COL775:Deep Learning

Assignment 1.2: Text to the Math program

Name: Bogam Sai Prabhath Entry Number:2023AIB2079

Link to the trained model is here

In Math Word Problem solving, given a mathematical problem specified in the text, the goal is to find the solution to the problem by applying mathematical reasoning to the input text. One of the ways to generate executable programs from their textual description is to make use of Encoder-Decoder architecture For this part of the assignment, we will make use of seq2seq architectures to convert input text into the corresponding symbolic form and use the provided Evaluator for finding the accuracy.

1. Architecture 1 :

As part of the first Architecture, A Seq2Seq model with GloVe embeddings, using a Bi-LSTM encoder and an LSTM decoder.The problem tokens embeddings are done using the Glove and prediction Token embeddings are learned.

1.1 parameters and Model details

Encoder: Bi-LSTM Encoder
Decoder: LSTM Decoder
Encoder embedding: Glove
Embedding_dimension: 200

Encoder hidden_size: 512
Dropout probability: 0.5
Decoder hidden_size : 512
Teacher Forcing ratio: 0.6
Loss : CrossEntropyLoss
Optimizer: Adam
Learning Rate: 0.001
Epoches : 50

1.2 Results :

Figure 1 : Train and Validation Loss curve over 50 Epochs for Architecture 1 TF=0.6

Accuracy :
For the Accuracy calculation, the provided Evaluator is used.

Beam search (k): 10
Max Prediction Length: 300

These parameters are followed for all the other models.

Accuracy Metric	Test	Validation
Exact Match Accuracy	22.28	21.37
Execution Accuracy	28.779	27.86

Table 1 : Train and Validation Accuracy curve for Architecture 1 , Beamsize=10

2. Architecture 2 :

As part of the second Architecture, the Seq2Seq+Attention model with GloVe embeddings, using a Bi-LSTM encoder and an LSTM decoder, is implemented

2.1 parameters and Model details

Encoder: Bi-LSTM Encoder
Decoder: LSTM Decoder
Encoder embedding: Glove
Embedding_dimension : 200
Encoder hidden_size : 256
Dropout probability: 0.5
Decoder hidden_size : 256
Teacher Forcing ratio: 0.6
Loss : CrossEntropyLoss
Optimizer: Adam
Learning Rate: 0.001
Epoches: 50

2.2 Results:

Accuracy Metric	Test	Validation
Exact Match Accuracy	16.155	16.54
Execution Accuracy	23.27	23.47

