NLU Coursework 1 Report

s2447930, s2602230

February 2024

Task 2a

LR	Hidden Units	Steps	Loss	New LR
0.5	25	0	5.0170	0.1786
0.1	25	0	5.2190	0.0357
0.05	25	0	5.4077	0.0179
0.5	25	2	5.0351	0.1786
0.1	25	2	5.2145	0.0357
0.05	25	2	5.4038	0.0179
0.5	25	5	5.0180	0.1786
0.1	25	5	5.2146	0.0357
0.05	25	5	5.4038	0.0179
0.5	50	0	4.9731	0.1786
0.1	50	0	5.1329	0.0357
0.05	50	0	5.3130	0.0179
0.5	50	2	5.0292	0.1786
0.1	50	2	5.1366	0.0357
0.05	50	2	5.3141	0.0179
0.5	50	5	5.0210	0.1786
0.1	50	5	5.1368	0.0357
0.05	50	5	5.3142	0.0179

Table 1: How the loss and last learning rate (LR) of the recurrent neural network (RNN) vary with different hyper-parameters. Hyper-parameters tested are initial learning rate, number of hidden units and number of truncated back-propogation through time steps (BPTT)

The results show that higher learning rates result in models with a lower cross entropy loss when all other hyper-parameters are the same. This is likely because the number of epochs is low and higher learning rates allow faster conversion towards a local minima. Also, increasing the number of hidden units always improves the loss, as the model has more representation power to learn better features. In the majority of cases, increasing the number of truncated BPTT steps increases the model loss, possibly as lower step numbers encourage focus on short range dependencies, although differences are small and could be down to chance. The best model has a LR of 0.5, 50 hidden units and 0 truncated BPTT steps.

Task 2b

Results of best model:

- Development set final loss: 4.4156
- Test set mean Loss: 4.4264
- Test set adjusted perplexity: 112.902
- Test set unadjusted perplexity: 83.631

Task 3c

Model	Hidden Units	Accuracy
RNN	10	71.6
RNN	25	72.6
RNN	50	74.0
GRU	10	71.8
GRU	25	75.7
GRU	50	78.1

Table 2: Comparison of RNN and gated recurrent unit (GRU) models on the number prediction task with varying numbers of hidden units

Overall, the GRU performs better than the RNN model, with a higher accuracy at every hidden unit size. The difference in accuracy becomes greater as the number of hidden units increases. Moreover, the accuracy increases faster for the GRU models compared to the RNN models when incrementing the hidden units.

Task 4

Steps	Loss	Accuracy (%)
1	0.48071	76.8
2	0.47657	77.1
3	0.47522	77.2
4	0.47553	77.3
5	0.47538	77.2
6	0.47537	77.1
7	0.47530	77.1
8	0.47528	77.1
9	0.47525	77.1
15	0.47524	77.1
50	0.47524	77.1

Table 3: RNN Model Performance across different number of truncated BPTT steps

Steps	Loss	Accuracy (%)
1	0.57631	79.3
2	0.57344	79.2
3	0.56570	79.4
4	0.56303	79.5
5	0.56173	79.6
6	0.55928	79.4
7	0.55854	79.4
8	0.55812	79.6
9	0.55928	79.6
15	0.55500	79.8
50	0.55463	80.1

Table 4: GRU Model Performance across different number of truncated BPTT steps

Increasing the number of truncated BPTT steps initially increases accuracy for the both the RNN and GRU model as they are able to better use context and model dependencies that are further apart. However, when the number steps is increased beyond four the RNN model performance starts to decrease, likely because the model is not incorporating information from long range dependencies well due to vanishing gradients. Unlike the RNN the GRU performance keeps increasing when the number of steps is increased even further.

This is because the gating mechanisms enable GRUs to retain information better and model long distance dependencies. The combined effects of the reset and update gate allow the model to select which information to keep at each time step, this means important information from time-steps far in the past can flow through unchanged. This is especially important in this task, where the dependencies can be arbitrarily long. This leads to increased performance of GRUs, especially when the number of steps is high, with the difference being 2.5% with 1 step and 4.0% with 50 steps.

Task 5

Hypothesis: RNNs performance drops much faster compared to the GRUs performance when dependency distance increases.

It is well known that GRUs are better at capturing long range dependencies than RNNs, but we wanted to investigate how exactly the performance of each model varied with dependency distance, and whether GRUs are still better than RNNs at capturing closer range dependencies, or whether their additional architecture elements only help over long ranges.

Analysis done in earlier questions leads us to believe that the majority of dependencies for the number prediction task in this dataset are very short range. This is evidenced by the modest improvement of GRUs over RNNs, being between 2% and 3%, despite RNNs being known to be bad a modelling long range dependencies. Furthermore, increasing the number of BPTT steps in task 4 leads to only slight accuracy improvements of less than 1% in both the RNN and GRU model.

