

Deep Learning CS69000 Homework 4

Tunazzina Islam, islam32@purdue.edu

March 2020

Q0

1. Student interaction with other students / individuals:
(c) No, I did not discuss the homework with anyone.
2. On using online resources:
(b) I have used online resources to help me answer this question, but I came up with my own answers.

Here is a list of the websites I have used in this homework:

- pytorch forum discussion
- <https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-with-graph-convolutional-networks-7d2250723780>

Graph Neural Networks (GNN)

Task 1a: GNN with Mean Aggregator

1. Formula: Let's assume \bar{A} is the average adjacency matrix, then $\bar{A}_{uv} = \frac{A_{uv}}{\sum_v A_{uv}}$

$$\mathbf{H}^{(k+1)} = \begin{cases} \sigma(\mathbf{W}^{(k+1)} [\mathbf{H}^{(k)}, \bar{\mathbf{A}}\mathbf{H}^{(k)}] + \mathbf{b}^{(k+1)}) & k+1 > 0 \\ X & k+1 = 0 \end{cases} \quad (1)$$

2.

| | Min | Max | Average |
|------------|-----|-----|---------|
| Training | 1 | 21 | 3.13 |
| Validation | 1 | 23 | 3.92 |
| Test | 1 | 168 | 4.81 |

3. (a)

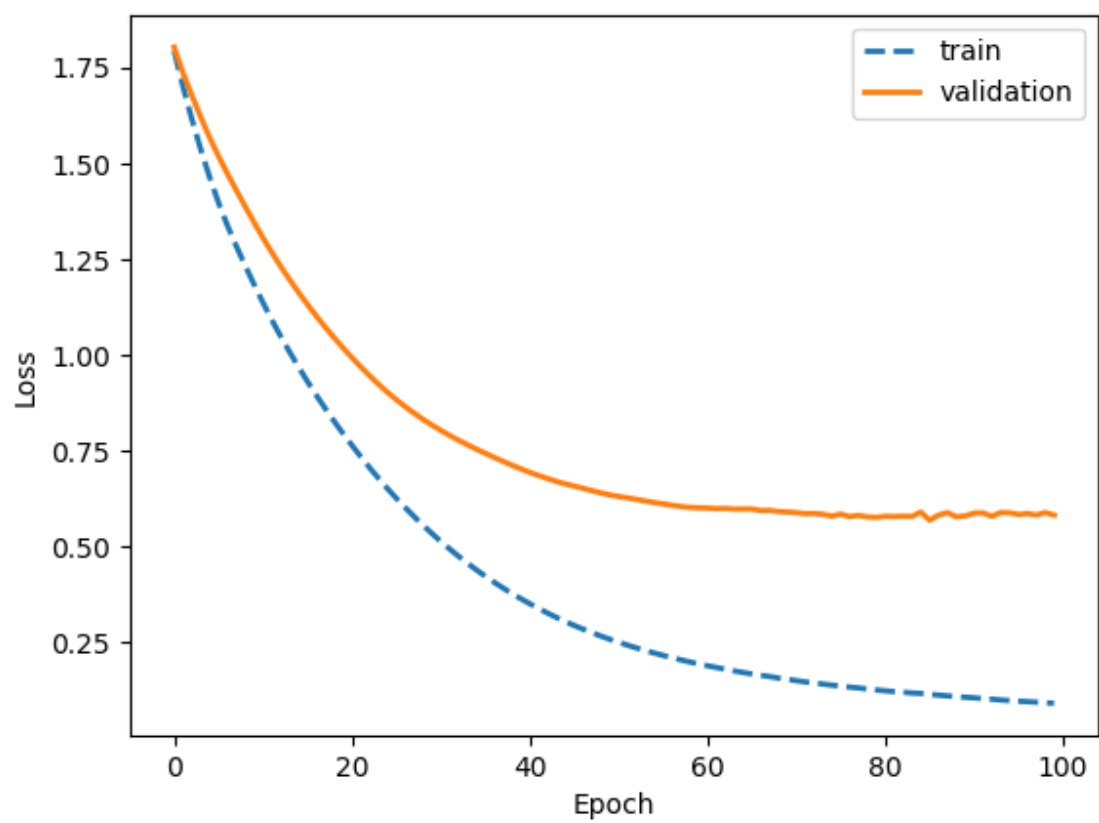


Figure 1: Learning curves of loss function of training and validation nodes.

3.(b)