Table 2 : Train and Validation Accuracy curve for Architecture 2 , Beamsize=10

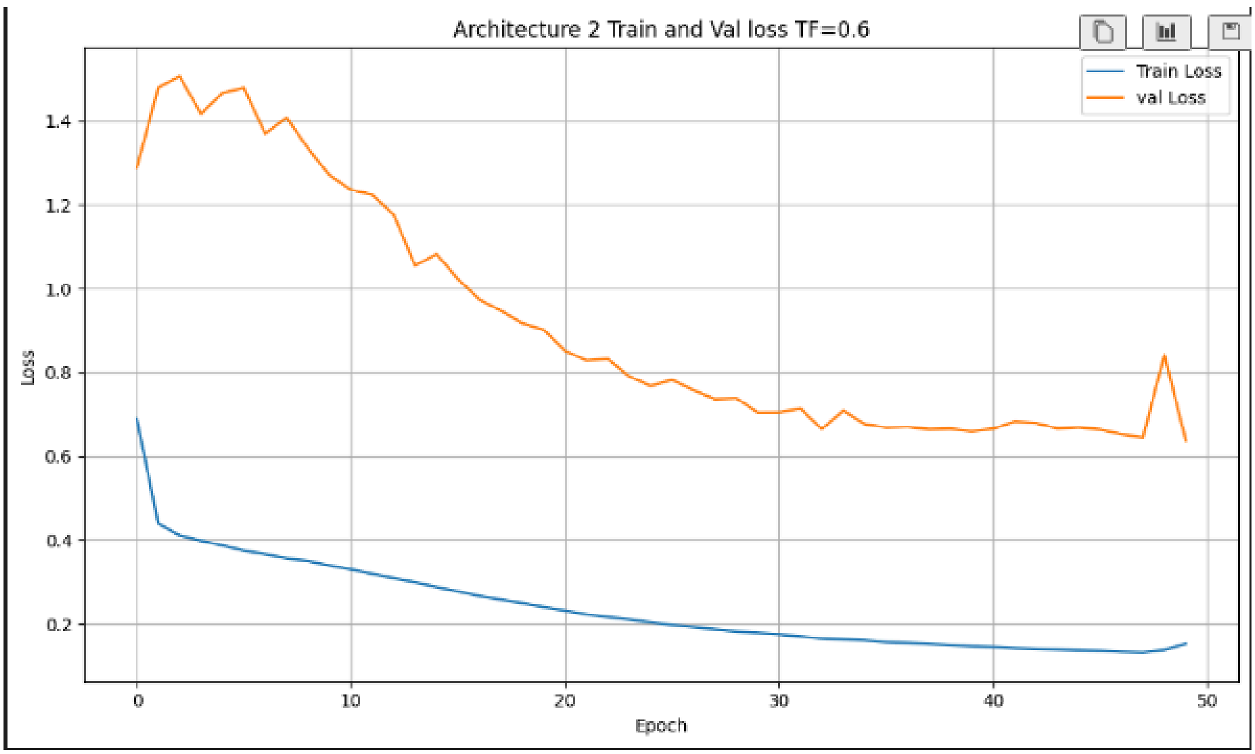

Figure 2 : Train and Validation Loss curve over 50 Epochs for Architecture 2 TF=0.6

2.3 Effect of Teacher Forcing Probability:

The same model is implemented by changing the Teacher Forcing Ratio to 0.3 and 0.9, and the following observations are made

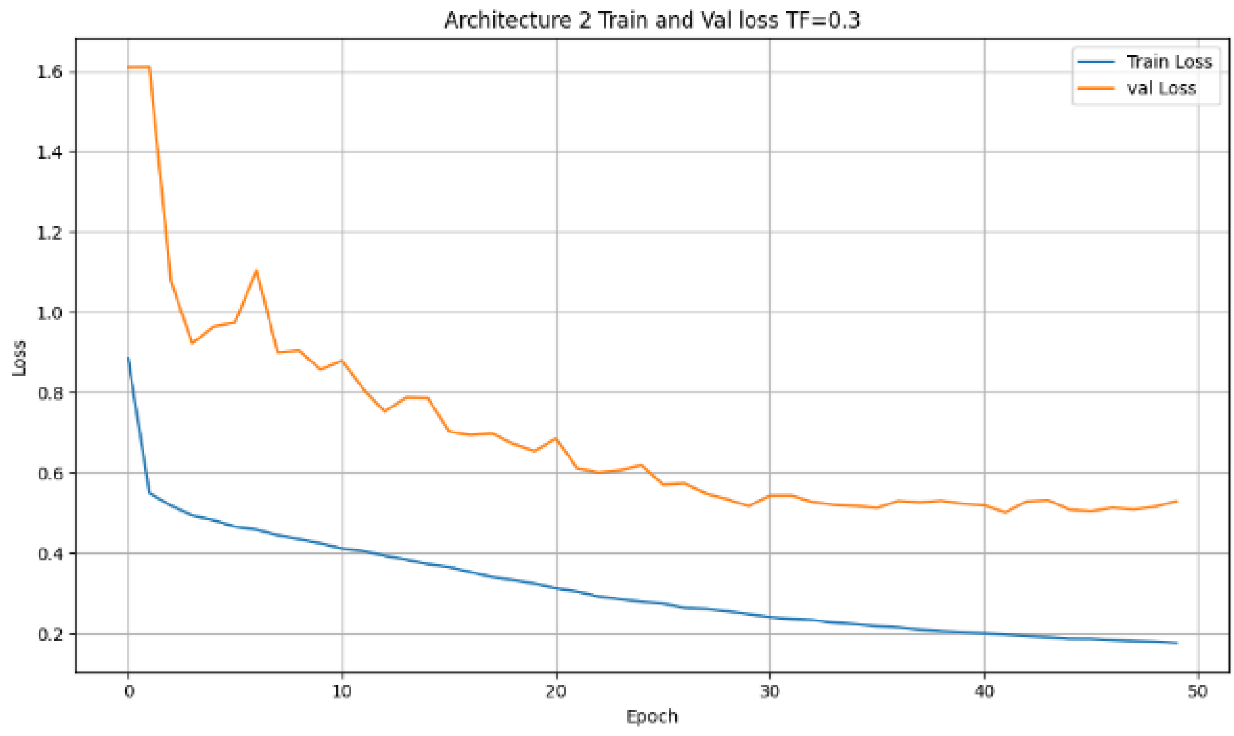

Figure 3 : Train and Validation Loss curve over 50 Epochs for Architecture 2 TF=0.3