We can test this insight by looking at the difference in the subject and verb index in the training data in Fig 1. Looking at the distribution of these differences we can see that as suspected most dependences are very short range, with around two-thirds of dependencies being distance one, note the graph is on a log scale.

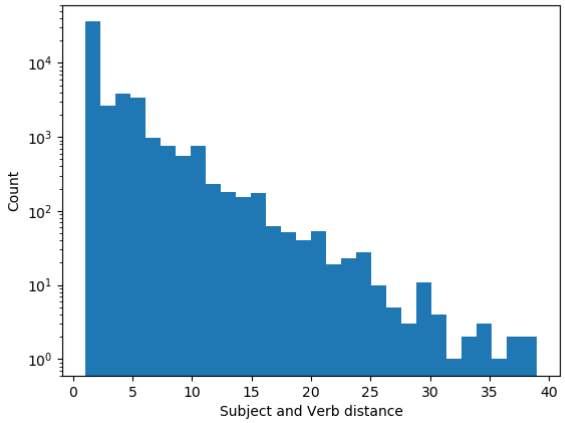

Figure 1: Graph showing distribution of subject and verb distance in the training dataset. As the distance increases, the count decreases exponentially.

To check whether the models are actually performing better on lower distance samples, we train two RNN and GRU models with identical hyper parameters - 0.5 LR, 50 hidden units, 10 epochs. We test models with both zero and five BPTT steps. We then check the accuracy of each model on samples of the development set with different subject and verb distance splits. As shown

in Fig. 2 and Fig. 3 the RNN model performs considerably better on the lower distance than high distance examples, with the gap increasing as the distance of dependencies increases, whereas the GRU performance is far more similar and stable across different distances.

Even on short range dependencies, GRUs usually outperform RNNs, but the difference is much less pronounced. This means that for tasks involving shorter range dependencies RNNs might be the better choice due to their much smaller memory consumption and faster runtime. However, for tasks involving longer range dependencies RNN performance drops off very fast, and GRUs are much more suitable.

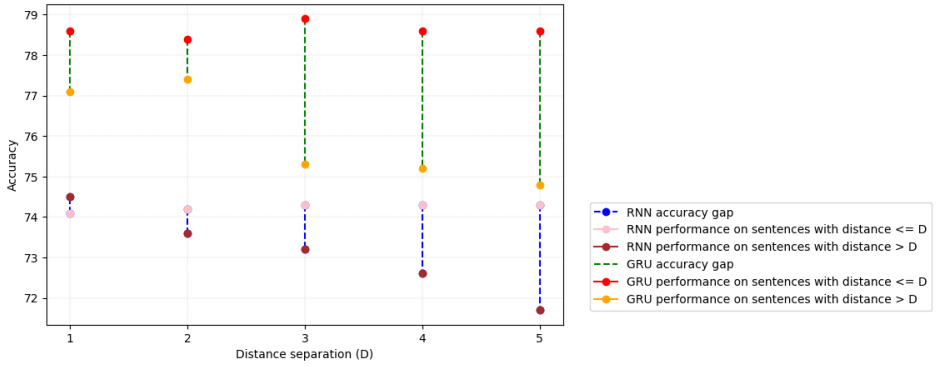

Figure 2: Graph visualising the accuracy gap based on the distance between the subject and the verb (D) with 0 BPTT steps. Each dot is accuracy on a subset of the development set either above or below a distance threshold, with the gap shown by a dotted line.

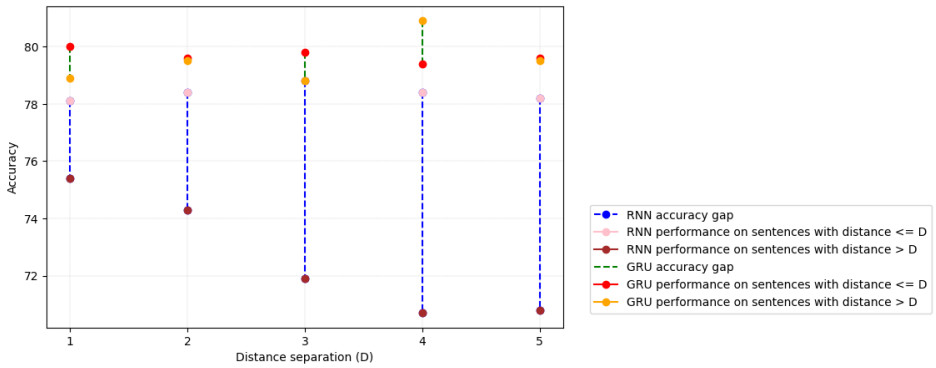

Figure 3: Graph visualising the accuracy gap based on the distance between the subject and the verb (D) with 5 BPTT steps.