Report:
Natural Language Processing

The implementation of the model was done as follows:
I first used the pre-existing LSTM and worked on implementing the MoE layer using the explanation provided in the paper. I started with the Softmax gating as that was the easiest to implement. Then, I implemented the LSTM from scratch after looking at a few articles online. The final combined network is made up of an MoE layer stacked between 2 LSTM layers.

1) Conll2003 dataset :
Preprocessing was done on the dataset after importing. I tokenized the input using the BERT tokenizer. Then, I used a function to align my target labels with the tokens, while also giving the label -100 to irrelevant tokens. Finally, I used the CrossEntropy loss to train my model (as it was a token classification job).
A few runs were conducted and I got the output as follows:
4 Experts: Loss = 1.14, Accuracy = 82.53
Without MoE: Loss = 1.14, Accuracy = 82.53
5 Experts: Loss = 1.14, Accuracy = 82.53
Analysis of the impact of the MoE layer and the number of experts is done below.

2) Squad dataset:
Preprocessing was done using the T5 Tokenizer and created a dataset with input_ids, decoder_ids and attention masks. I ran out of time while working on this dataset, so I had to settle for a very mediocre model as I had to use MSELoss to optimize it. I compared the predicted sequence with the target sequence. The code runs as of now (no errors). I'll try to achieve better performance after the submission is over. CrossEntropy loss works best here as the predicted sequence is just classifying the token over the maximum number of token ids. I tried it out, but Colab was giving me issues.

Impact of the MoE layer depends highly on the data diversity and complexity, which is why I feel that the model performance didn't change for the conll2003 dataset (the task being very simple)..
As for the number of experts, the impact can be either positive or negative depending upon the task at hand. On one hand, it provides more flexibility and improved specialization to the model, but on the other hand, it is also prone to overfitting, high computational complexity.

I was hoping to implement the Noisy Top-K gating, but couldn't due to lack of time. It was a very good learning opportunity for me as I was new to NLP before starting the assignment.