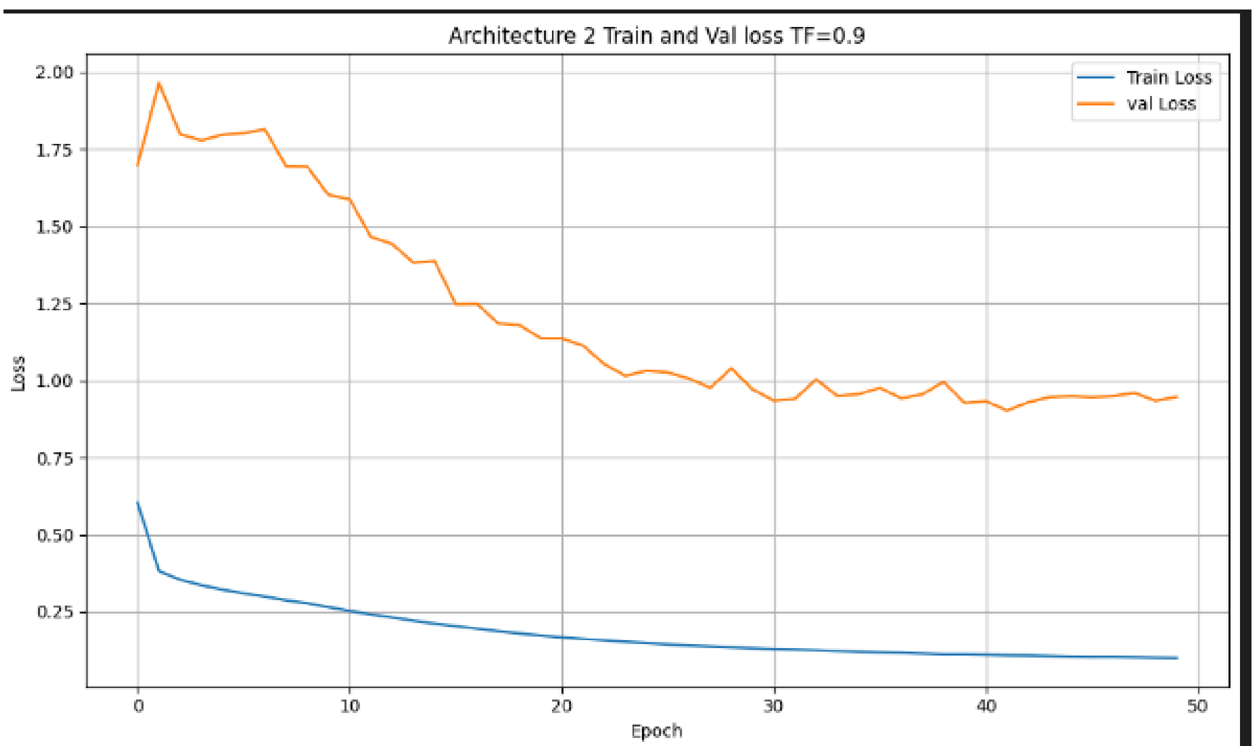

Figure 4 : Train and Validation Loss curve over 50 Epochs for Architecture 2 TF=0.9

Accuracy Metric	TF=0.3		TF=0.6		TF=0.9	
	Test	Validation	Test	Validation	Test	Validation
Exact Match Accuracy	14.59	14.85	**16.155**	**16.54**	11.06	11.21
Execution Accuracy	18.44	18.20	**23.27**	**23.47**	20.0	20.22

Table 3 : Train and Validation Accuracy curve for Architecture 2,Beamsize=10 with TF={0.3,0.6,0.9}

Observations:

1. Compared to TF 0.9 and 0.3, 0.6 performs the best in both Test and validation datasets with two Accuracy metrics.
2. This signifies that a high TF ratio will overfit the model, and it can not be able to infer from unseen data.
3. If the tf is too low as 0.3, the model won't be trained well the intermediate TF ratio of 0.5-0.6 is a good ratio for the best model.

3. Architecture 3 :

A Seq2Seq+Attention model using a pre-trained frozen BERT-base-cased encoder and an LSTM decoder is implemented as part of the Third Architecture.

3.1 parameters and Model details

Encoder: pre-trained frozen BERT-base-cased encoder
Decoder: LSTM Decoder
Decoder Embedding_dimension : 200
Encoder hidden_size : 768
Dropout probability: 0.5
Decoder hidden_size : 128

Teacher Forcing ratio: 0.6
Loss : CrossEntropyLoss
Optimizer: Adam
Learning Rate: 0.001
Epoches: 40

3.2 Results:

Accuracy Metric	Test	Validation
Exact Match Accuracy	28.15	25.86
Execution Accuracy	37.61	34.07

Table 4 : Train and Validation Accuracy curve for Architecture 3, Beamsize=10

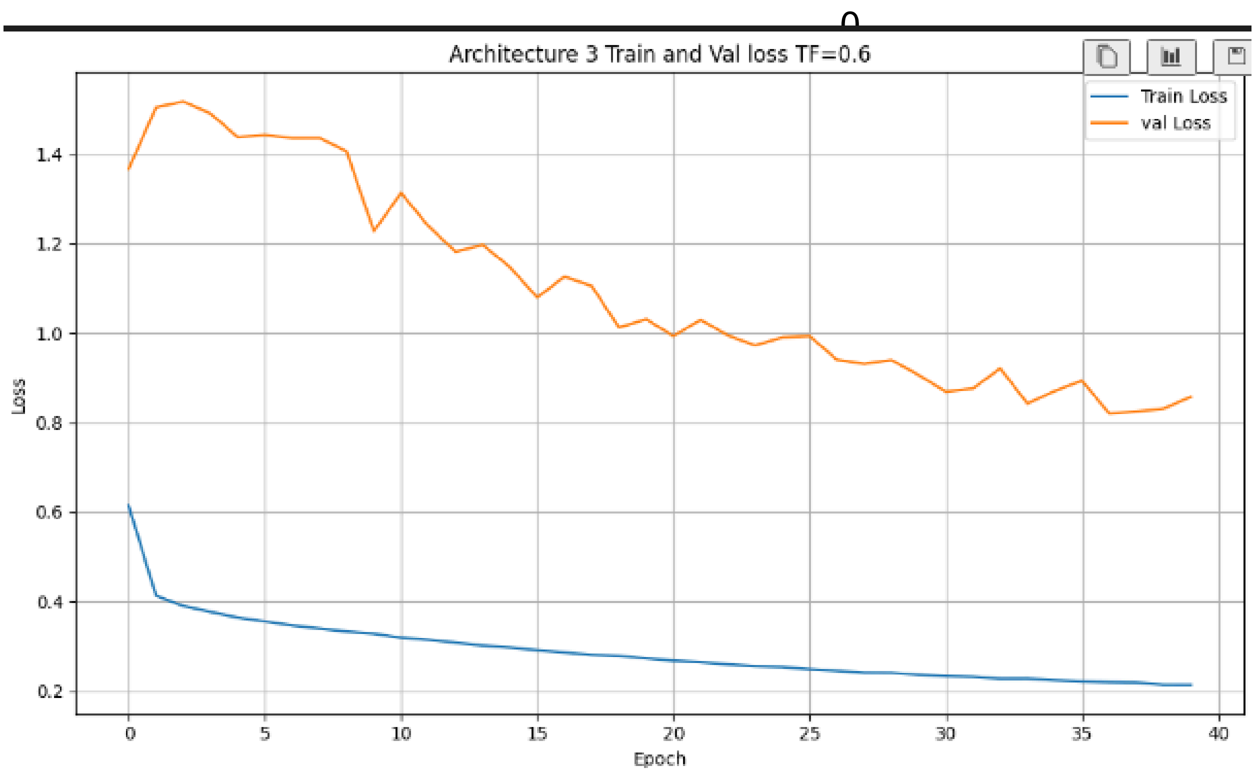

Figure 5 : Train and Validation Loss curve over 40 Epochs for Architecture 3 TF=0.6

4. Architecture 4 :
A Seq2Seq+Attention model using a pre-trained frozen BERT-base-cased encoder and an LSTM decoder and fine-tuning the BERT encoder is implemented as part of the Third Architecture.