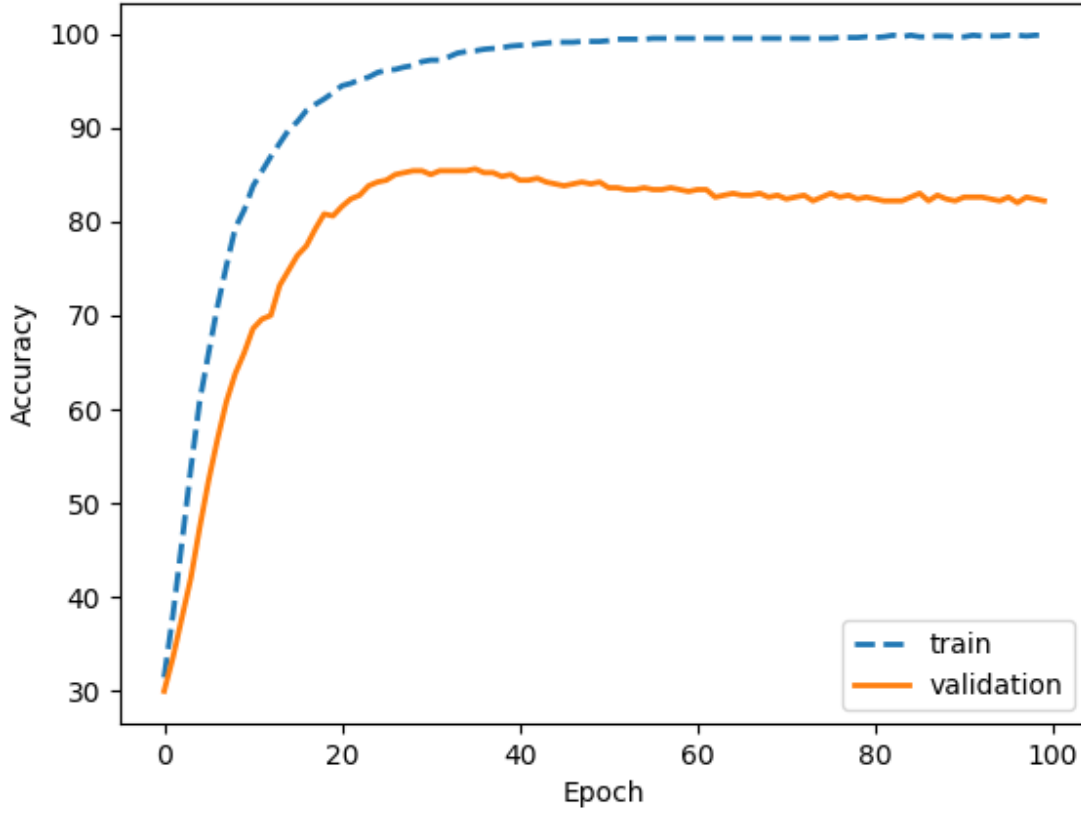


Figure 2: Learning curves of accuracy of training and validation nodes.

3.(c) Test Loss : 0.591586

Test Accuracy : 82.300%

4. Maximum MSE for layer 2 : 2.00

Maximum MSE for layer 20 : 3.8025e-14

We plotted learning curve (loss and accuracy) for layer 20. We noticed that there is not much improvement with change of epoch during training process. Please see Fig. 3 and Fig. 4. This is because of embedding mixing problem with too many layers. Maximum MSE loss is also close to zero for layer 20.

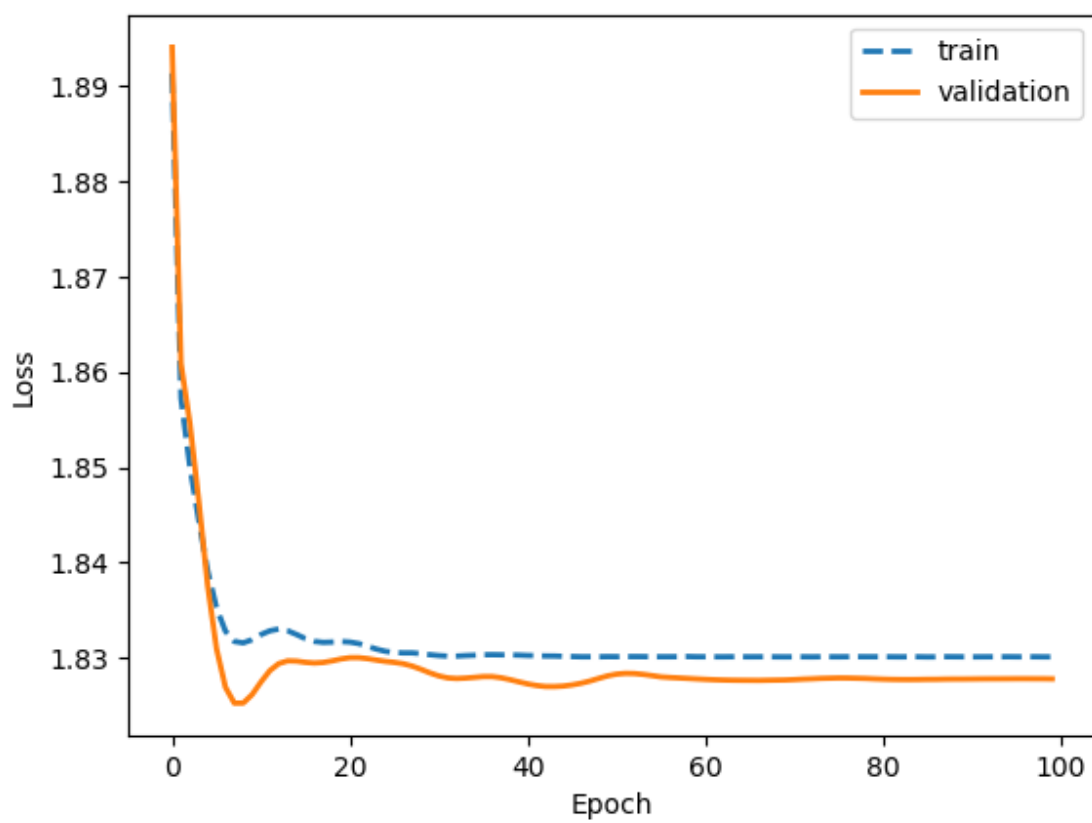


Figure 3: Learning curves of loss function of training and validation nodes for layer 20.

