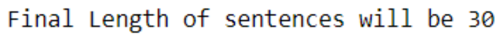
Manan Rajdev

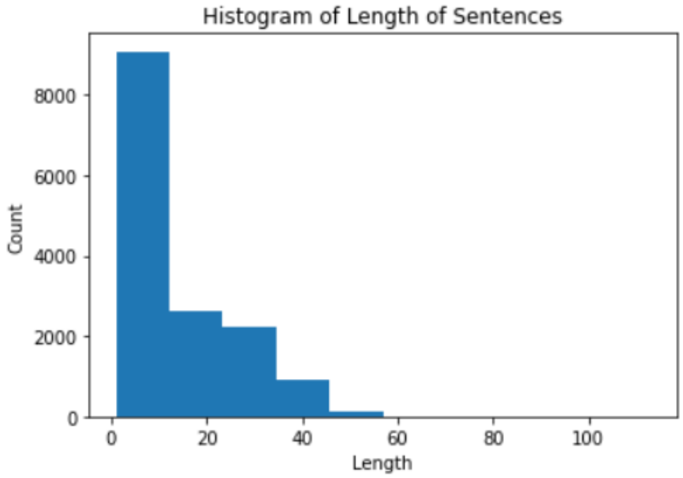
Task 1

Converting Sentences to equal lengths:

I trimmed the sentences which were more than 30 and those which were less than 30, I added 0 (padding) at the end. I choose 30 after analyzing this histogram



```
Final Length of sentences will be 30
```



Initialization of weights:

To initialize all parameters from normal distribution with mean=0 and std=0.1 as it helps with converging during training

Model Parameters:

output_size=128,

hidden_dim=256

embed_size=100

dropout_rate=0.33

n_layers=1

criterion = nn.CrossEntropyLoss(ignore_index=0)

optimizer = torch.optim.SGD(model.parameters(), lr=0.1, momentum=0.9)

N_EPOCHS=15

Epoch: 15 Train Loss: 0.047 | Val. Loss: 0.133

Task 2

Handling Case Insensitivity

Adding 0 or 1 as the first element of the embedding deciding on whether the first letter is uppercase or lowercase,This will help in providing case sensitivity as Glove embeddings are case insensitive.

Initialization of weights:

Done by using the Glove embeddings

Model Parameters:

output_size=128,

hidden_dim=256

embed_size=100

dropout_rate=0.33

n_layers=1

criterion = nn.CrossEntropyLoss(ignore_index=0)

optimizer = torch.optim.SGD(model.parameters(), lr=0.1, momentum=0.9)

N_EPOCHS=15

Epoch: 15 Train Loss: 0.083 | Val. Loss: 0.058

Performs better than task 1 due to the pretrained glove embeddings

Task 3

Converting Sentences and words to equal lengths:

I trimmed the sentences which were more than 28 and those which were less than 30, I added 0 (padding) at the end.

For words, I chose the max length that is 19, all words smaller than that were padded with 0

I used Conv1d layer for character embeddings with parameters mentioned below

Model Parameters:

```
The model has 1,110,488 trainable parameters.
Model_Glove_LSTM_CNN(
  (embedding): Embedding(20610, 100, padding_idx=0)
  (emb_dropout): Dropout(p=0.33, inplace=False)
  (char_emb): Embedding(85, 30, padding_idx=0)
  (char_cnn): Conv1d(30, 150, kernel_size=(3,), stride=(1,), groups=30)
  (cnn_dropout): Dropout(p=0.25, inplace=False)
  (lstm): LSTM(250, 256, batch_first=True, bidirectional=True)
  (fc_dropout): Dropout(p=0.33, inplace=False)
  (fc): Linear(in_features=512, out_features=128, bias=True)
  (elu): ELU(alpha=1.0)
  (linear_classifier): Linear(in_features=128, out_features=10, bias=True)
)
```

N_EPOCHS=15

```
Epoch: 15 Train Loss: 0.033 | Val Loss: 0.072
```

Performs the best due to CNN char embedding