Natural Language Understanding, Generation, and Machine Translation (2021–22)

Coursework 1: Recurrent Neural Networks

Question 1: Training RNNs

See code.

Question 2: Language Modeling

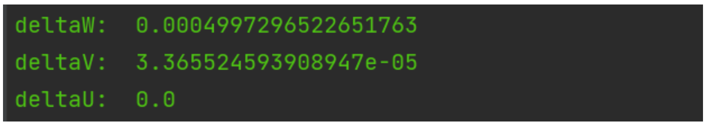
(a)

Hidden units	Look back	Learning rate	Loss
25	0	0.5	5.017
25	0	0.1	5.219
25	0	0.05	5.408
25	2	0.5	5.035
25	2	0.1	5.215
25	2	0.05	5.404
25	5	0.5	5.018
25	5	0.1	5.215
25	5	0.05	5.404
50	0	0.5	**4.973**
50	0	0.1	5.133
50	0	0.05	5.313
50	2	0.5	5.029
50	2	0.1	5.137
50	2	0.05	5.314
50	5	0.5	5.021
50	5	0.1	5.137
50	5	0.05	5.314

Table 1: Loss for different parameters in RNN when making predictions at every time step. **bold** indicate the best loss.

We explored the parameter settings of the model to maximize generalization performance. Parameters include: 1)number of hidden units: 25 or 50. 2)look back steps: 0, 2, or 5. 3)learning rate: 0.5, 0.1, or 0.05. The results are showed in Table 1.

According to the table, the best performance was achieved when the number of hidden units was 50, the look back step was 0 and the learning rate was 0.5. The performance was better when the number of hidden units was 50 than 25. This is because the sigmoid function is used as the activation function in the hidden layer, and assuming the same rate of inactivation of units, when there are more hidden units, more units are activated and therefore the model performs better. Furthermore, we found that with 50 hidden units, the model performed better with 0 look back steps than 5 look back steps. We suspect that this is because of the vanishing gradients problem. Then, we took 50 hidden units, 5 look back steps and a learning rate of 0.1 as parameters and checked the gradient. After taking the absolute value of the gradient matrix and then taking the mean value, we could find that ΔU became 0, as shown in Figure 1, confirming the existence of the vanishing gradients problem. In addition, we discovered that the model performed best with the largest 0.5 learning rate. The loss continued to drop during training process and all of the best losses were obtained at epoch 10. This means the cross-entropy loss function has not completely dropped to the minimum, thus a relatively large learning rate can help the function converge to the minimum within 10 epochs.



```
deltaW:  0.0004997296522651763
deltaV:  3.365524593908947e-05
deltaU:  0.0
```

Figure 1: Vanishing gradients problem.(hidden units:50, look back steps:5, learning rate:0.1)

(b) Based on the analysis in the previous question, we selected 50 hidden units, 0 look back steps and a learning rate of 0.5 as parameters and used a larger training set to test the performance of the model. The model was trained on 25,000 sentences and the vocabulary size is 2000. Table 2 shows the evaluation results.

	Criteria	Value
Dev set	Mean loss	4.416
	Adjusted perplexity	111.613
	Unadjusted perplexity	82.787
Test set	Mean loss	4.427
	Adjusted perplexity	112.987
	Unadjusted perplexity	83.687

Table 2: Evaluation on the dev set and the test set

Question 3: Predicting Subject-Verb Agreement

(a) See code.

(b) We explored the parameter settings of the RNN model that makes predictions in the final time step. Parameters include: 1)number of hidden units: 25 or 50. 2)look back steps: 0, 2, or 5. 3)learning rate: 0.5, or 0.1. The results are showed in Table 3.

According to Table 3, the best performance was achieved when the number of hidden units was 50, the look back step was 5 and the learning rate was 0.5. The model with 5 look back steps performs better than the model with 2 look back steps, because it can capture longer linguistic dependencies. We calculated the average distance between subject and verb in the dataset, with a value of approximately 2.46826 in the training set and 2.588 in the dev set. Therefore, the linguistic dependencies are not well captured when look back steps is 2, which confirms our results. Based on the analysis in the previous question, we selected 50 hidden units, 5 look back steps and a learning rate of 0.5 as parameters. Then, we evaluate the model trained on 25,000 sentences with the vocabulary size is 2000. The evaluation results are shown in the Table 4.

