NLU – Recurrent Neural Networks Coursework

1. Training RNNs

The question is implement in `rnn.py` and it does not require a writeup.

2. Language Modeling

(a) Choice of Hyper-parameters

We conducted tuning of hyper-parameters in two steps:

(i) First, we performed a grid-search with the recommended values: hidden units 25 and 50, look-back steps 0, 2 and 5, and learning rate 0.5, 0.1 and 0.05.

(ii) Second, based on our findings, we performed: 1) a *targeted* search where we selected the best-performing combinations and adjusted the parameters by a small magnitude, and 2) a *wild* search where we ventured into extreme values. In total, we tried 140 combinations.

Hyper-parameter	Value
Hidden Units	**50**
Back-propagation Steps (Look-back)	**2**
Learning Rate	**1.5**

Table 1. Final choice of parameters for the language modeling task.

We found combinations of many hidden units and a slow learning rate to underperform, unsurprisingly. This matches our expectations. More complex architectures, of several layers and many hidden units, generally thirst for longer training times and more data points. Additionally, our experiments were conducted on just 10 epochs of training and we found a high learning rate to be preferred as it allows faster reaching of the vicinity of the local minimum. Indeed, the validation error and generalization to the test dataset improved significantly when for a basis of comparison we trained over 50 epochs. We also found high values of lookback to underperform suggesting that long context does not help predict the next word, perhaps because the additional information confuses the network. We investigate this hypothesis further in Question 4 (b).

(b) Evaluation

Table 2. reports results with the chosen hyper-parameters. The final learned matrices are submitted as part of the assignment.

Metric	Dev set	Test set
Mean Loss	4.4234	**4.4293**
Unadjusted Perplexity	83.84	**83.88**
Adjusted Perplexity	112.53	**113.28**

Table 2. Results from evaluation on the test dataset.

3. Predicting Subject-Verb Agreement

(b) Hyper-parameter Tuning & Results

The hyper-parameters for the number prediction task were optimized analogously to the procedure described in Question 2 (a). We ran 140 experiments and report our final choice in Table 3.

Hyper-parameter	Value
Hidden units	**75**
Look-back in back-propagation	**2**
Learning rate	**2**

Table 3. Best parameters for binary classification model.

4. Number Prediction with an RRNLM

(a) Results

Dataset	Accuracy
Development	**62.8%**
Test	**57.3%**

Table 4. Evaluation results on a number prediction task of the RNNLM implemented in Question 1 and trained in Question 2.

(b) 1. Observations

The number prediction task, or subject-verb agreement, should be a simple task, particularly in English. Why would a model not learn to use a verb's inflected form after subjects ending in *-s*, and then learn to recognize the few exceptions? We hypothesize that the neural network which we implemented is sufficiently complex to predict with higher accuracy than observed the correct number of a verb (singular or plural), given a noun. We suspect that performance suffered partly due the difficulty in identifying the head of the sentence. In other words, since there can be a varied number of words between the subject and the verb, the model struggles to determine which word the verb must agree with.

To better understand context, we plotted a histogram which groups the training sentences into one of 10 categories based on the difference between the indices of the subject and verb position (Figure 1). Clearly, in most cases, the verb and subject are next to one another in the sentence. This shows that our data is highly skewed and can explain why multiple look-back steps resulted in under-performance, as reported in Question 2 (a). The model can get all *difficult* sentences wrong and still achieve a high overall accuracy.

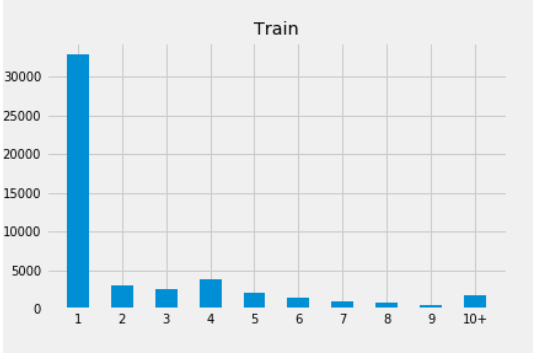

Figure 1. Histogram of distance between the indices of subject and verb in the training dataset. The distribution in the test and the development datasets is similar.

With the observation in our pocket, we designed a few simple experiments to get intuition how much the incorrect number agreement can be attributed to the inability of the model to recognize the head of the syntactic subject.

(b) 2. Empirical Experiments

We took the setup from Question 3 as a baseline (*A*) and designed three additional experiments (*B*, *C* and *D*). The experiments varied the length and the difficulty of *training* input.

Please note that regardless of varied *training* strategies, all models were *evaluated* under the same conditions, as specified in Question 3 for the number prediction task with the exception of training to 25 epochs.

Experiment *A*, baseline, uses all words from the beginning of the sentence up until the position of the verb index. Experiment *B* uses the words from the subject index (inclusive) until the verb index. We expected this experiment to achieve better accuracy than A as we removed the words at the start of the sentence which could disrupt the prediction. In Experiment *C*, we tested how well the network will do if we just give it a noun, the subject, and ask it to predict whether it is plural or singular. In that experiment, the RNN functioned as a feed-forward network. We expected this experiment to yield the best predictions. Finally, for *D*, we wondered if training on *difficult* sentences only, defined as sentences in which the subject and the verb are at least 5 words apart, the learned weights will generalize well to the overall dataset, in which subject and verb are typically adjacent.

