# Deep Learning 2022


Homework 2

Due date : 2022/05/06 23:55:00

- High-level API is forbidden in this homework, such as Keras, slim, TFLearn, Hugginface, Allennlp etc.
- Homework submission - Please zip each of your source code and report into a single compressed file and name the file using this format: HW2_StudentID.zip (rar, 7z, tar.gz, ... etc are not acceptable)
- If you have any problem about implementation, DO NOT directly upload your code on the E3 discussion

## 1 News Document Classification

To start this problem, some preliminary steps need to be conducted first:

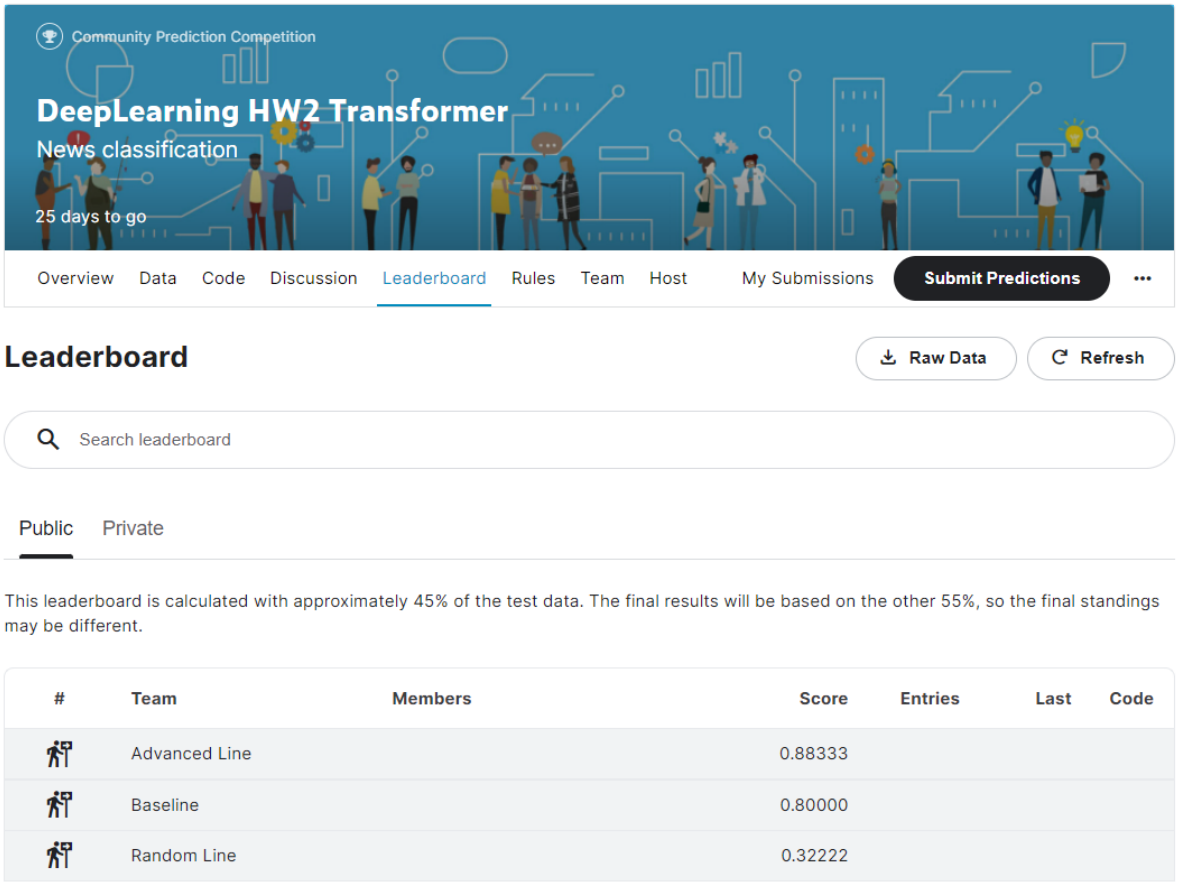
1. **Join the in-class competition on Kaggle (Here)**
   (or search DeepLearning HW2 Transformer in Kaggle)
2. **Change the team name by using your student id number**
3. **Download the data and check for the description**

In this problem, you are given a csv file (train.csv) gathered from more than 2000 news sources by ComeToMyHead, contained the corresponding Label, Title and Content. You are required to implement a Transformer to correctly classify the news document in (test.csv) and submit the classification result to the Kaggle competition.