Hidden units	Look back	Learning rate	Loss	Accuracy
25	0	0.5	0.645	0.658
25	0	0.1	0.695	0.659
25	2	0.5	0.645	0.659
25	2	0.1	0.692	0.659
25	5	0.5	0.645	0.659
25	5	0.1	0.693	0.659
50	0	0.5	0.645	0.669
50	0	0.1	0.679	0.669
50	2	0.5	0.642	0.669
50	2	0.1	0.678	0.669
50	5	0.5	**0.641**	**0.669**
50	5	0.1	0.677	0.669

Table 3: Loss for different parameters in the model when making predictions at the final step. **bold** indicate the best performance.

	Criteria	Value
Dev set	Loss	0.411
	Accuracy	80.9%
Test set	Loss	0.408
	Accuracy	82.6%

Table 4: Evaluation on the dev set and the test set

Question 4: Number Prediction with an RRNLM

RNNLM is implemented for the task in question 3 with 67.5% prediction accuracy on the dev set and 65.125% prediction accuracy on the test set, indicating that the model is suitable for this task. However, the accuracy is reduced compared to the method of using direct supervision in question3, suggesting that the use of supervised learning can significantly improve the performance of the model, and that RNNLM needs to be tuned to the task in order to achieve good performance.

(Report continues on next page.)

Question 5: Exploration

In this section, we continue to explore the RNN model and the vanishing gradients problem. The questions include: 1)How to make modifications on the RNN to address the vanishing gradients problem? 2)How to choose the best parameter in this method? 3)Implement the tasks in question2 and question3, and analyze the results.

For the first question, We use gradient clipping to modify the RNN model. Specifically, when the gradient is less than a certain threshold, the model will update the gradient to the threshold to avoid the gradient from vanishing. In question 2, we confirmed that vanishing gradients problem occurs when the number of hidden units is 50, the number of look back steps is 5, and the learning rate is 0.5. Therefore, we use the model with these parameters as the baseline and perform gradient clipping with different thresholds on its basis. The models are trained on 1,000 sentences with a fixed vocabulary size of 2000. Table 5 shows the results of gradient clipping under different thresholds.

Threshold	0.0001	0.000055	0.00005	0.000045	0.00003	Baseline
Loss	5.042	4.976	**4.965**	4.968	4.999	5.021

Table 5: Comparison of models after gradient clipping at different thresholds.(hidden units=50, look back steps=5, learning rate=0.5, epoches=10) **bold** indicate the best performance.

According to Table 5, it can be found that the performance of the model has been improved after performing gradient clipping. When the threshold is selected as 0.00005, we can obtain the minimum loss, and it is lower than the minimum loss 4.973 in question2(a). Then, we use this threshold to train the new model on a much larger training set. The training set contains 25,000 sentences with a vocabulary size of 2,000, and the first 1,000 sentences in the development set is used to evaluate the performance. Table 6 shows the evaluation results. The results on both the dev set and the test set are better than those in question2(b).

	Criteria	Value
Dev set	Mean loss	4.414
	Adjusted perplexity	111.429
	Unadjusted perplexity	82.632
Test set	Mean loss	4.424
	Adjusted perplexity	112.754
	Unadjusted perplexity	83.543

Table 6: Evaluation on the dev set and the test set

To explore the performance of the model, we implement it for the task in question3. Using the trained model to make predictions, we get 67.2% prediction accuracy on the dev set and 65.325% prediction accuracy on the test set. The new model performs worse than the model in question4(hidden units=50, look back steps=0, learning rate=0.5) on the dev set and better than the model in question4 on the test set.

Therefore, we considered to further analyze the performance of the model to investigate whether the performance of the model is improved after solving the vanishing gradient problem. We take sentences with different subject-verb distances from the dev set for testing. The results are shown in the Table 7. Although the prediction accuracy of the model on the dev set has dropped, we can find from Table 7 that the performance of the model is improved when the subject-verb distance is 4 or 5, and the accuracy of the model drops sharply when the subject-verb distance is 6. This is because RNN without vanishing gradients is able to capture all linguistic dependencies in the range of look back steps.

Subject-verb distance	1	2	3	4	5	6
look back=0	0.688	0.596	0.696	0.712	0.655	0.667
Our model(look back=5)	0.679	0.596	0.661	**0.726**	**0.724**	0.583
Sample proportion	65.9%	5.3%	5.7%	7.4%	3.0%	2.5%

Table 7: Accuracy in different Subject-verb distance sentences. **bold** indicate improved performance.

From the results, we can confirm that performing gradient clipping can address the issue of gradient disappearance, and after solving the vanishing gradients problem, the model can be improved within the scope of the look back, especially when the Subject-verb distance is close to the look back steps.