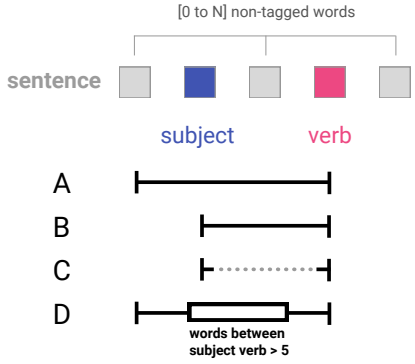

Figure 2. Experiment word window context length comparison

Length of Training Context:

Exp. A All words prior the verb

Exp. B All words between subject and verb (incl. subject)

Exp. C Subject only

Exp. D All words prior the verb. However, only *difficult* sentences were used for training, that is where subject and verb are separated by 3 or more words

(b) 3. Results

This section summarizes the loss and accuracy which we achieved at each experiment, after hyper-parameter optimization and final training on the best combinations.

To our surprise, model *A* scored best, as we had expected superior performance from *B*. Reflecting back, we suspect that *B* underperformed because it learned naively to agree the verb with the first word in the context. During evaluation that was often but not always the case, so *A* generalized

Metric	A	B	C	D
Accuracy (pct.)	88.5	82.4	79.1	71.6
Best Loss	0.278	0.444	0.464	0.601

Table 5. Evaluation results from the four experiments on the test dataset.

better. We see similar loss and accuracy in experiments *C* and *B*. Experiment *D* was arguably the most difficult. We provided the network with only sentence examples of increased difficulty in inspiration of the approach by Linzen (Linzen et al., 2016), but were unable to replicate their results of accuracy increase due to over-sampling of difficult cases.

Lastly, we wanted to get intuition what the data looks like and how easily separable it is. We visualized the weights learned by the hidden layer by activating it with 1000 samples and then reducing dimensionality to two 'principal components.' We opted for t-SNE rather than other dimensionality reduction approaches such as PCA and SVD because of t-SNE's nonlinearity which plots in low dimensions more realistically the true distance of data-points. (van der Maaten & Hinton, 2008).

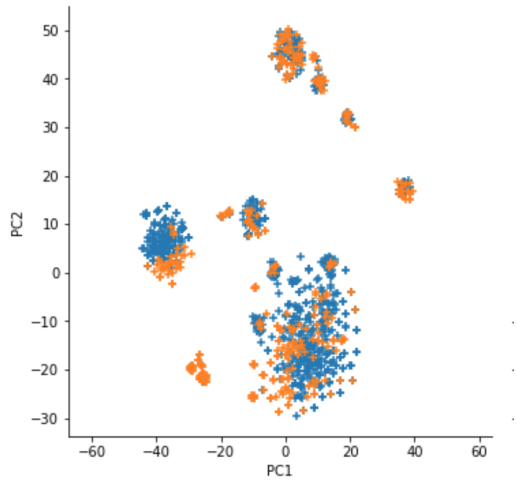

Figure 3. t-SNE activations of the hidden layer in Experiment C. The red and orange labels indicate the true number of the verb, VBZ or VBP. We notice several clusters and speculate that they may correspond to number of words separating the verb from the subject in the data. The model clearly misclassifies the verb number of the sentences clustered in top right, while we see two larger blobs in the center bottom and middle left which can be . The plots from the other experiments did not look too dissimilar so we are not including them with this report.

(b) 4. Limitations

(i) *Subject always proceeds verb in the dataset.* We noticed that our dataset has been filtered to include sentences in which the subject always precedes the verb. That ordering tends to be the norm but it is not always the case in the English language, i.e. questions or inverted sentences. In those scenarios, a RNN which looks for context prior to the verb only does not learn enough information to correctly predict a verb's plurality. In the **Next Steps** section we propose bi-directional recurrent neural networks as one possible remedy.

(ii) *Verb is always presented in last position.* In a grammar correction application, the network would need to learn to spot subject-verb disagreement without a tagged subject or verb. This extends the current problem definition. One possible ensemble solution can incorporate a Part-of-Speech Tagger (POT) to find the root of the sentence which will support our architecture.

(b) 5. Next Steps

(i) Bi-directional recurrent neural networks (BRNN) could be an extension of the architecture implemented in this coursework. A BRNN is unique in that it can capture context in a backward fashion, as illustrated in Figure 4.

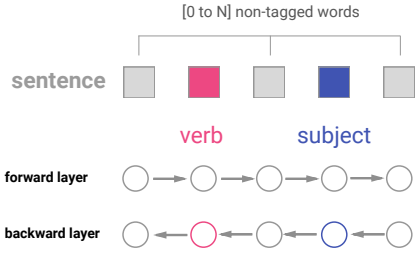

Figure 4. BRNN layer structure where the backward layer can represent subject post verb.

(ii) Long short-term memory networks (LSTM) are also known to be outperforming recurrent neural networks in grammar learning tasks (Linzen et al., 2016). A RNN is not able to hold a longer context as gradients of previous time steps quickly vanish. An LSTM's gated cell state structure allows for information to pass further down the sentence.

References

Linzen, Tal, Dupoux, Emmanuel, and Goldberg, Yoav. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Ieee Signal Processing Magazine*, cs.CL, 2016.

van der Maaten, Laurens and Hinton, Geoffrey. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, November 2008.