Deep Learning
Coding Assignment – 2

Name – Nikhil Saraswat

Roll No. – 20CS10039

Task: Named Entity Recognition

This is a Named Entity Recognition (NER) model using PyTorch and PyTorch Lightning. The model architecture consists of an embedding layer, an LSTM layer, a fully connected layer, and a cross-entropy loss function. The purpose of the model is to identify and classify entities in text data.

The hyperparameters used to train the model are defined at the top of the script. The EMBEDDING_DIM and HIDDEN_DIM hyperparameters control the dimensions of the embedding layer and the LSTM hidden layer, respectively. A larger embedding dimension can capture more information from the input data, while a larger hidden dimension can help the model capture more complex patterns in the data. However, using larger dimensions can also increase the computational cost of training the model.

The NUM_EPOCHS and BATCH_SIZE hyperparameters control the number of epochs to train the model and the batch size to use during training. Increasing the number of epochs can lead to better model performance, but may also increase the risk of overfitting. The batch size determines how many samples are used in each iteration of the model training process. A larger batch size can lead to faster convergence, but may also require more memory.

The NERModel class defines the model architecture, and the __init__ method initializes the layers of the model. The forward method defines the forward pass through the model. The training_step, validation_step, and test_step methods define the training, validation, and testing loops, respectively. These methods take a batch of input data and calculate the output of the model for that batch. The loss_fn attribute of the NERModel class is used to calculate the loss between the predicted output and the true output for each batch. The log method is used to log the loss value for each batch.

The configure_optimizers method is used to define the optimizer used to train the model. In this case, the Adam optimizer is used, which is a popular optimizer that works well in many applications.

In summary, the hyperparameters and functions of this NER model provide a simple yet effective implementation of a machine learning model for identifying and classifying entities in text data. By tuning the hyperparameters, the model can be optimized for a specific NER task. The use of PyTorch Lightning simplifies the training process and allows for easy logging and tracking of the model performance.

The hyper-parameters which have been used in implementation of this model are as following: -

```
EMBEDDING_DIM = [520, 520]
HIDDEN_DIM    = [520, 520]
NUM_EPOCHS    = [10, 10]
BATCH_SIZE    = [35, 35]
SEQ_LEN = 25
```

Note :- We have chosen Sequence length (SEQ_LEN) to 25 only, because if we try to cover all length sentences, then it goes to overfit the model, so we have truncated all the sentences to 25 (Maximum length) in order to maximize the accuracy on test data.

Since we have taken same values for both coarse and fine, so both values of these hyper-parameters are same.

Precision, re-call and F1-score and support values for English fine-grained Model are as follows: -

```
              precision    recall  f1-score   support

    accuracy                           0.91   6249500
   macro avg       0.52      0.35      0.39   6249500
weighted avg       0.91      0.91      0.91   6249500
```

Precision, re-call and F1-score and support values for English Coarse Model are as follows: -

```
              precision    recall  f1-score   support

    accuracy                           0.94   6249500
   macro avg       0.75      0.64      0.68   6249500
weighted avg       0.94      0.94      0.94   6249500
```

Precision, re-call and F1-score and support values for Hindi fine-grained Model are as follows: -

```
              precision    recall  f1-score   support

    accuracy                           0.96    459975
   macro avg       0.78      0.64      0.68    459975
weighted avg       0.95      0.96      0.95    459975
```

Precision, re-call and F1-score and support values for Hindi Coarse Model are as follows: -

```
              precision    recall  f1-score   support

    accuracy                           0.97    459975
   macro avg       0.90      0.76      0.81    459975
weighted avg       0.97      0.97      0.96    459975
```

Precision, re-call and F1-score and support values for Bengali fine-grained Model are as follows: -

```
              precision    recall  f1-score   support

    accuracy                           0.95    496475
   macro avg       0.69      0.64      0.64    496475
weighted avg       0.95      0.95      0.95    496475
```

Precision, re-call and F1-score and support values for Bengali Coarse Model are as follows: -

```
              precision    recall  f1-score   support

    accuracy                           0.96    496475
   macro avg       0.82      0.77      0.79    496475
weighted avg       0.96      0.96      0.96    496475
```

Link to all models is as follows: -

https://drive.google.com/drive/folders/1-CWkC-VEl2bLbKP9M-yCXpNR13AcJpHY?usp=sharing