AIST4010 Foundation of Applied Deep Learning Kaggle Assignment 2 - Report

Name: Wong Shing Lok
SID: 1155156680

1 Implementation Details

1.1 Data Preprocessing

The *ProteinSequenceDataset* class in the code is used for loading and preprocessing the protein sequences. The sequences are loaded from a FASTA file and encoded using the ESM's alphabet. The sequences are grouped by label and a maximum number of samples per class can be specified to alleviate problems that the imbalanced dataset could bring.

1.2 Model Architecture

In this assignment, I mainly utilized the **Evolutionary Scale Modeling (ESM-2)** model[2]. The exact version of the model adopted in this assignment is **esm2_t6_8M_UR50D**, which is the smallest scale model among the pretrained ESM-2 variants, with 6 layers, 320 embedding dimensions and 8 milions parameters. It is a transformer model pre-trained on a large corpus of protein sequences. The model is used as a feature extractor, and its output is fed into a linear classifier to predict the class of the input sequence. The classifier is a simple linear layer with a number of output units equal to the number of classes. The pre trained model can be found on Github repository published by Facebook Research[1]

As for the fine-tuning, the **Low-Rank Adaptation (LoRA)** model was employed[1]. The detailed implementation is stated in section 1.4.

1.3 Training

The model was trained using the **cross-entropy** loss function. The **Adam** optimizer was used with a learning rate starting of 0.001. The model was trained for 15 epochs. The batch size was just between 2 to 4 due to the lack of GPU memory. The training was run multiple of times using the same epoch setting, but the learning rate was decreased by 10 times for each run. During training, the F1 score on the validation set is monitored, and the model parameters that achieve the best F1 score are saved.

1.4 Fine-tuning with Low-Rank Adaptation (LoRA)

To tailor the pre-trained ESM-2 model for ARG prediction, I applied Low-Rank Adaptation (LoRA)[1], focusing on the model's final classifier layer. LoRA introduces small, efficient updates to the model, enabling fine-tuning without extensively retraining the entire network. The LoRA implementation was adopted from Github repository[2] published by Microsoft.

The orginal classifier adopted during training was replaced with a **lora.Linear** layer, set with a rank of 8, an alpha of 32, and a dropout rate of 0.1. During training, only LoRA parameters were updated, preserving the original pre-trained weights.

[1] https://github.com/facebookresearch/esm

[2] https://github.com/microsoft/LoRA/tree/main

2 Result

In the validation set, the model achieved an F1 score of about 0.93. The use of a pre-trained ESM-2 model as a feature extractor likely contributed to the model's strong performance, as transformer-based models have been shown to be effective at capturing the complex patterns in protein sequences. However, the use of LoRA did not enhance the overall performance very well. It could be due to the hyperparameter for fine-tuning was not optimal enough.

References

[1] Edward J Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[2] Zeming Lin et al. "Language models of protein sequences at the scale of evolution enable accurate structure prediction". In: *bioRxiv* (2022).