Installation Instructions (with Anaconda)

**CPU / GPU Version : (change 10.2 to 11.3 if you have CUDA version 11.3)**

1. conda create -n graphnn_env
2. conda activate graphnn_env
3. conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
4. conda install matplotlib scikit-learn jupyter requests dgl-cuda10.2 -c dglteam

**CPU only version :**

1. conda create -n graphnn_env
2. conda activate graphnn_env
3. conda install pytorch torchvision cpuonly -c pytorch
4. conda install matplotlib scikit-learn jupyter requests dgl -c dglteam
Your job as a Deep Learning Engineer …

We will use the Protein-Protein Interaction (PPI) network dataset which includes:

- 20 graphs for training
- 2 graphs for testing

One graph of the PPI dataset has on average 2372 nodes.

Each node has:

- 50 features: positional gene sets / motif gene / immunological signatures …
- 121 labels: gene ontology sets (way to classify gene products like proteins).

**Node (multi-label) classification task (supervised learning):**

For a given PPI graph, being able to predict the correct node’s labels.
Graph-NN

features

\[
\begin{bmatrix}
 f_1 \\
 f_2 \\
 \vdots \\
 f_{50}
\end{bmatrix}
\]

Labels

\[
\begin{bmatrix}
 L_1 \\
 L_2 \\
 \vdots \\
 L_{121}
\end{bmatrix}
\]
Homework tasks (grade: /20)

1. Improve the given model i.e the code provided in the course materials (8/20)
   (Hint: modify the architecture and use a Graph Attention Network)

2. Produce a short report including: (12/20)
   - A diagram of the architecture you’re using (with shape information!) (6/12)
   - A small paragraph in which you interpret your results (6/12):
     - What kind of architectures did you try?
     - What your result means (f1-score? - plot of f1-score wrt epochs and comparison with original provided model)?
     - Why would Attention Network perform better than GraphConv?
Homework tasks

To summarize, what you have to return:

- The modified code (`train_ppi_baseline.py`)
- The weights of the model (`model_state.pth`)
- The report in PDF format.

⚠ Do not change the signature of the `train()` and `test()` functions (or otherwise you’ll get 0/8 at the first task)!

Before submitting make sure the following runs correctly in your conda environment:

```
python3 train_ppi_baseline.py --mode test
```
Examples of diagram (perfectible)

Multi-head attention
(shape information is missing)

ResNeXt
(a good legend is missing)

Ideas of diagram; good legend + shape information needed
Useful links

- If you are curious about the dataset and some applications:

- Graph Attention Network: https://arxiv.org/abs/1710.10903

- The DGL library: https://github.com/dmlc/dgl

- You can take inspiration (but don’t use it directly) from pytorch geometric: