

Task 1 Details:

The model summary is as follows:

- Embedding Dimensions = 100
- Hidden Dimension = 256
- Num_layers = 1
- Dropout = 0.33
- Pad_idx = 0
- Output_dim = 128

The classifier for the model of task one is a linear layer with:

- Input dimension = 128
- Output dimension = 9

The loss used is Cross Entropy Loss with following parameters:

- Ignore Index = -1 {for padding}
- weight = [1,0.7,1,1,1,1,1,1,1]

The optimizer used is SGD with following parameters:

- learning rate = 0.01
- weight_decay = 1e-6
- momentum = 0.99

The learning rate scheduler used is ReduceLROnPlateau with the following parameters:

- factor = 0.85
- patience = 2

The first task was trained on 55 epochs with the batch_size of 32.

After evaluating the results using the script provided, the following results were achieved for the dev data:

Accuracy: 95.79%

Precision: 78.71%

Recall: 79.10%

FB1: 78.91%

Task 2 Details:

For task 2, I have generated the sentences and ner using the index as with task 1. I have additionally generated the cases of every word for the sentence as a separate tensor.

The embedding layer in the model generates the embeddings of the sentences and then concatenates the cases tensor with it keeping the embedding dimension as the previous model i.e 100.

The embedding matrix is generated using the glove embeddings given to us and it is then trained using the embedding layer. The embedding layer has a padding_idx = 0 so that the padding tokens are not trained.

The model summary is as follows:

- Embedding Dimensions = 100
- Hidden Dimension = 256
- Num_layers = 1
- Dropout = 0.33
- Pad_idx = 0
- Output_dim = 128

The classifier for the model of task one is a linear layer with:

- Input dimension = 128
- Output dimension = 9

The loss used is Cross Entropy Loss with following parameters:

- Ignore Index = -1 {for padding}
- weight = [1,0.7,1,1,1,1,1,1,1]

The optimizer used is SGD with following parameters:

- learning rate = 0.01
- weight_decay = 1e-6
- momentum = 0.99

The learning rate scheduler used is ReduceLROnPlateau with the following parameters:

- factor = 0.85
- patience = 2

The first task was trained on 120 epochs with the batch_size of 32.

After evaluating the results using the script provided, the following results were achieved for the dev data:

Accuracy: 97.96%

Precision: 89.05%

Recall: 87.46%

FB1: 88.25%