represent the position. Similar values should be represented by similar colors.

(c) Using the positional encodings calculated in (b), compute the dot product for the encoding vector at position 100 with all 200 positional encoding vectors. Plot the results of all 200 dot products in a scatter plot, where the x-axis displays the positions and the y-axis the results of the dot products. Do you see a pattern? What does this pattern mean for the results of the attention function?

Excercise 2.2 *Separating positional and token type information* (30 Points)

To encode the input sequence, Transformer Networks divide the input sequence into tokens. For each token, we combine information about the token type and its position by simply adding a token embedding and a position vector. Does this lead to a loss of information, or can we still get both information (token type and position) from the resulting vector?

To answer this question, I generated a dataset with 2 500 input embeddings of dimension 1 280 using the trained protein Transformer Network ESM-1b. Each embedding encodes an amino acid and its position in the input sequence. The input embeddings are stored in the file *data/worksheet2/embeddings.npy*, the encoded positions for all embeddings are stored in *data/worksheet2/labels_pos.npy*, and the encoded amino acids are stored in *data/worksheet2/labels_AA.npy* (represented by numbers from 0 to 19).

(a) Fit a classification model, for example a logistic regression model, to predict the amino acid type from the 1280-dimensional embeddings. First, randomly shuffle the data set and split it into 80% training and 20% test data. Calculate the accuracy on the test set.

(b) Fit a classification model, for example a logistic regression model, to predict the encoded position from the 1280-dimensional embeddings. First, randomly shuffle the data set and split it into 80% training and 20% test data. Calculate the accuracy on the test set.

Excercise 2.3 *Visualizing attention scores* (30 Points)

The file *data/worksheet2/attention_head_max_P00644.npy* contains the attention scores for a trained 33-layer protein transformer network (ESM-1b) for the protein P00644. For each of the 33 layers, there is a matrix showing the pairwise attention scores for all amino acids in the P00644 protein sequence. Each row of the attention matrix corresponds to the attention scores of one amino acid in the input sequence: That means the row shows the attention scores compared to all other tokens in the sequence that are required to update the amino acid representation.

(a) Plot all 33 attention matrices by representing the entries of the matrices through colors (similar as in this blog post https://deepfrench.gitlab.io/deep-learning-project/#Visualizing-the-Attention).

(b) Do you see any interesting patterns? How do you interpret them?

Excercise 2.4 *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.