Molecule Toxicity Prediction

1. Project Specification

Deep neural network has become a hot research topic in machine learning in recent years. Compared to other methods, deep learning have shown its advantages in handling large amount of data and achieving better performance. Based on the structure of the network and the neurons, there are many types of deep neural networks. Among them, the Deep Convolutional Neural Network is extremely popular in the area of image processing.

In this **individual** project, you are required to develop a deep neural network to classify on some bioinformatics data using **TensorFlow**. TensorFlow is an open source software library for numerical computation using data flow graphs. It has many convenient APIs for implementing deep neural networks, which makes the implementation more convenient.

2. Dataset

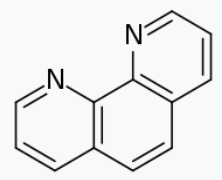
The dataset is about the toxicity of some small molecules. We provide two folders for you, one is the training data NR-ER-train (about 8k samples), and the other one is test data NR-ER-test (about hundreds samples). There are three files in each folder:

- names_labels.txt: each line contains a molecule's name and its label, where 0 means non-toxic and 1 means toxic.
- names_smiles.txt: each line is the molecule's name and its SMILES notation
- names_onehots.npy: there are two ndarray in this .npy file. One is the names of the molecules, and the other is the one hot representation of the molecule SMILES. You can read this file using the following code:

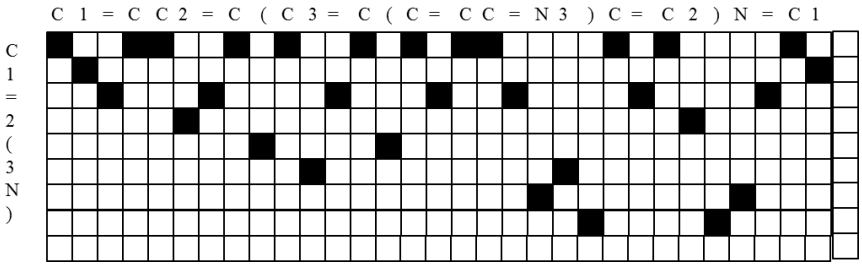
  dictionary = np.load("names_onehots.npy").item()

  dictionary["names"] is the ndarray of molecules's names
  dictionary["onehots"] is the ndarray of the SMILES.

SMILES is a linear notation to present the 2D structure of molecules with 1D ASCII string. For example, the 2D structure of phenanthroline is



And its SMILES notation is "C1=CC2=C(C3=C(C=CC=N3)C=C2)N=C1". The one hot format of SMILES is a 2D {0,1} matrix, where each column represents for a symbol in the SMILES notation of the current molecule, and each row is one ASCII character appeared in SMILES notations. The size of the 2D matrices is the length of the longest molecule SMILES * the number of ASCII symbols appeared in the dataset, which means we have zeros padded after short molecule SMILES. One at row i, col j means the corresponding symbol exist at position j for the SMILES. The one hot example for phenanthroline is:



The molecules in NR-ER-train and NR-ER-test do not overlap with each other. The data we use to mark your model is in a folder named NR-ER-score(hundreds samples), and you have no access to it. There are two files in the folder:

- names_smiles.txt
- names_onehots.npy

The format of these two files are exactly the same with those for training and testing, but the molecules are new.

3. Assignment requirements

1) This is an **individual assignment**.
2) Prediction task
   You are going to predict toxicity of the molecules based on their structures. The output of your model is the probability that the current molecule is toxic.
3) Data
   a. NR-ER-train
   b. NR-ER-test
   c. NR-ER-score (not accessible)
   You can train your model on the NR-ER-train and test its performance on NR-ER-test, or you can train your model with both of them.
4) Features
   You can either use the one hot of SMILES as features for molecules, or directly use the SMILES notation as you wish.
5) Model
   You can use any deep neural network architecture in this assignment to achieve good performance, e.g. convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM), etc.

6) Marking
   Your model will be tested on the NR-ER-score dataset. Your submission package will be put alongside the NR-ER-score folder, and it means that you have to use the relative path "../NR-ER-score/" in your submitted "test.py" file to access the marking data. Thus, in your test.py file, you need to first restore your model parameters and then test your model on the marking data ("../NR-ER-score/"). Your final score of this assignment depends on the AUC-ROC among your predictions and the true labels:

   $$\text{Score} = \text{AUC} * 6.$$

   PS: It is hard to get an AUC above 0.8
7) Output
   For the AUC, we will run your "test.py" file, which is expected to read the files described above in "../NR-ER-score/". You should try to predict the labels for the samples in NR-ER-score, and output your predictions (probabilities) into a file named "**labels.txt**". Each line in "labels.txt" file is probability for the corresponding sample.
8) Deep learning library
   TensorFlow only. Please do not use other libraries. Otherwise, you will get zero mark for this assignment.
9) Programming language
   The only supported language for this assignment is **python2.7**. We do not accept python3.x program.

4. Submission requirements

1) Submission list
   a. Source file for training. Name it as "train.py"
   b. Source file for recovering your network model. Name it as "test.py"
   c. TensorFlow generated files, which store the values of your variables.
   d. Any other files that help your programs to work, such as preprocessing files, format-converting files.
2) Submission Package

Put everything in the submission list above into a folder, name the folder as your **student id**, zip the folder WITHOUT encryption. Submit the **zip file** into our submission system.

Do not add any of the data in training and test folder in your package, because we will check your model solely on the score folder, which is already put in the server.

5. Important Points

To make this project fair and meaningful, there are some other points you **MUST** follows:

1) The size of your "test.py" should not exceed 4k Bytes (to prevent you remember the whole dataset in your test.py)
2) The maximum number of lines in your "test.py" that do not in your "train.py" is 20 (to make sure that "test.py" is only used to recover the trained network)
3) The time limits to run your "test.py" is 60s
4) The size of the whole zip file should be less than 200 MB (just keep the trainable variables, not the whole variables)
5) Plagiarism will be SERIOUSLY punished (ZERO mark plus reporting to department)
6) Late submission will NOT be entertained according to our submission system settings

6. Late Submission

No. of Days Late	Marks Deduction
1	10%
2	30%
3	60%
4 or above	100%