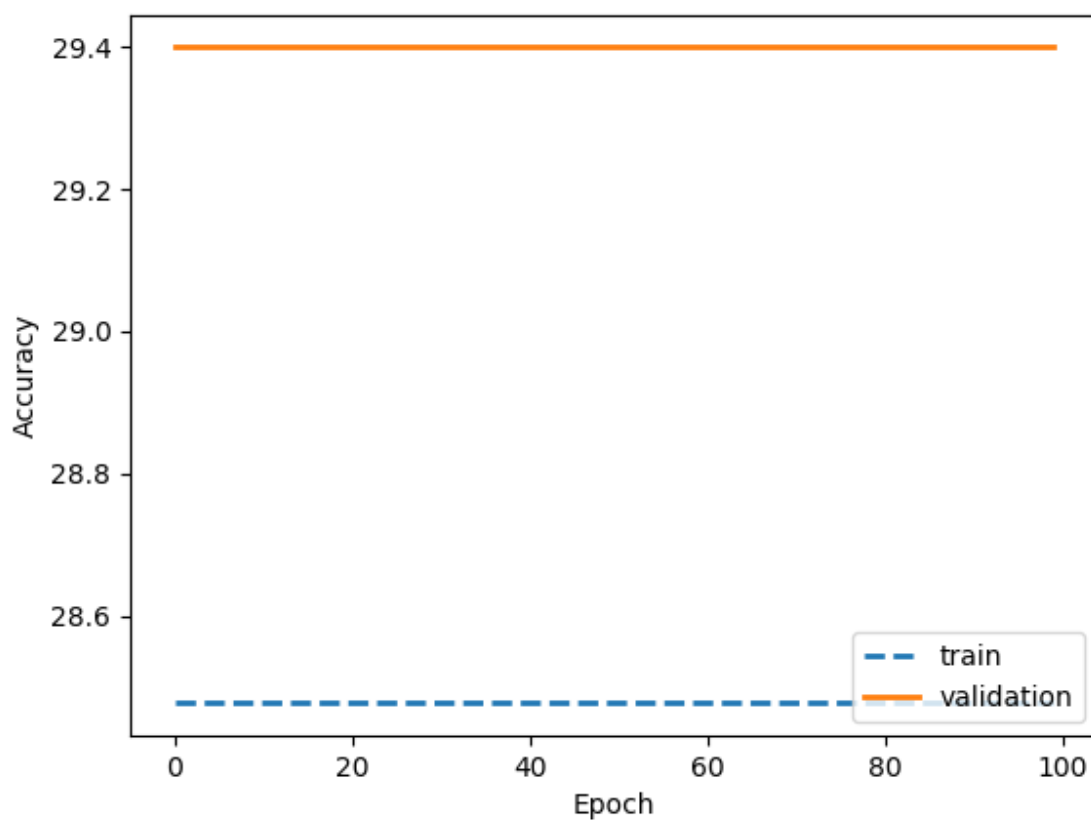


Figure 4: Learning curves of accuracy of training and validation nodes for layer 20.

Task 1b: GNN with Mean and LSTM Aggregator

1. We think both mean and LSTM aggregator will face embedding mixing problem for layer 20 like previous section. But LSTM aggregator might do better as we are using 5-ary Janossy pooling trained with π -SGD.
2. We filled the missing part in model.py.
3. (a) Loss for Mean Aggregator:

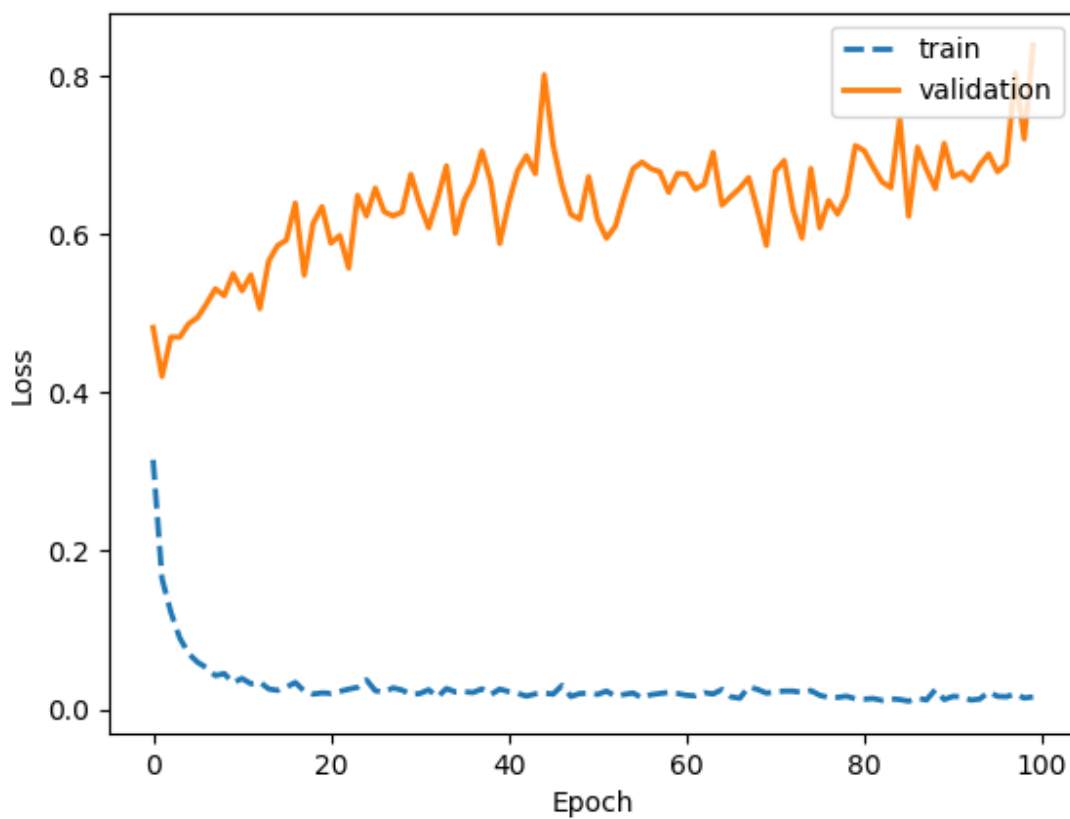


Figure 5: Learning curves of loss function of training and validation nodes Mean aggregator.