### 1.1 Data Preprocessing (10%)

In this homework, you cannot directly use high-level API to help you process the text data. You are asked to convert a text string into a list of integers by these package: FastText, TorchText, NLTK, spaCy or Gensim.

1. How do you choose the tokenizer for this task? Could we use the white space to tokenize the text? What about use the complicated tokenizer instead? Make some discussion. (5%) (You might want to explain it by showing the performance comparison with different tokenizer. If you are not familiar with tokenizer, check (`https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/`))
2. Why we need the special tokens like ⟨pad⟩, ⟨unk⟩? (2%)
3. Briefly explain how your procedure to handle the text data. (3%) (e.g. Which tokenizer do you choose? Why? ; what's your min_count setting, etc.)




## 1.2 Transformer (50%)

Build the Transformer to solve this task and answer the following questions. (Hint: You might want to read this tutorial first (`https://pytorch.org/tutorials/beginner/transformer_tutorial.html`))

1. Pass the Baseline on the Kaggle in-class competition (Public). (20%)
2. Please show the attention map in the last layer of Transformer of some examples to find out which token more likely affects the classification result. Do you think the results make sense? Please make some discussion. (5%)


Figure 1: example for attention map

3. Discuss the model structure or hyperparameter setting in your design. (5%) (e.g. hyperparameters of transformer: d_model, nhead, d_hid , nlayers, dropout, etc. Why do you choose this setting?)

4. Kaggle Challenger Award! After Kaggle competition, the private leaderboard will reflect the final standings. In private leaderboard, you will be finally recognized as
   - Challenger (10%): Pass the Advanced Line
   - Second prize (5%): Pass the Baseline
   - Consolation prize (5%): Join Kaggle competition and pass Random Line

### 1.3 Hint

You might want to follow these steps to start your work:

1. Tokenize the given text data with some off-the-shelf software.
2. Build the vocabulary (like the dictionary object in python) to map the token into some unique ID.
3. Select the pretrained embedding (Glove, Fasttext, Word2vec) as the initialization of your embedding layer. (Not necessary, but recommended)
4. Construct your transformer model and finally end up with some simple feed forward module.
5. Choose the suitable optimizer (Adam might be not suitable) and activation function (ReLU might be not suitable. Try Tanh or Swish?)
6. Try some tricks like learning rate scheduler and packed_padding_sequence.
7. Check some tutorial such as Here.

## 2 VAE for Image Reconstruction

It is important to study the paper on variational autoencoder (VAE) before the implementation. You need to implement VAE with two kinds of decoder where different variational posteriors are calculated. The decoder should be selected according to data type (as given in the Appendix C in VAE paper). There are two datasets. The first one is the TibetanMNIST (gray images, TibetanMNIST.npz) and the other one is the animation faces (RGB images, anime-faces.zip). Some examples are shown below.

Figure 2: Left: TibetanMNIST, Right: animation faces

This task is to build a VAE to reconstruct images. The following steps should be implemented individually for these two datasets. Some examples are shown in each step.

1. Implement VAE and show the learning curve and some reconstructed samples like the given examples. (10%)
   Hint:
   - Convert the gray images into binary images first
   - Variational posteriors are Bernoulli for binary data and Gaussian for real value data

Figure 3: Left : Real samples in dataset, Right : Reconstruction samples using VAE

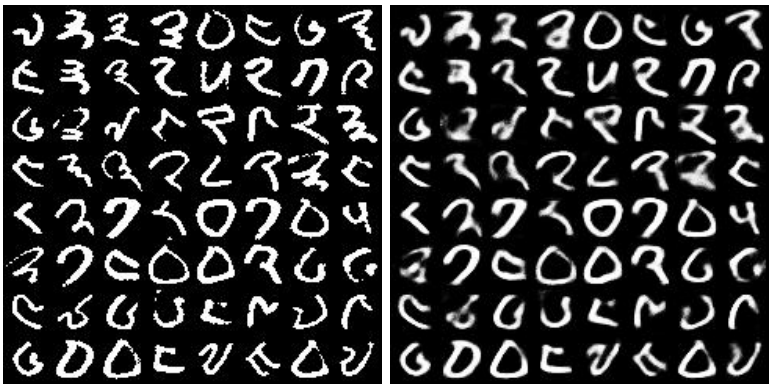


Figure 4: Left : Real samples in dataset, Right : Reconstruction samples using VAE

