Lab 09: Vector Semantics
Part – I: Language Model in Recurrent Neural Network

Adnan Irshad, University of Trento
adnan.irshad@unitn.it

1. Introduction

The objective of this lab exercise was to implement and evaluate various recurrent neural network (RNN) language models. The focus was on exploring different model architectures and training configurations to understand their impact on language modeling performance from pre-trained models like genism to training own deep learning in RNN. Additionally, we referred to two research papers, "Recurrent Neural Network Based Language Model" by Mikolov et al. (2010) and "Extensions of Recurrent Neural Network Language Model" by Mikolov et al. (2011), to guide our implementation and gain insights from established approaches in the field. The goal was to investigate the effects of three modifications on the language modeling performance: A. Replace RNN with LSTM; B. Add two dropout layers: one on embeddings and one on the output; C. Replace SGD with AdamW optimizer. The experiments were focused on minimizing perplexity (PPL) and evaluating the performance of the model with the best configuration and the best model with **177.57** PPL were recorded during the experiments.

2. Baseline Model: LM_RNN

The baseline model was a PyTorch implementation of a language model. It consisted of an embedding layer, an RNN layer, and a linear projection layer which provides word embedding retrieval and similarity calculation. The size of the output vector is equal to the size of the vocabulary, meaning the model cannot predict tokens that are not present in the vocabulary. After training the baseline model has the the test ppl: 316.92.

In the RNN-based language model, the task treated as a sequence labeling problem, where the input sequence was the previous context, and the output was the next word. For example, given the sentence "I go to Miami," the input sequence would be "I go to," and the output sequence would be "go to Miami."

3. Improvements Made

To enhance the baseline model's performance, several improvements were implemented which was the actual tasks of Exercise 1 of Lab-9:

3.1 Replacement of RNN with LSTM: The RNN layer in the LM_RNN model was replaced with an LSTM (Long Short-Term Memory) layer. The LSTM architecture provided enhanced capability to capture long-range dependencies in the input sequence & alleviate the vanishing gradient problem.

3.2 Addition of Dropout Layers: Two dropout layers with the probability 0.1 were added to the LM_RNN model:

- **Embedding Dropout:** A dropout layer was applied to the embeddings before feeding them into the LSTM layer. This helped in regularizing the model and preventing overfitting.
- **Output Dropout:** Another dropout layer was added to the output of the LSTM layer. This further enhanced regularization and generalization of the model.

3.3 Replacement of SGD with AdamW: The optimization algorithm used in the baseline model, Stochastic Gradient Descent (SGD), was replaced with the AdamW optimizer. AdamW combined the benefits of Adaptive Moment Estimation (Adam) and weight decay, leading to more effective parameter updates and improved convergence.

4. Experiments and Results

The modifications made to the baseline LM_RNN model were evaluated through experiments to minimize perplexity (PPL). The PPL metric measures the model's ability to predict the next word in a sequence. The lower the PPL, the better the model's performance.

The experiments were conducted with different hyperparameter configurations to find the best-performing model. The PPL values were recorded for each experiment to assess the impact of the improvements on the model's performance. The results achieved with the best configuration were reported.

The experimental results of the improved model in vector semantic modeling are summarized below:

For the SGD optimizer:

- Exp #1: With Hyperparameters (lr 0.0001, epochs=100, hidden size=200, embedding size=300) got the test PPL of 9725.15
- Exp #2: With Hyperparameters (lr 0.1, epochs=100, hidden size=50, embedding size=100) got the test PPL of 419.15
- Exp #3: With Hyperparameters (lr 0.1, epochs=100, hidden size=200, embedding size=300) got the test PPL of 222.34

For the AdamW optimizer:

- Exp #1: With Hyperparameters (lr 0.01, epochs=100, hidden size=200, embedding size=300) got the test PPL of 740.95
- Exp #2: With Hyperparameters (lr 0.001, epochs=100, hidden size=200, embedding size=300) got the test PPL of 181.15
- Exp #3: With Hyperparameters (lr 0.0001, epochs=100, hidden size=200, embedding size=300) got the test PPL of 171.15

Best model was found with AdamW optimizer with LR 0.0001, hidden size 200 and Embedding size 300 with probability 0.1 for both.

5. Conclusion

In this lab, I successfully improved the baseline LM_RNN model for vector semantic modeling by incorporating various enhancements. Replacing the RNN with LSTM, adding dropout layers on embeddings and output, and utilizing the AdamW optimizer led to improved performance. The experiments conducted and the reported perplexity values provided insights into the effectiveness of the improvements. These enhancements contribute to the advancement of vector semantic modeling and provide a foundation for further research in the field.

6. References

- *https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5947611&tag=1*
- *https://www.fit.vutbr.cz/research/groups/speech/publi/2010/mikolov_interspeech2010_IS100722.pdf*