COL870 Special Topics in GNNs **Semester 2 2022-23**

Assignment 1

Instructor: Sayan Ranu Rishi Shah

1 GNN Implementation

The architecture includes the following :

1. Mean Aggregation
2. MLP(single layer NN followed by RELU)

The above is done 'k' times. Instead of single layer MLP, I tried multi-layer however the time taken to train the model increased significantly(>1 hr to run experiments), so I sticked with single layer.
Now, the mean aggregation was implemented using matrix multiplication. From the edge indexes, I create adjacency matrix. Then I add self loops to the adjacency matrix. The aggregation formula is :
$X = AX$
This does the work specified in the assignment statement.
Training, validation, testing strategy followed is standard. I used CrossEntropy Loss and Adam Optimiser(with weight decay). Also, I have used batch normalisation.

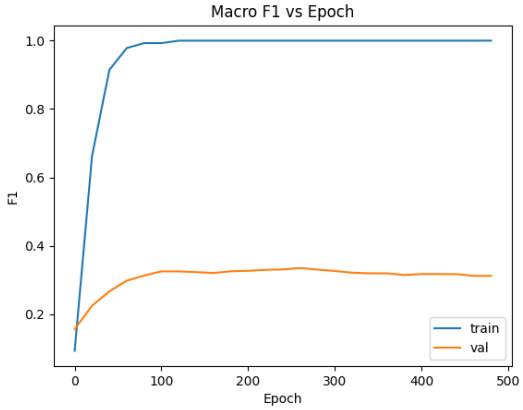
2 Impact of Topology Experiments

To get the impact of topology in the prediction task, I varied the number of GNN layers used and analysed the results. The number of GNN layers directly affects the distance of nodes from which the messages aggregates. To elaborate, if the model has 'k' GNN layers then message get aggregates from 'k' hop neighbours.

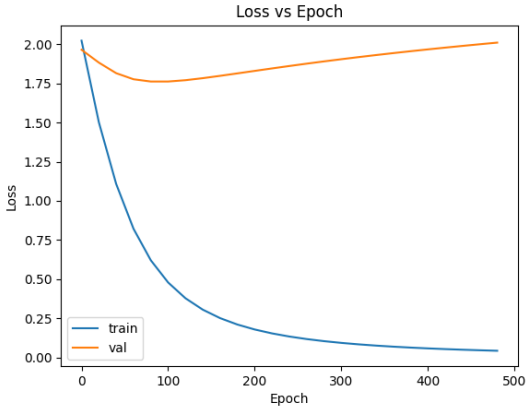
2.1 Results

Here are loss and f1 plots of varying the value of 'k'(number of gnn layers) throughout.