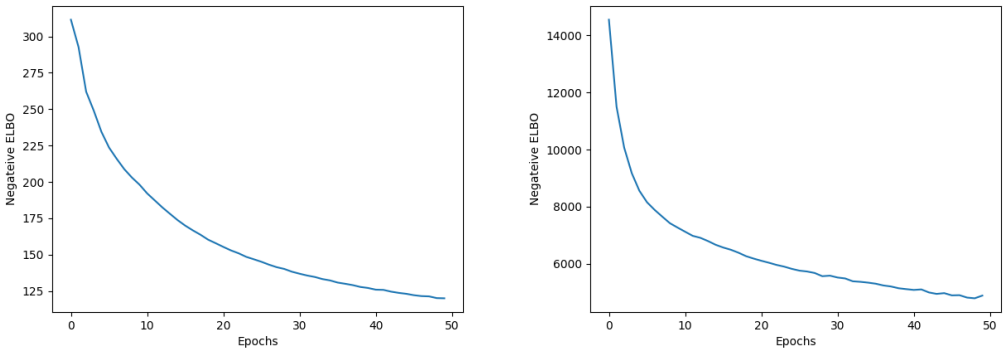



Figure 5: Left : learning curve of TibetanMNIST, Right : learning curve of animation faces

2. Sample the prior $p(\mathbf{z})$ and use the latent codes $\mathbf{z}$ to synthesize some examples when your model is well-trained. (10%)

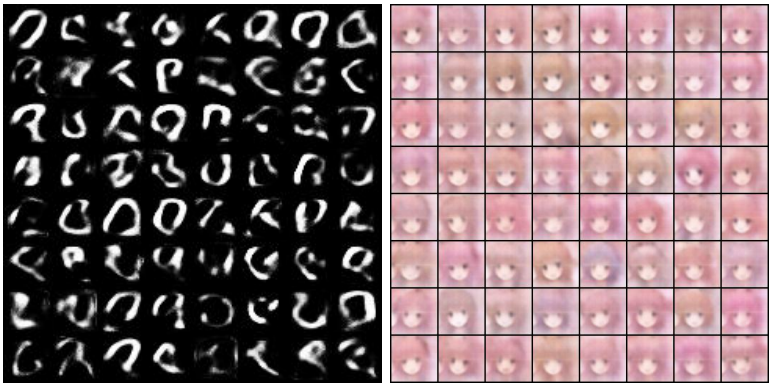


Figure 6: Left : synthesized images of TibetanMNIST, Right : synthesized images of animation faces

3. Show the synthesized images based on the interpolation of two latent codes $\mathbf{z}$ between two real samples. (10%)

Figure 7: Left : synthesized images of TibetanMNIST, Right : synthesized images of animation faces

4. A famous issue with VAE is the posterior collapse, in which the latent posterior $q_\phi(z|x)$ collapses towards the prior $p(z)$. Multiply the Kullback-Leiblier (KL) term with a scale $\lambda$ and tune $\lambda$ (e.g. $\lambda = 0$ and $\lambda = 100$). Repeat steps 1, 2, 3 and show the results. After training, select an image as input and observe the mean and variance of its latent posterior $q_\phi(z|x)$. Finally, do some analyses with above results. (10%)

## 3 Rules

- Please name the assignment as hw2_StudentID.zip (e.g. hw2_0123456.zip).
- In your submission, it needs to contain four files.
  - three **.py** files which contain the codes for each question in homework (e.q. 0123456_transformer.py , 0123456_vae_mnist.py , 0123456_vae_anime.py).
  - one **.pdf** file which is the report that contains your description for this homework.
- Only Python implementation is acceptable.
- Only Pytorch library is acceptable for model implementation.
- Don't use high level API (e.g. Keras, slim, TFLearn, Hugginface,Allennlp etc).
- DO NOT PLAGIARISM. (We will check program similarity score.)