3.(a) Loss for LSTM Aggregator:

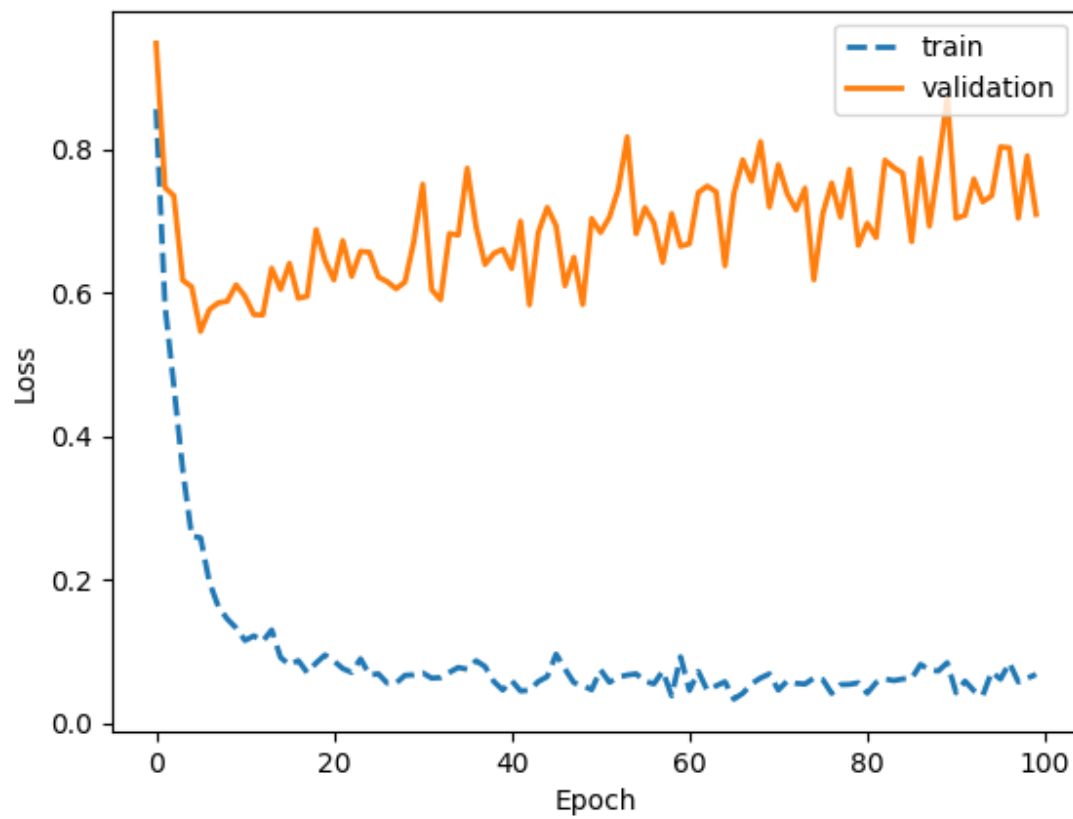


Figure 6: Learning curves of loss function of training and validation nodes LSTM aggregator.

3.(b) Accuracy for Mean Aggregator:

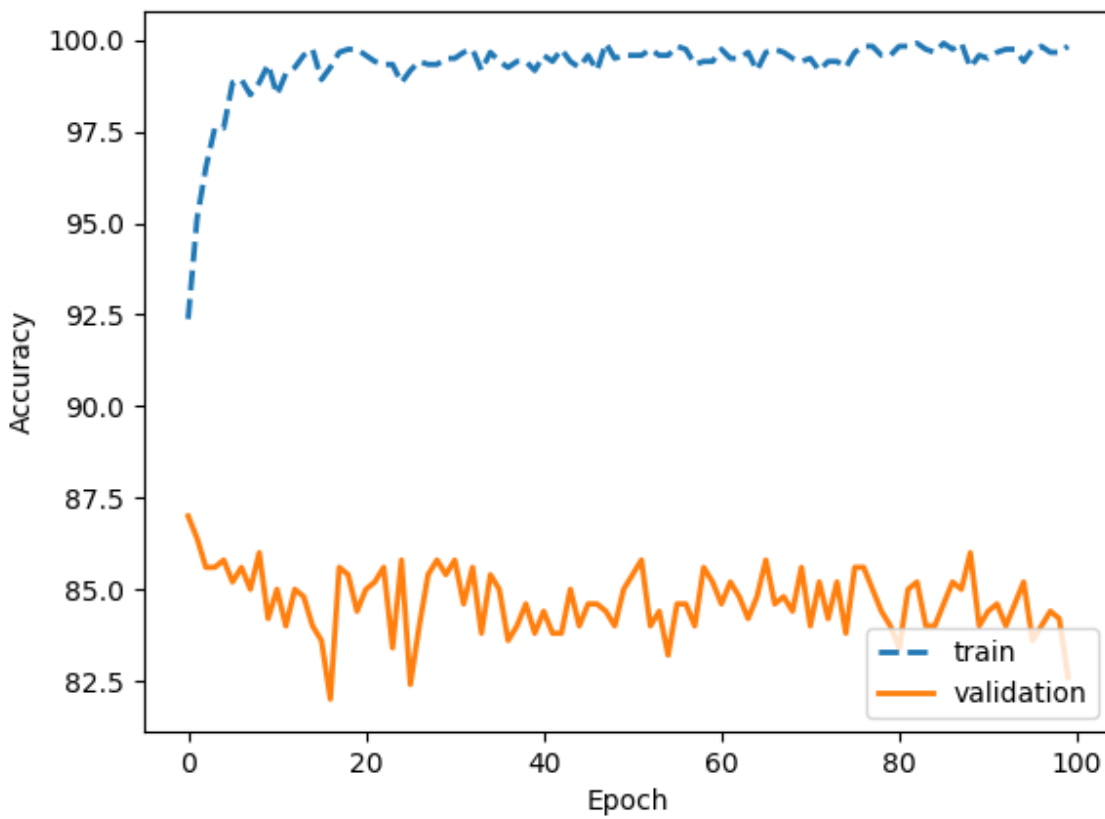


Figure 7: Learning curves of accuracy of training and validation nodes Mean aggregator.

