Tox21_AR

Information

Total 9362 molecules (graphs) in the dataset.

5617 molecules (graphs) in the training set.

2808 molecules (graphs) in the testing set.

937 molecules (graphs) in the scoring set.

The molecules are represented as graphs (with neighboring matrices), instead of Molecular Formula.

Files

1. <train/test/score>_labels.csv

A csv file which contains the graph index and corresponding label, where 0 means non-toxic and 1 means toxic.

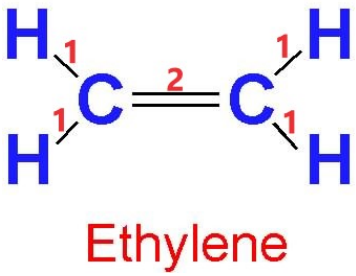
2. <train/test/score>_graph_size.csv

A csv file which contains the graph index and graph size (the number of nodes in this graph).

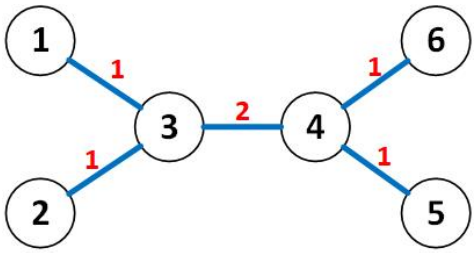
3. <train/test/score>_graphs.csv

A csv file which contains the neighboring matrices of graphs in <train/test/score> set. The largest molecule (graph) has 132 atoms (nodes). So each graph in this file is represented as a 132x132 matrix. There are three types of edges in these graphs. If two nodes are not connected, the value in the matrix is 0. If they are connected by TYPE-1 edge, the corresponding value in the neighboring matrix is 1. If they are connected by TYPE-2 edge, the corresponding value in the neighboring matrix is 2. And if connected by TYPE-3 edge, the value is 3.

Here is an example: The molecule formulation of Ethylene is C2H4. The structure is:



There are 2 types of edges in this molecule. The corresponding graph representation is:



The matrix representation is:

	1	2	3	4	5	6	...	...	132
1	0	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0
3	1	1	0	2	0	0	0	0	0
4	0	0	2	0	1	1	0	0	0
5	0	0	0	1	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0
...	0	0	0	0	0	0	0	0	0
...	0	0	0	0	0	0	0	0	0
132	0	0	0	0	0	0	0	0	0

Each graph is represented as a 132x132 matrix. In this example there are only 6 nodes, therefore only the first 6 rows and the first 6 columns have non-zero values. If two nodes are not connected, the corresponding value is 0. If they are connected, the value represents the edge type. The matrix representation is symmetrical, i.e. the graph is an undirected graph.

4. <train/test/score>_nodes.csv

Each line in this csv file is the feature (atomic feature) of the node in the graph. By default, each graph has 132 nodes. If the number of the nodes in one graph is less then 132, i.e. N <= 132, then the features of the (132-N) nodes are set as 0. Features of the N nodes are their real features.

DNN

1. You can try to use traditional neural networks (e.g. TensorFlow MNIST example code) to solve this problem. Maybe you only need the neighboring matrix.
2. You can try some new and popular neural networks, e.g. Graph Convolutional Neural Networks. You can use the node features provided in the dataset.
3. You can implement any data processing (preprocessing or postprocessing) algorithms in your code.

ATTENTION

1. One person per project.

2. Auto-submission will be provided.
3. Repeated submission allowed.
4. Python 3.
5. TensorFlow 1.5.0 and NumPy only.
6. Stick to the output format and the file names specified.
7. The time limits to run your `test.py` is 60s
8. **Plagiarism will be SERIOUSLY punished (ZERO mark plus reporting to department)**.

Submission

1. Submission list
   - Source file for training. Name it as `train.py`.
   - Source file for recovering your network model. Name it as `test.py`.
   - TensorFlow generated files, which store your trained model, e.g. ckpt files.
   - Any other files that help your programs to work, such as preprocessing files, format-converting files.
2. Submission packaging

   Put everything in the submission list above into a folder, name the folder as your student id, zip the folder **WITHOUT** encryption, also name the zip file as `your_student_id.zip`, e.g. `1155088239.zip`. Submit the zip file to our submission system. The size of the whole zip file should be less than 200 MB.
3. **DO NOT** add any of the data in training and test folder in your zip file, because we will grade your model solely on the score folder, which is already put in the server.
4. You model will be tested on the Tox21_AR dataset. Your submitted folder will be extracted and put alongside the Tox21_AR folder, and it means that you must use the relative path `../Tox21_AR/` in your submitted `test.py` file to access the marking data (e.g. `../Tox21_AR/score_graph_size.csv`, `../Tox21_AR/score_graphs.csv`, and `../Tox21_AR/score_nodes.csv`). These marking data are in the same format with the data we have provided to you. In your `test.py` file, you need to first restore your model parameters and then test your model on the marking data files in `../Tox21_AR/`. You should output your predictions into a file named `labels.txt`, which should be in the same directory as `test.py`. Each line in `labels.txt` file is {0, 1} toxicity label for the corresponding sample.
5. You should load your model (e.g. ckpt files) by yourself in your `test.py`. You must use the relative path in your submitted `test.py` file to access the your ckpt files.
6. Your final score of this assignment will depend on the Balanced Accuracy among your predictions and the true labels:

$$balance\_accuary = \frac{1}{2} \times (\frac{TP}{TP+FN} + \frac{TN}{TN+FP}).$$

Late Submission

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

Q&A

1. Can I use `tf.keras`?

   Of course, it's a TensorFlow API.

2. What are the commands used for grading?

   Grading program is on a private server, and it's not open for you. But you can submit your program to test your code.

3. Dataset is unbalanced?

   The dataset provided to you is unbalanced, also that's a common situation in practical applications. You can sample some points in each training step from the provided dataset to make it balanced and robust. You can also try other solutions. You can implement these in your preprocessing file. Or you can directly ignore these if your model is robust enough. The final score is based on your Balanced Accuracy.

4. How to implement matrix multiplication algorithms?

   There are some basic matrix multiplication algorithms implemented in NumPy.

5. Can I use TensorFlow 2.0?

   NO. Only TensorFlow 1.5.0 is installed on our grading server.