1. k=0

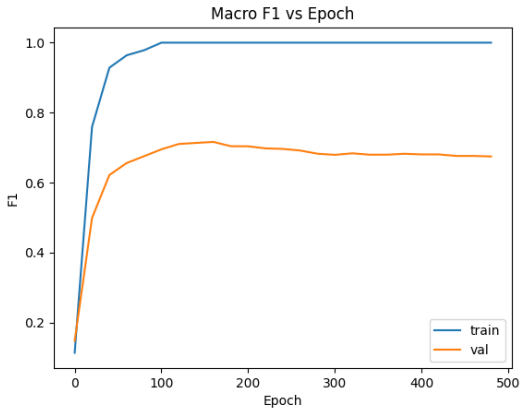


(a) F1 vs Epochs

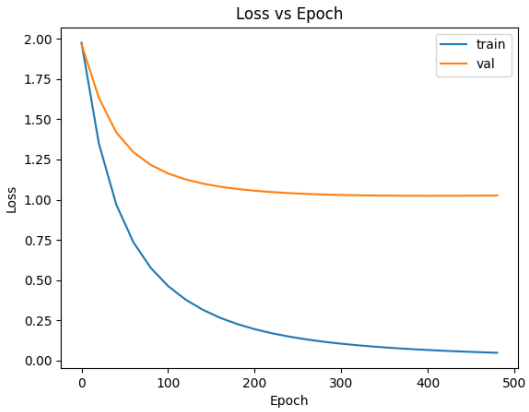


(b) Loss vs Epochs

2. k=1

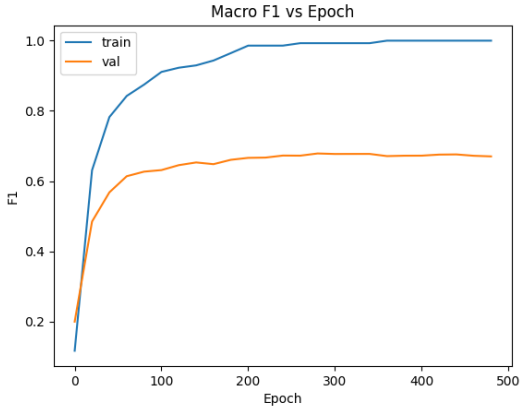


(a) F1 vs Epochs

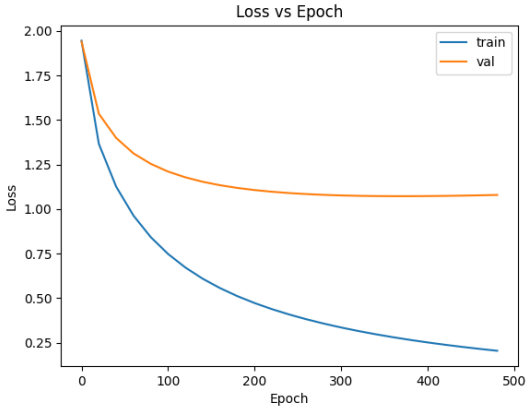


(b) Loss vs Epochs

3. k=2

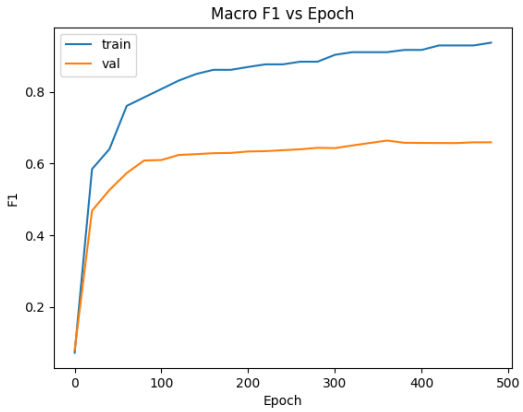


(a) F1 vs Epochs

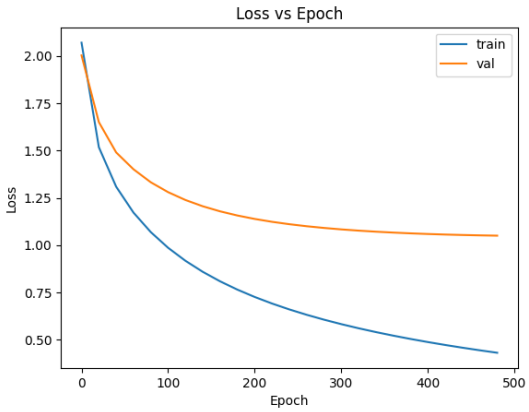


(b) Loss vs Epochs

4. k=3

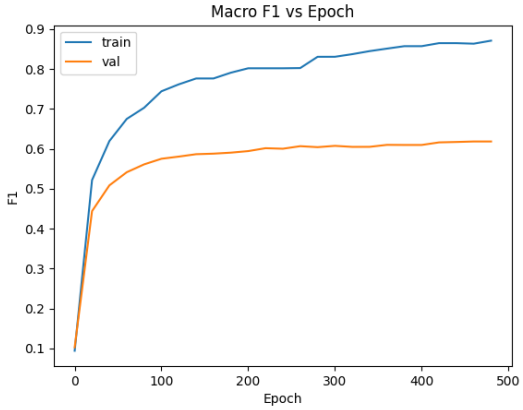


(a) F1 vs Epochs

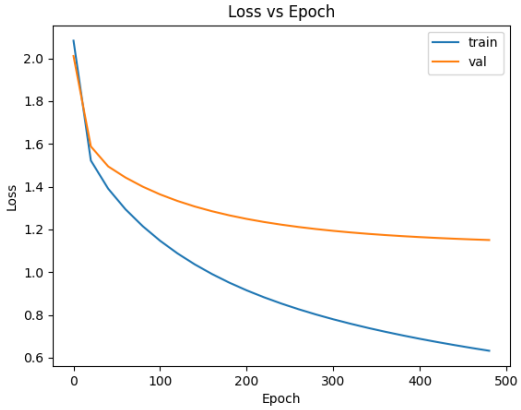


(b) Loss vs Epochs

5. k=4

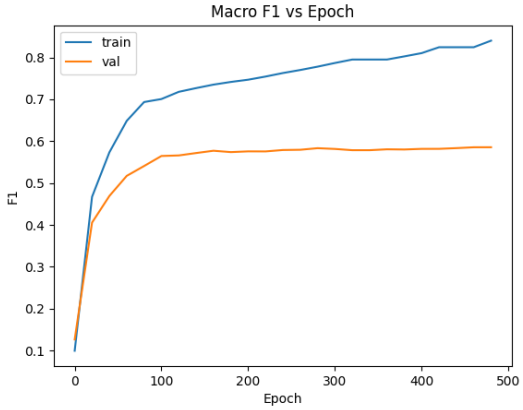


(a) F1 vs Epochs

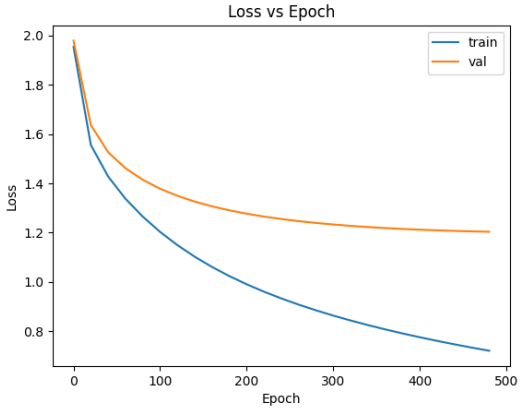


(b) Loss vs Epochs

6. k=5

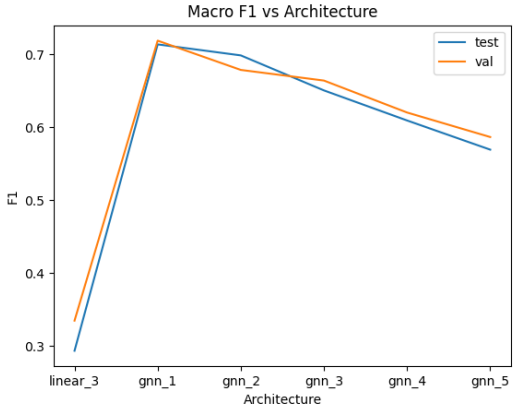


(a) F1 vs Epochs



(b) Loss vs Epochs

Finally, here is a plot which summarises all the results -



(a) F1 vs Architectures

2.2 Analysis

It can be seen that when we move from simple MLP to GNNs we get a huge improvement. The reason is that messages get aggregated from neighbours and information gets shared. It helps to provide generalisation much better.
However, as the model gets deeper it leads to problems of over-smoothening. When message gets aggregated from far neighbours, what ends up happening is that every node starts getting similar features. So, we loose the adventures of aggregation. Hence, the F1-score and accuracy decreases. One thing to note is that the hyperparameters are kept same for all the experiments. Also, the total parameters increase as 'k' increases, so there is a slight possibility that other than over-smoothening, overfitting is also the reason for bad results as large model is getting overfitted on the training nodes.

2.3 Conclusion

The final conclusion is that in case of MPNNs in general deep networks do not perform very well. We have to find the correct value of 'k' using the validation set.