4.1 parameters and Model details

Encoder: pre-trained frozen BERT-base-cased encoder
Decoder: LSTM Decoder
Fine-Tuned_Layer(s) : Last 2 Layers of BERT
Decoder Embedding_dimension: 200
Encoder hidden_size: 768
Dropout probability: 0.5
Decoder hidden_size: 128
Teacher Forcing ratio: 0.6
Loss: CrossEntropyLoss
Optimizer: Adam
Learning Rate: 0.001
Epoches: 40

4.2 Results:

Accuracy Metric	Test	Validation
Exact Match Accuracy	20.311	18.98
Execution Accuracy	27.94	26.34

Table 5 : Train and Validation Accuracy curve for Architecture 4, Beamsize=10

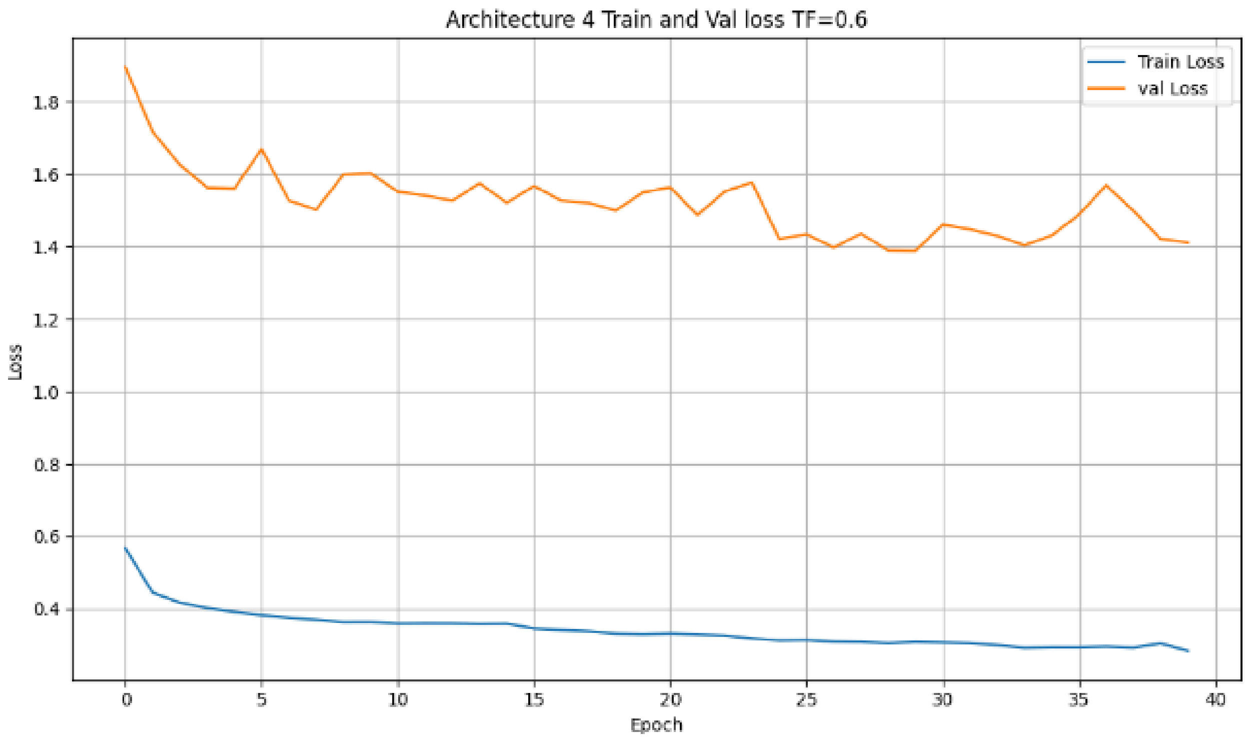

Figure 6: Train and Validation Loss curve over 40 Epochs for Architecture 4 TF=0.6

5.Comparison of models:

5.1 Results:

Accuracy Metric	Model 1		Model 2		Model 3		Model 4	
	Test	Validation	Test	Validation	Test	Validation	Test	Validation
Exact Match Accuracy	22.28	21.37	16.155	16.54	**28.15**	**25.86**	20.311	18.98
Execution Accuracy	28.779	27.86	23.27	23.46	**37.61**	**34.07**	27.94	26.34

Table 6: Train and Validation Accuracy curve for Architecture of all 4 models, Beamsize=10

5.2 Observations:

1. From Table 2, we can infer that the 3rd model which is pre-trained frozen BERT-base-cased encoder and an LSTM decoder performs best.
2. If the fine-tuning is done more accurately on model 4, it may perform better than model 3.
3. Model 1 also performs better because the hidden dimension which is used is high which is 512 .
4. The hidden dimension for the rest of the three models is reduced as there is a memory constraint