ASSIGNMENT 2

KAGGLE PART

Name: TAM Ka Ho • SID: 1155175983 • Deadline: 19 Mar 2024

1. Introduction

Protein pattern recognition have always been a topic of interest in the fields of computational biology. In this assignment, by training a simple multi-layer perceptron (MLP) over the embeddings generated by the pre-trained ProtTrans transformer model, we are able to obtain a validation accuracy of 98.1%, public score macro F-score of 98.0% and private score macro F-score of 94.2%. For the procedures of using our code, please refer to the *readme.md* file attached.

2. Model Architecture

In this assignment, we first prepared the sequence embeddings by passing the retrieved protein sequences into the pre-trained ProtTrans *ProtT5-XL-UniRef50* model [1], [2]. The relevant code is in *transforms.py* and the embeddings are saved to the directory `cache`.

Next, we trained MLPs over the embeddings with cross-entropy loss and Adam optimizer with a learning rate of 0.001 and an exponential learning rate decay of gamma value 0.99. We tested over different configurations of MLP, including different sizes of hidden layers and different numbers of hidden layers. The final best models are then trained with both *train* and *validation* datasets to further improve the accuracy, the results are listed in Table 1:

Hidden Layer Sizes (***** means trained with both *train+val* datasets)	# Epochs	Accuracies in %		Accuracies in decimal	
		Train	Validation	Public Score	Private Score
[512, 256]*	150	100.0%	--	0.98042	**0.94161**
	200	100.0%	--	**0.98438**	0.9343
[512, 256]	200	100.0%	**98.0%**	**0.97865**	0.92916
[256, 256]	300	99.9%	**98.1%**	0.97162	0.93108
[512, 512]	300	100.0%	**98.1%**	0.97632	0.91528
[256]	1000	100.0%	97.6%	0.97031	0.91962
[512, 256, 128]	300	100.0%	**98.1%**	0.94419	0.9267
#trained with raw embeddings with shape (8, 1024) instead of the suggested mean of the first 7 raw ones.					
[512, 256]#	300	99.9%	97.5%	0.97052	0.9084
[512]#	200	100.0%	97.8%	0.97615	0.91899

Table 1 Performance Comparison for Models Trained

Table 1 also shows that a 2-layer MLP with hidden layer sizes of 512 and 256 performs the best after training for around 200 epochs. This model is then re-trained with the combined dataset of train and validation to further improve the accuracies.

We stopped the training at around 200-300 epochs as there is no significant improvements afterwards, and continue training would only decrease the accuracy, which can be seen in the train-validation loss history plot in Figure 1:

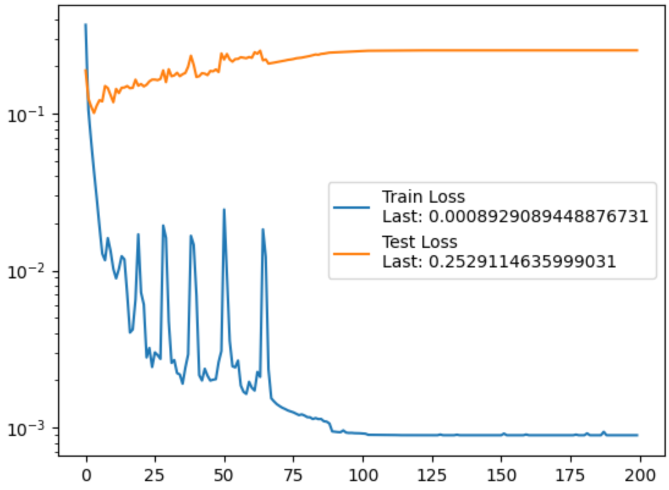

Figure 1 Loss History of the best-performing model trained on *train* dataset only

In our experiments with different models, we observed that fine-tuning the transformer model itself did not improve the results due to the small size of the dataset. Therefore, we instead cache the embeddings generated from ProtTrans and trained a simple MLP on it, which displayed a higher accuracy and significantly shorter training time. In this assignment, we did not have sufficient time to train an extra transformer model on the full embedding outputs from the pre-trained ProtTrans model, but we suspect that it could potentially further improve our accuracies.

Finally, note that the code for training and testing are separated from the main code and is placed in *libs.py*, while the model definition is placed in *models.py*. The class mapping dictionary and dataset loaders are placed in *datasets.py*.

3. References

[1] A. Elnaggar *et al*, "ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High-Performance Computing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, 2021, doi: 10.1109/TPAMI.2021.3095381.

[2] https://github.com/agemagician/ProtTrans