3.(b) Accuracy for LSTM Aggregator:

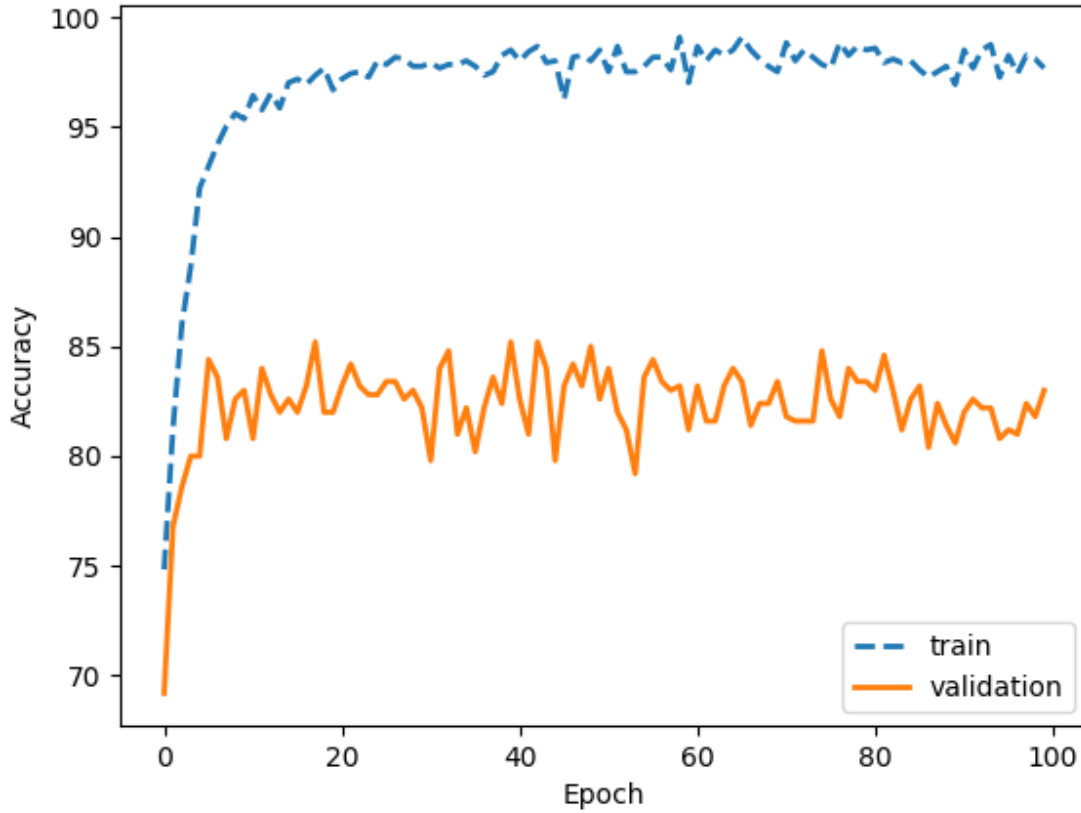


Figure 8: Learning curves of accuracy of training and validation nodes LSTM aggregator.

3.(c) For Mean Aggregator:

Test Loss : 0.482320

Test Accuracy : 84.700%

For LSTM Aggregator:

Test Loss: 0.724231

Test Accuracy: 78.700%

4. To run the code for average prediction of 20 permutations following command has been used:

```
python GraphSAGE/main.py - -agg LSTM - -num_samples 20 - -device 0 > log_GraphSAGE_LSTM_perm20
```

Test Loss : 0.622885

Test Accuracy : 80.100%

We can see that sampling 20 permutations improves the performance than the 1 permutation because of sampling more permutations will help acquire true π -SGD.

Language Model (LM)

Task 2a: Markov Chain Modeling

2. Perplexity:

| Order | Train | Validation | Test |
|-------|--------|------------|----------|
| 1 | 68.663 | 506.526 | 448.552 |
| 2 | 8.790 | 1738.608 | 1735.377 |
| 10 | 1.985 | 8290.073 | 8955.008 |

Task 2b: LSTM Modeling

1. We splitted training sequence following way:

$$\text{BPTT Batches} = \left\lceil \frac{\text{Batch Sequence Length}}{BPTT} \right\rceil$$

2. a) Learning curves of loss function for $\text{bptt} = 5$:

