Homework 2: Sequence Models

COMSW 4995 - Deep Learning
Instructor - *Iddo Drori*
Assignment CAs - *Robert Kwiatkowski, Isht Dwivedi*

September 27, 2018

The goal of this homework is to implement a sequence model using TensorFlow/Keras to predict protein secondary structure.

The homework should be done in pairs, and will be evaluated using an in-class Kaggle competition and based on submitted code. The homework is worth 10 points.

The homework is due **Thursday, Oct 11, 11:59 pm** and there will be no deadline extension.

The assignment should be done in Python 3 using Tensorflow / Keras. You may use Jupyter Notebooks or Google Colaboratory (colab.research.google.com) to do this assignment. This will let you and your teammate collaborate on one notebook and will allow you to use a GPU.

1 Protein secondary structure prediction

- **Input of your model:** A protein is represented as a sequence of amino acids. Each amino acid is represented by one of 21 characters. Different proteins may have different amino acid sequence lengths. A sequence of amino acids is the input to your model.
- **Output of your model:** Q8 protein secondary structure is a sequence of the same length as the corresponding amino acid sequence over an alphabet of 8 characters (G,H,I,T,E,B,S,-). Your model should output a predicted sequence with the same length as the input sequence.

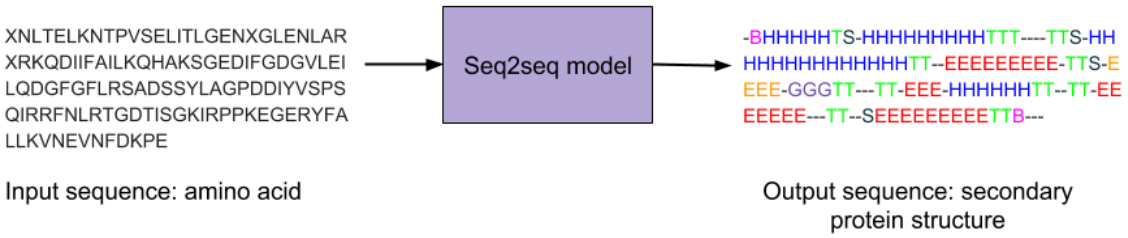

Figure 1: Input and output of sequence model for protein secondary structure prediction.

1.1 Dataset

The dataset consists of 6,685 proteins. You will be given a train-test split. For each protein you are given 4 fields: the protein name, sequence length, amino acid sequence, and Q8 secondary structure sequence. You are only allowed to use the data available as part of the homework, and no external data can be used.

1.2 Implementing the Network

A baseline implementation for the task is a bidirectional LSTM model with the input tokenized using n-grams, which is provided in the corresponding Python file. You may optimize the network architecture, optimization method, and regularization parameters. You may use attention, transformers.

1.3 In-Class Kaggle Competition

1.3.1 Kaggle Sign-Up

In Kaggle first form a team and then submit your results, see `https://www.kaggle.com/wiki/FormingATeam`. Team names should be a concatenation of the UNI's: abc1234def5678 Submissions on the leaderboard should be titled with the UNIs of the team members.

1.3.2 Kaggle Submission

In order to join the competition you will have to navigate to the invite here:
`https://www.kaggle.com/t/7e05e920ce0246598b3bf930ac4792a2` and the site for the competition will be here:
`https://www.kaggle.com/c/dl-2018-hw2`.
You can submit at most 2 times every 24 hours on Kaggle.

1.3.3 Kaggle Scoring

The public leader board will only show results on some part of the test set sequences. This is to prevent tuning on the test set. Kaggle will expect submissions to be a .CSV file with formats "id (String), expected (String)".

1.4 Code Submission

Students are required to submit code corresponding to their best model along with a pdf report. Zip all files and upload this zipped file with name "HW2-UNI.zip" on Courseworks. Both members of each group should submit the assignment. The pdf report should include the following:

1. Name and UNI of both members of your group.
2. The architecture of your network, and the training regime used (optimizer used, number of epochs trained, learning rate decay policy, regularization, etc.)

3. Instructions that can be used to run your code and replicate best results submitted on Kaggle.

4. The accuracy on the test set using your best model using baseline accuracy function (available in baseline code).

5. You need to do 10 fold cross validation on the train set and report the average validation accuracy(computed by the baseline accuracy function) over all the 10 folds.

1.5 Grading

Your final grade is a combination of your final accuracy and your position on the private kaggle leaderboard.