CS 561 – Project #3
Neural Networks for Gene Regulation

Language: Python 3 and Keras/TensorFlow
System: Duke DCC compute cluster
Data: On DCC at: /hpc/group/cscbb561s23/projects/project3

Preparation for the Assignment:

Before doing this assignment, you will need to set up a specific conda environment on DCC. If you have not installed conda, do so using these two commands in your home directory:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
sh Miniconda3-latest-Linux-x86_64.sh
```

Then create a TensorFlow version 2 environment (do this in your home directory):

```
conda create -p TF2
conda activate ~/TF2
PS1="($(basename $CONDA_DEFAULT_ENV)) \[\e[7m\][\h \W]\[\e[27m\] "
conda install pip
pip install keras-nlp
pip install pandas
pip install scikit-learn
```

Once these commands have finished, you will need to obtain an interactive GPU node and activate the new environment:

```
srun --x11 -p gpu-common --mem=1000 --gres=gpu:1 --exclusive --pty bash -i
conda activate ~/TF2
```

You will need to perform the last two commands every time you log in to DCC to work on this assignment. You may need to increase the requested memory via the --mem option.

Part 1 (50 points): Implement and Test Neural Network Models for Binary Classification

For this assignment, you will implement a series of neural networks, one for each of four simulated data sets. The task is binary classification. The four data sets are:

- sim1
- sim2
- sim6
- sim7

Each of these data sets is contained in a separate subdirectory, and contains three fasta files: one for training, one for validation, and one for testing. You will use the training and validation files during the model fitting step, using this command:

```
model.fit(X_train, Y_train, verbose=1,
       validation_data=(X_valid, Y_valid),batch_size=128, epochs=100,
       callbacks=[EarlyStopping(patience=10, monitor="val_loss",
       restore_best_weights=True), History()])
```

You will need to write code to load the DNA sequences and convert them to 1-hot encoding.

The class of each sequence is annotated on the defline:

```
>1 /class=1
ATGAGAGGGAAAGGGTCAGTTTGTACCAATTGTTCAAGTGGGGCATTATGACGGAGTATG
CTGACCTATACAACTTTAATTCCCGTAGTCCACCCGGCTGACCGAAGGCTCTTAAGTGAT
TACACTCAAAAACCGAAAAGTAGGACCAGACTCAATCGCCTATCCACGAAGAGTCCGGCC
ATGCATGTTAGTAGACATGAAACTAGGTATGCGATCCAAGATGGCGTTGTTGTCTCGGGT
GTAGTCCCC
```

The two classes are 1 and 0, where 1 means the sequence is an enhancer, and 0 means the sequence is not an enhancer. Perform binary classification and evaluate the accuracy of your model on the test set via this command:

```
pred = model.predict(X_test, batch_size=128)
```

That command will give you the predictions; you can compute an accuracy score by dividing the number of correct classifications by the total number of sequences in the test set. For the report (part 2—see below), you will also need to plot ROC curves and report AUC values of each model on each test set.

You may use any neural net architecture, with any number and type(s) of layers, including dense, convolutional, recurrent (LSTM / GRU / Bidirectional), as well as normalization, pooling, dropout, and activation layers. You may use residual skip connections and/or dilation if you wish. You may combine different types of layers in one model.

For each of the four data sets, find the best model architecture—i.e., the one that produces the highest classification accuracy or AUC.

Submit your code on Sakai, after combining all source code files into a single gzipped tarball (i.e., source.tgz or source.tar.gz). Submit your report as a separate file.

Part 2 (50 points): Write a Report Summarizing Results and Conclusions

Once you have completed Part 1, write a report describing in detail (1) the models developed for each of the four data sets, (2) the training and testing procedures used, and (3) the results. You may include figures and tables. Do not include references. Your report should not exceed three pages in length (font size at least 11 point), unless you include a fourth page for Part 3 (see below). Submit your report on Sakai in PDF format.

Part 3 (Extra Credit: maximum 50 points): Quantitative Prediction of Real Enhancer Data

As an optional exercise for extra credit, develop a model to predict, quantitatively, the activation values of real enhancer sequences in a Massively Parallel Reporter Assay (MPRA). The data are located in the subdirectory "MPRA". There are again training, validation, and test files. However, because this is not a binary classification task, there is no class on the defline (in the FASTA file). Instead, the activation value of each enhancer is given in the first two columns of the associated "activity" file: the first column (Dev_log2_enrichment) is for developmental genes and the second column (Hk_log2_enrichment) is for housekeeping genes. The activation values are given in the same order as the sequences in the FASTA files. You will make a neural network that takes sequences as inputs, and produces two outputs: one to predict the MPRA activation value for Dev, and one to predict activation for Hk. Evaluate the accuracy of your model via MSE (mean squared error). Report the accuracy of your model on Dev and Hk separately, and describe the architecture you used and why you chose that architecture. For this part, you may include a fourth page in your report describing the results of Part 3.

Additional Notes:

- All team members must contribute to the project.
- You must write the code from scratch—do not use code obtained from a third party, other than Keras, TensorFlow, and standard python modules (as well as SciPy, NumPy, Pandas, itertools, and pickle). You may use basic functions from sklearn—i.e., to perform basic tasks such as computing mean squared error or AUC. You may use any plotting software to create ROC curves or other plots if desired.
- In your report, for each model, clearly describe the architecture that worked best. You may describe the model architecture in words, or (optionally) via a clearly labeled figure that depicts all layers (including normalization, dropout, pooling, etc.).
- Your models can be different for the different data sets. You can also include a baseline for comparison (optionally)—for example, random guessing.
- Include a table in your report indicating the accuracy of each model on each data set, as well as an ROC curve and AUC values. You may combine the ROC curves into one plot, or include separate plots for each data set if you wish.
- Remember that your report should not exceed 3 pages in length, unless you included a fourth page for the extra credit excercize (Part 3).