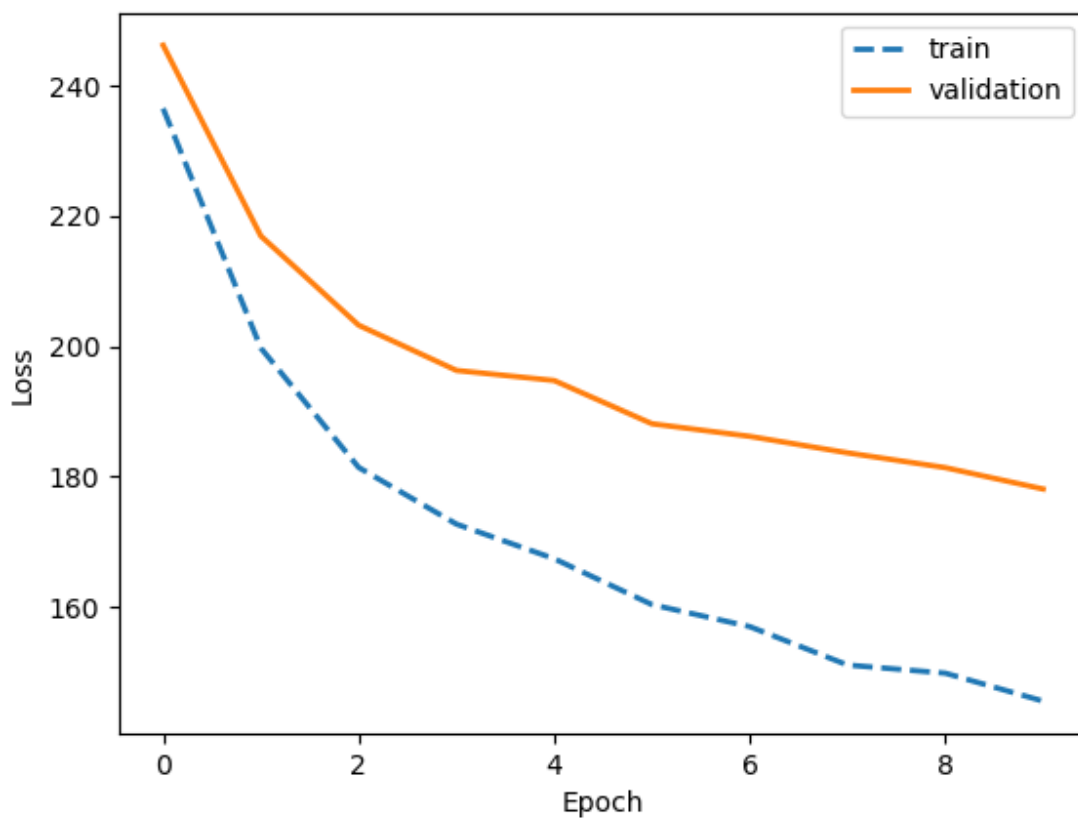


Figure 9: Learning curves of loss function of training and validation nodes for $\text{bptt} = 5$.

2.(a) Learning curves of loss function for $\text{bptt} = 35$:

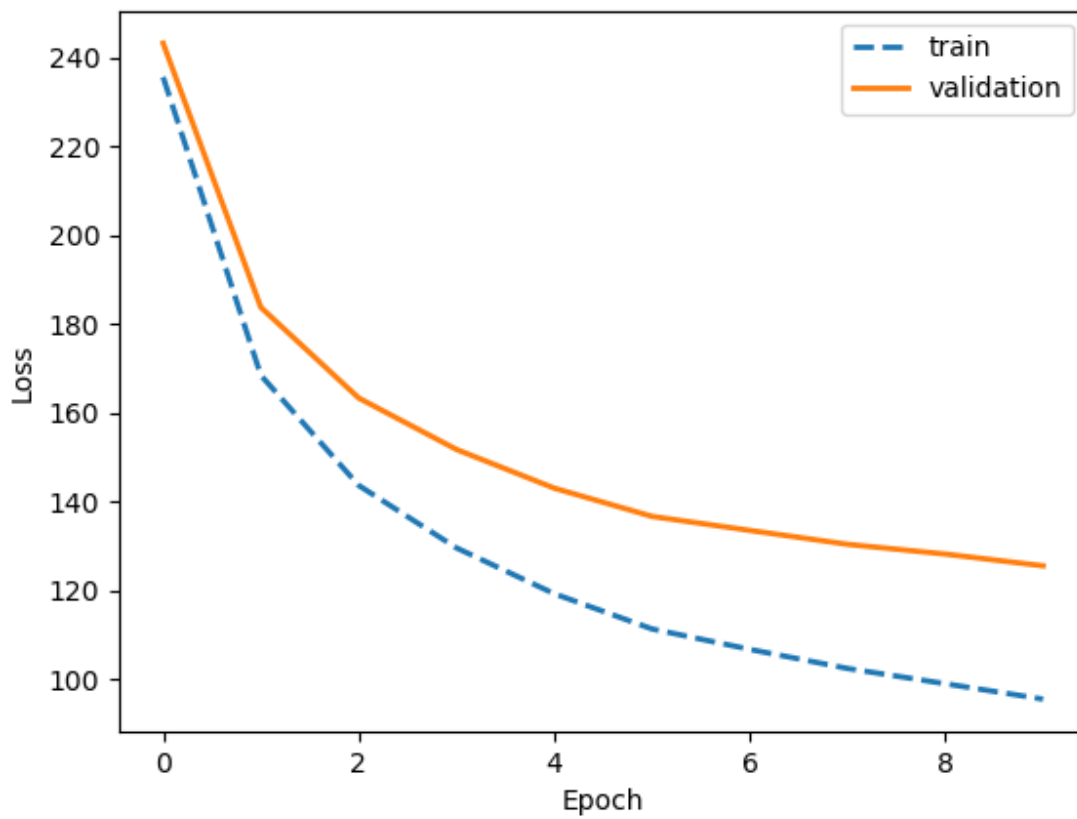


Figure 10: Learning curves of loss function of training and validation nodes for $\text{bptt} = 35$.

2.(a) Learning curves of loss function for $\text{bptt} = 80$:

