HW3 – Asaf Mizrahi & Yahel Sherman

Our model is:

Input >>> Embeding layer + pad (for uniform output) >>> dropout >>> LSTM\* >>> averaging >>> linear layer

\*LSTM model is:

self.lstm = nn.LSTM(input\_size=self.lstm\_args.input\_size, hidden\_size=self.hidden\_size,

batch\_first=self.lstm\_args.batch\_first, bias=self.lstm\_args.bias,

num\_layers=self.lstm\_args.num\_layers, bidirectional=self.lstm\_args.bidirectional,

dropout=self.lstm\_args.dropout, proj\_size=self.lstm\_args.proj\_size)

parameters choosed:

model\_args:

output\_size: 5 # Should correspond to the number of classes

dropout: 0.5 # Dropout of the final classifier in the model

lstm\_args:

input\_size: 50 # Size of the input to the LSTM

hidden\_size : 256 # Size of the hidden state of the LSTM

num\_layers: 3

bias: true

batch\_first: true

dropout: 0.2

bidirectional: true

proj\_size: 0

we didn’t played much with the given parameters, we did added OneCycleLR scheduler to the learning rate and changed the batch size, all of the details mentioned above you can find in the attached .py files.

Our final results after 50 epochs are:

