COL 775 Assignment 1 Part 2

Malik Hammad Faisal
2021CS10559

April 2024

1 Architectural Experiments

Link to models:
https://drive.google.com/drive/folders/18kT_MJgbbjq2auz1mH3cwGg5SLKUPGao?usp=sharing

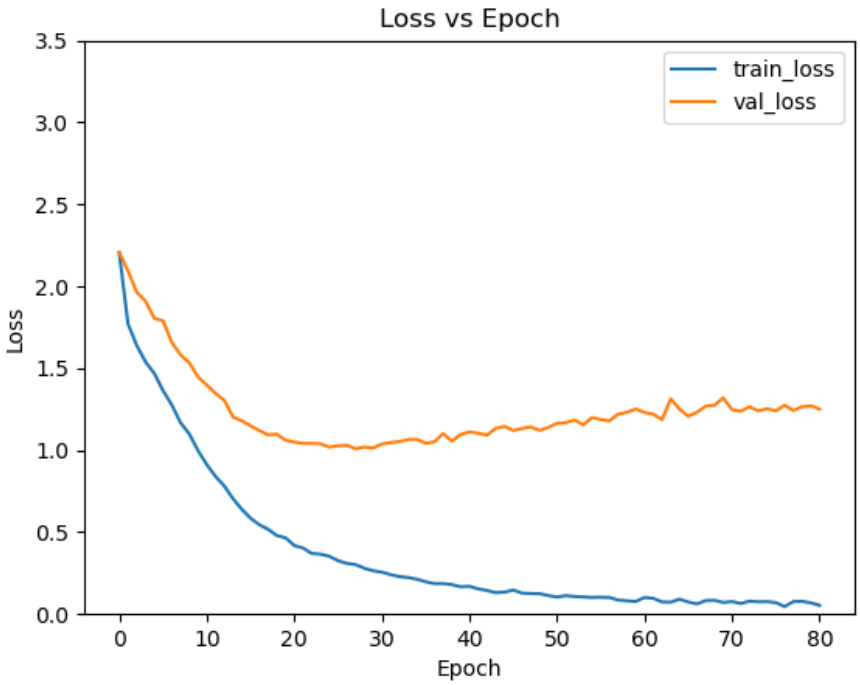
Common settings

- Output embedding dimension: 100
- Adam optimizer with weight decay 0.00001
- Target max size: 32
- Source max size: 64
- Max gradient norm clipped at 15
- Target vocabulary is generated by splitting linear formula in train set on punctuations.
- In glove embedding using models source vocabulary is generated by splitting problems in train set along whitespace. In Bert using models, BertTokenizer is used.

1.1 Seq2Seq with Glove

- Source max size: 64
- Embedding dimension: 100
- Hidden dimension of Encoder: 200
- Hidden dimension of Decoder: 400
- Training batch size: 32
- Learning rate: 0.0005
- Number of layers in both encoder and decoder: 1

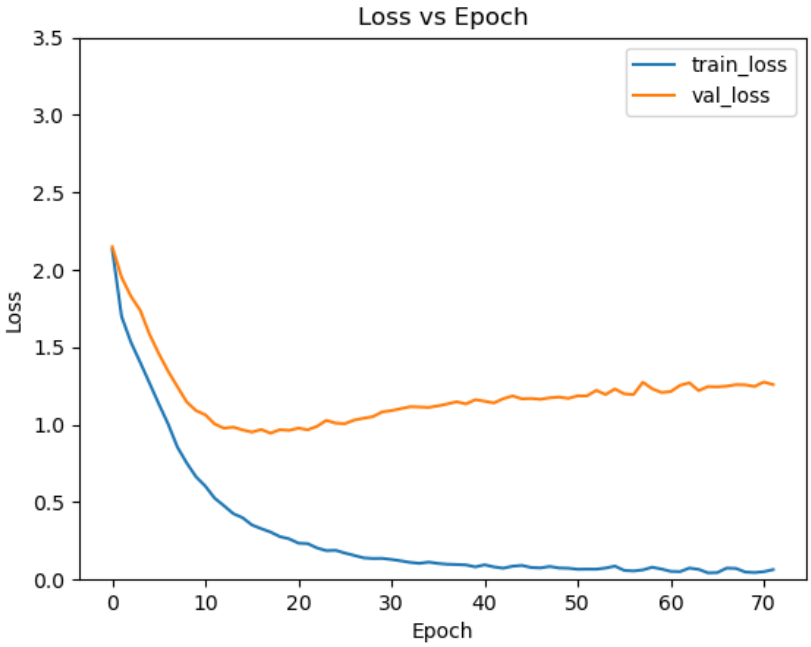
Metric	Dev	Test
Exact Match(%)	69.73	70.23
Execution Accuracy(%)	72.54	73.55



1.2 Seq2Seq + Attention with Glove

- Source max size: 64
- Embedding dimension: 100
- Hidden dimension of Encoder: 200
- Hidden dimension of Decoder: 400
- Training batch size: 32
- Learning rate: 0.0005
- Number of layers in both encoder and decoder: 1

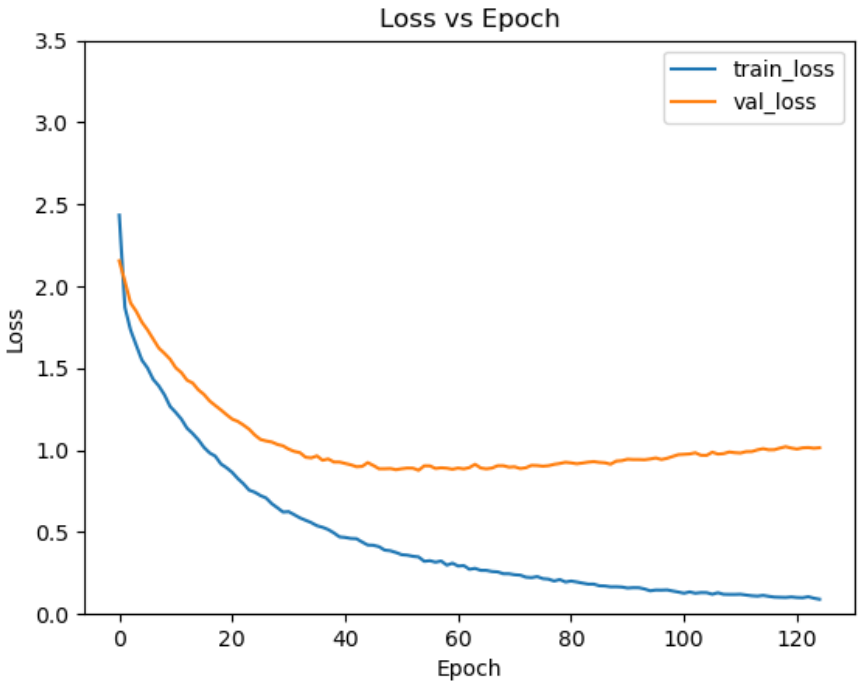
Metric	Dev	Test
Exact Match(%)	71.12	71.42
Execution Accuracy(%)	74.40	74.64



1.3 Seq2Seq + Attention with Bert Encoder Frozen

- Hidden dimension of Decoder: 768
- Training batch size: 32
- Learning rate: 0.00005
- Number of layers in decoder: 1

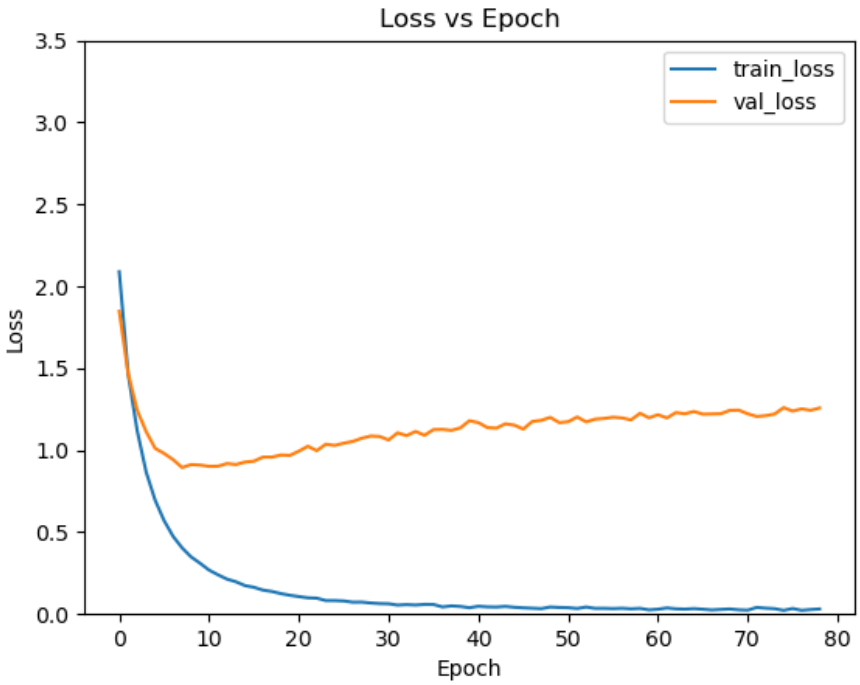
Metric	Dev	Test
Exact Match(%)	69.73	71.89
Execution Accuracy(%)	72.64	74.49



1.4 Seq2Seq + Attention with Bert Encoder Fine Tuned

- Hidden dimension of Decoder: 768
- Training batch size: 32
- Learning rate: 0.00005
- Number of layers in decoder: 1

Metric	Dev	Test
Exact Match(%)	72.54	73.50
Execution Accuracy(%)	75.95	77.55



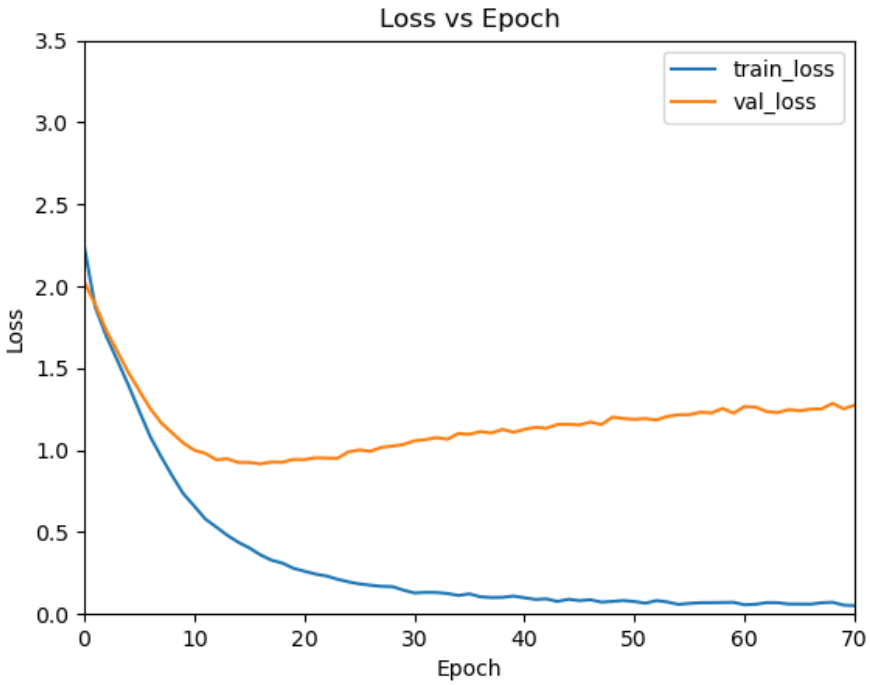
1.5 Observations

- The stopping rule for training was based on greedy decoding accuracy. The training ends when the best accuracy on dev set does not increase for 15 epochs. The loss curve reaches its minima much before accuracy reaches its maxima so the accuracy increases slightly even when the loss starts increasing for all models.
- The embedding and hidden dimensions were kept low because of the small vocabulary sizes.
- Among the first two models, the one with attention converged faster and also gives slightly better accuracies.
- The first two models were robust to change in learning rates. I tried training with 10x the rate here and it also had similar results.
- Models using Bert were less robust to higher learning rates and accuracy of Bert with fine tuning based model dropped to 0 when learning rates of the order of 5e-4 was used.
- The problem statements were padded or truncated to same limit for Bert. When this was not done, during inference with batch size 1, the performance became much worse.
- Overall, Bert with fine tuning was the best performing model.