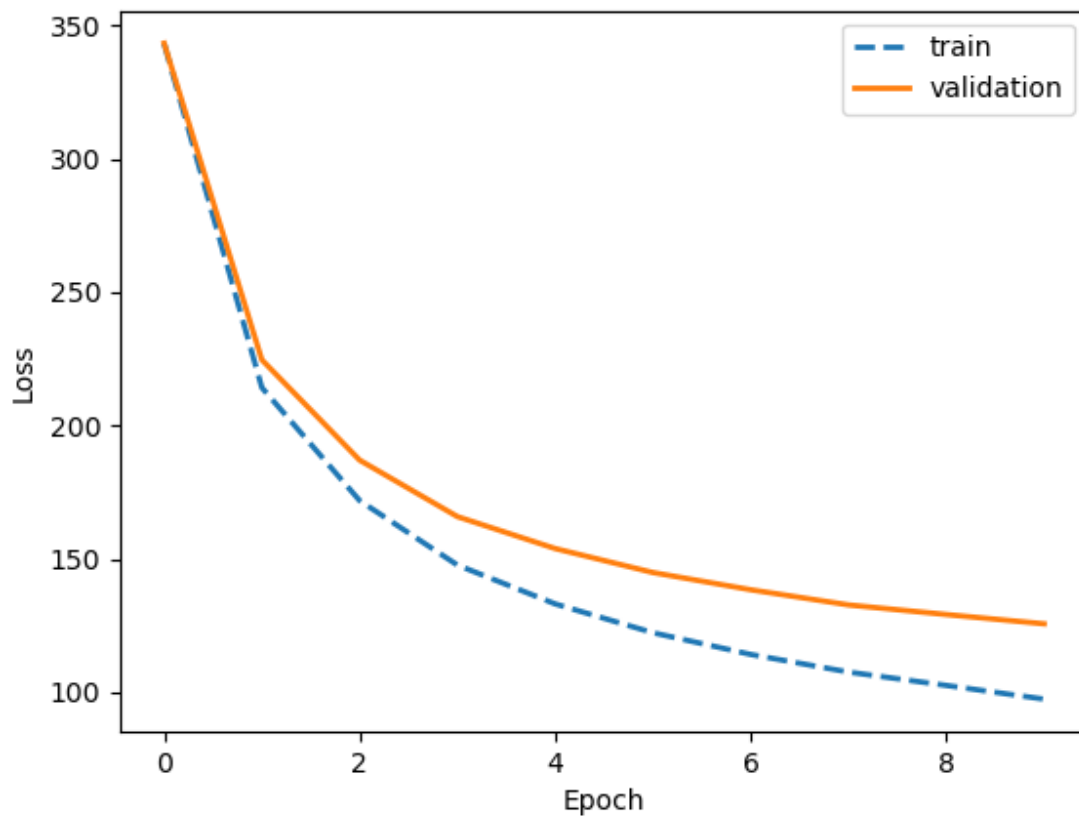


Figure 11: Learning curves of loss function of training and validation nodes for $\text{bptt} = 80$.

3. b) Learning curves of accuracy for $\text{bptt} = 5$:

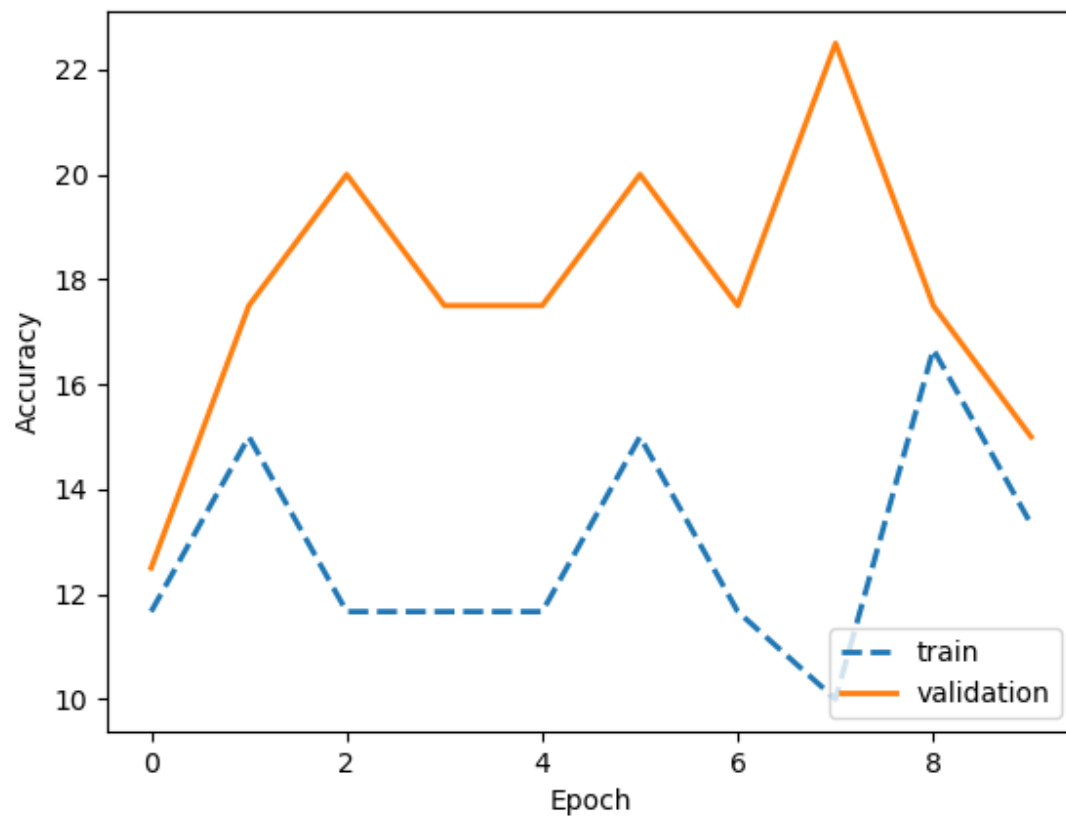


Figure 12: Learning curves of accuracy of training and validation nodes for $\text{bptt} = 5$.

