NLU course projects: lab 4

Simone Roman (mat. 247181)

University of Trento

simone.roman@studenti.unitn.it

1. Introduction

This assignment aims to improve the performance of a basic language model by incrementally incorporating advanced techniques. Starting from a Long Short Term Memory (LSTM) network that replaces the basic Recurrent Neural Network (RNN), I introduced two dropout layers for regularization. Next, I switched from Stochastic Gradient Descent (SGD) to the Adaptive Moment Estimation (AdamW) optimizer. To further improve the model, I applied advanced regularization techniques such as weight tying, variational dropout and non-monotonically activated Avarage Stochastic Gradient Descent (AvSGD). Each modification was tested by Calculating the Perplexity (PPL) to gain detailed information on the effectiveness of each technique

2. Implementation details

The implementation of the first part of the assignment involved three main modifications to the original model: switching from RNN to **LSTM**, adding **Dropout Layers** and switching from SGD to **AdamW**. For each modification, I tuned the hyperparameters to achieve optimal performance.

For the second part of the assignment, I started with the initial optimization (LSTM). The first modification I made was implementing **weight tying**, which equalizes the weights between the linear and embedding layers.

Next, I implemented **variational dropout**, inspired by the paper [1]. During training, a Bernoulli mask is sampled for each batch and kept constant across the sequence positions. This mask is then expanded to match the input size. At the end the input is multiplied by the expanded mask and normalized to maintain the correct scale.

In my final modification, I implemented the non-monotonically activated **AvSGD**. For this, I proposed two separate implementations.

In the first implementation inspired by [1], I used PyTorch's ASGD module. Initially, I applied the traditional SGD algorithm. The transition to ASGD was based on a specific condition: if the current perplexity is higher than the minimum perplexity observed in the recent epochs (excluding the last window size), the switch to ASGD is triggered.

In the second implementation, instead, I manually computed the moving average of the weights while maintaining the SGD optimizer. Thus when the PPL-based condition (as described above) was satisfied, I began to manually compute the moving average of the model weights and continued to apply these averaged weights to the SGD optimizer. In both implementations, during the evaluation phase, I applied the averaged weights to the model and then reset the weights to those derived from training.

Throughout this second part of the assignment, I included a variable learning rate, reducing its value by 60% whenever a worsening of the perplexity is detected.

3. Results

Perplexity was used as the measurement metric. The results below are based on the evaluation of the test set.

For the **first part** of the assignment, the results are reported in Table 1. The results demonstrate that each change led to an improvement in performance.

For the **second part** the results are shown in table 2. as you can see the weight tying combined with the variational dropout brought an excellent improvement in performance while the AvSGD did not modify the result, this is probably because the model is already stable therefore this technique It doesn't bring such obvious benefits.

I have also added two graphs showing the loss for the model 'model_13' (which incorporates all the optimizations from the first part) and the model 'model_23_manual' (which incorporates all the optimizations from the second part). These graphs can be seen in Figure 1 and Figure 2 respectively.

n.	Model name	Description	PPL
0	model_11	LSTM	131
1	model_12	Dropout	125
2	model_13	AdamW	106

Table 1: *Results for the first part of the assignment. The first row shows the implementation of LSTM instead of RNN, the second row shows the results after adding dropout layers after the embedding and the linear layer, and the third row shows the results after switching from SGD to AdamW. The results are cumulative, with each row building upon the previous optimizations.*

n.	Model Name	Description	PPL
0	model_21	Weight Tying	113
1	model_22	Variational Dropout	90
2	model_23_manual	Manual AvSGD	90
3	model_23_auto	Automatic AvSGD	90

Table 2: *Results for the second part of the assignment. The first row shows the implementation of weight tying, the second row shows the results after adding variational dropout, the third row shows the results after applying manual AvSGD, and the fourth row shows the results after applying automatic AvSGD. The results are cumulative, with each row building upon the previous optimizations.*

4. References

[1] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and Optimizing LSTM Language Models," *arXiv preprint arXiv:1708.02182*, 2017.

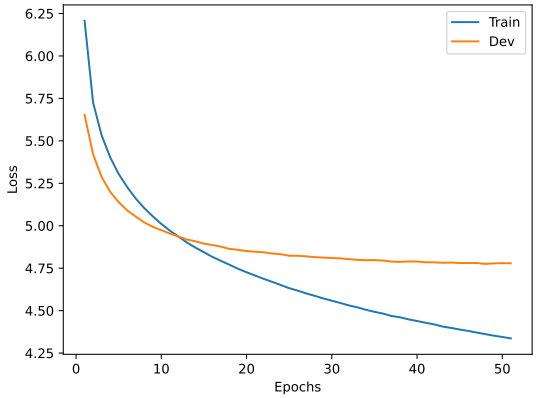

Figure 1: *Loss Training vs Loss Validation part 1 with all three optimization(LSTM, Dropout and AdamW)*

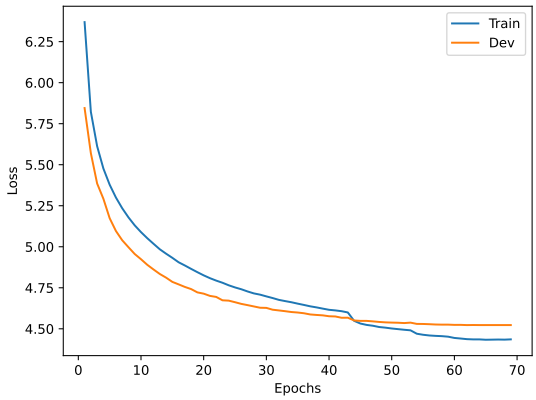

Figure 2: *Loss Training vs Loss Validation part 2 with all three optimization (Weight tying, Variational Dropout and manual AvSGD)*