2 Effect of Teacher Forcing Probabilty

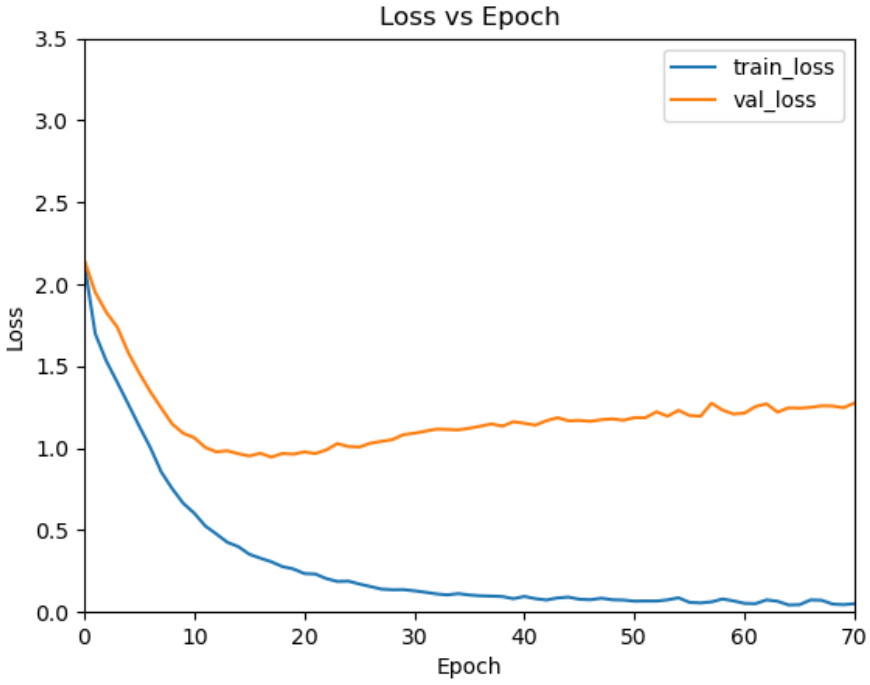
2.1 Probability 0.3

Metric	Dev	Test
Exact Match(%)	71.66	72.83
Execution Accuracy(%)	75.34	75.74
Token Match (%)	82.82	83.09



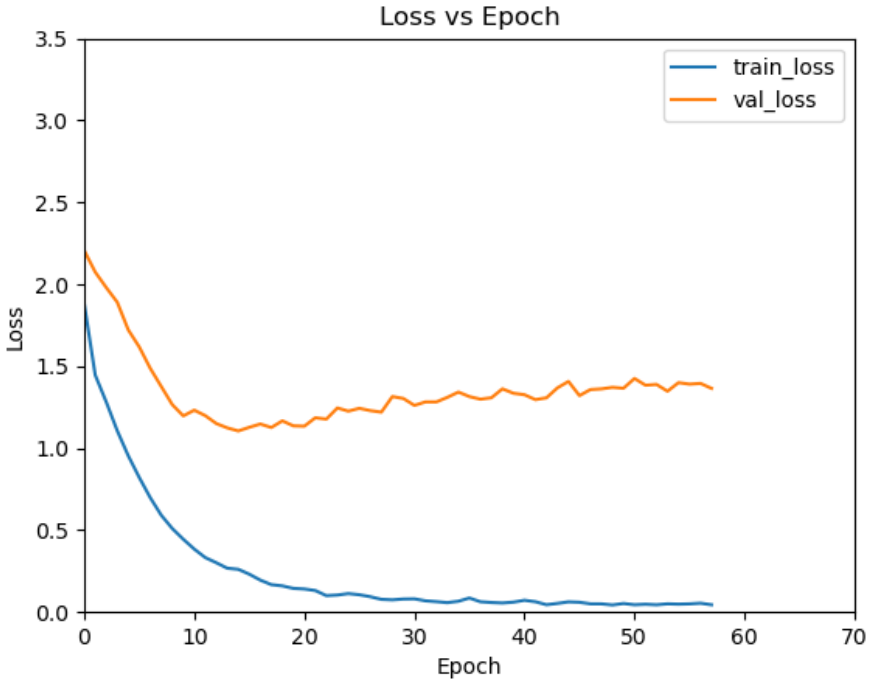
2.2 Probability 0.6

Metric	Dev	Test
Exact Match(%)	71.12	71.42
Execution Accuracy(%)	74.40	74.64
Token Match (%)	82.20	82.59



2.3 Probability 0.9

Metric	Dev	Test
Exact Match(%)	71.15	71.63
Execution Accuracy(%)	74.87	74.80
Token Match (%)	81.35	82.04



2.4 Observations

- The loss curves on dev and train set remain together for longest time in probability 0.3 and diverge fastest in probability 0.9.
- Probability 0.9 model converges fastest even though its performance is slightly short of probability 0.3 one which has the best accuracies.

3 Effect of Beam Size

3.1 Beam Size 1

Metric	Dev	Test
Exact Match(%)	72.54	73.55
Execution Accuracy(%)	76.02	77.50
Token Match (%)	85.24	85.23

3.2 Beam Size 10

Metric	Dev	Test
Exact Match(%)	72.54	73.50
Execution Accuracy(%)	75.95	77.55
Token Match (%)	85.15	85.02

3.3 Beam Size 20

Metric	Dev	Test
Exact Match(%)	72.47	73.40
Execution Accuracy(%)	75.88	77.45
Token Match (%)	85.22	85.13

3.4 Observations

- There is no significant difference among any of three beam size results.
- This may be because the target vocabulary is already very small(140) and if the model at least understands the difference between arguments and operations, then the possible tokens for a position given the past reduces further.
- Because of the small target vocabulary, there are not many diverse options for beam search to explore.