2.(b) Learning curves of accuracy for $\text{bptt} = 35$:

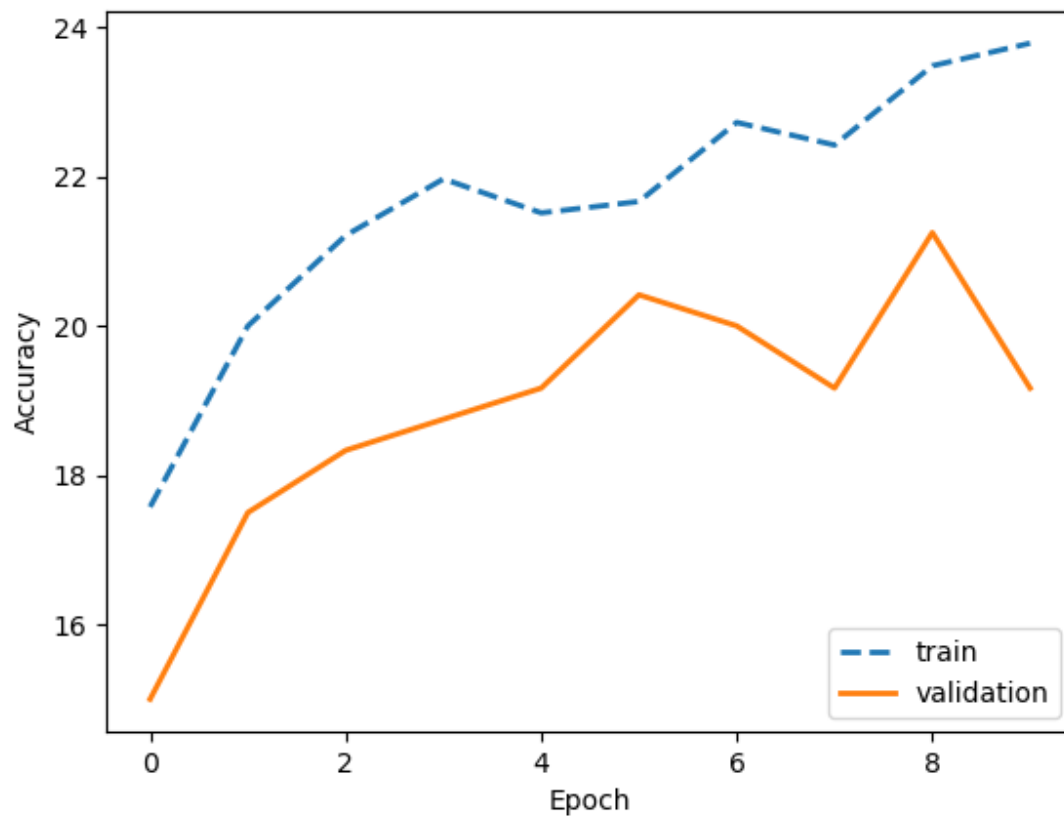


Figure 13: Learning curves of accuracy of training and validation nodes for $\text{bptt} = 35$.

2.(b) Learning curves of accuracy for $\text{bptt} = 80$:

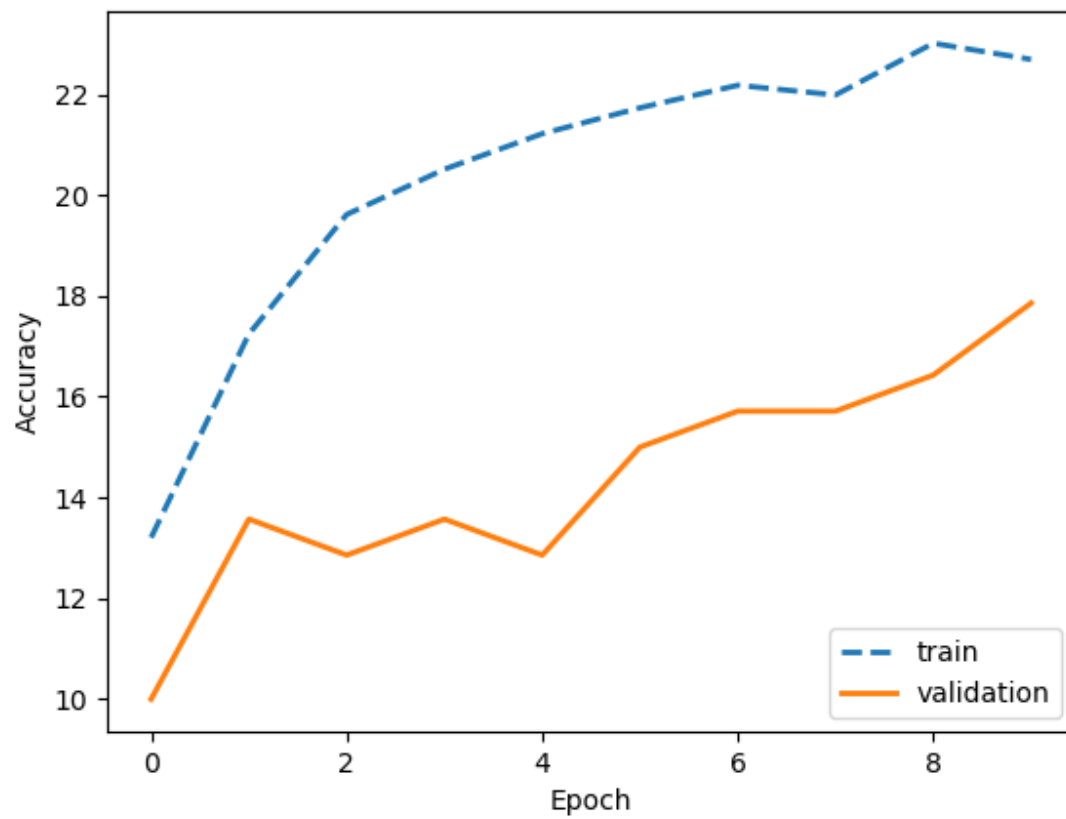


Figure 14: Learning curves of accuracy of training and validation nodes for $\text{bptt} = 80$.

2.(c) Loss and accuracies of Test nodes:

| BPTT | Loss | Accuracy(%) |
|------|---------|-------------|
| 5 | 171.376 | 25 |
| 35 | 122.351 | 25 |
| 80 | 123.242 | 17.241 |