NLU course project Part 1 - Language Modeling

Alessia Ianes (256181)

University of Trento

alessia.ianes@studenti.unitn.it

1. Introduction

Language modeling is a fundamental task in Natural Language Processing (NLP), aiming to predict the next word in a sequence given its context. This part of the project focuses on improving a baseline Recurrent Neural Network **(RNN)**-based language model by incrementally incorporating different advanced techniques: Long Short-Term Memory **(LSTM)** networks, **dropout regularization**, **AdamW** optimizer, **weight tying**, **variational dropout**, and **non-monotonically triggered AvSGD**. These modifications aim to reduce perplexity (PPL), a key metric for evaluating language models, ensuring **PPL** $<$ **250** across all experiments. By applying these changes and evaluating their impact, this study identifies the most effective strategies for enhancing language model performance.

2. Implementation details

To improve the baseline RNN architecture for language modeling, I implemented a series of architectural and optimization changes starting with the replacement of the network with an LSTM followed by the addition of two dropout layers, one after the embedding layer and one before the last linear layer. After that, I replaced the SGD optimizer with Adam. Next, I incrementally applied to the LSTM some advanced regularization techniques, systematically reducing validation PPL below the target threshold of 250. Firstly, I implemented weight tying by sharing weights between the embedding and output layers, then I used variational dropout by applying the same dropout mask to recurrent connection. Finally I switched the SGD optimizer with the non-monotonically triggered AvSGD (NT-AvSGD).

3. Results

The initial vanilla RNN model was replaced by an LSTM network [1] to address the well-documented vanishing gradient problem inherent in standard RNNs [2], thereby enhancing the network's capacity to capture long-range dependencies within the data. An initial phase of hyperparameter tuning was conducted by evaluating performance across distinct learning rate and batch size values (respectively ranging from 0.01 to 4 and from 32 to 128) (Figure 1). This optimization process resulted in a reduction in the test PPL compared to the baseline model, **from 155.87 to 136.83**, obtained using a learning rate of 1 and a batch size of 128.

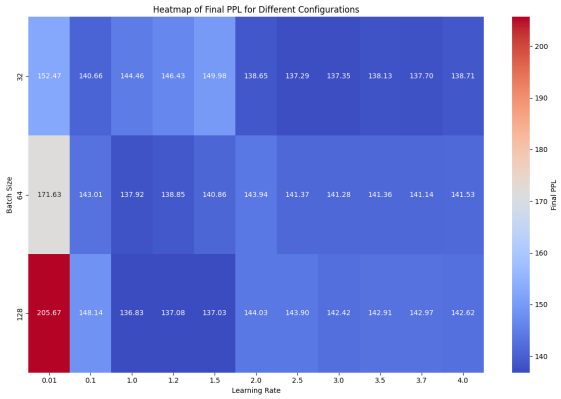

Figure 1: *Heatmap of test PPL in LSTM network*

Building upon this optimized LSTM architecture, the network was further refined to mitigate overfitting and improve generalization adding two dropout layers [3]: one positioned subsequent to the embedding layer and another prior to the final linear output layer. Another phase of hyperparameter exploration was undertaken, investigating again different learning rate values (from 0.01 to 1.2) alongside the previously evaluated batch sizes (32, 64, 128) (Figure 2). This tuning yielded a new best test PPL of **122.15**, achieved with a learning rate of 0.1 and a batch size of 32.

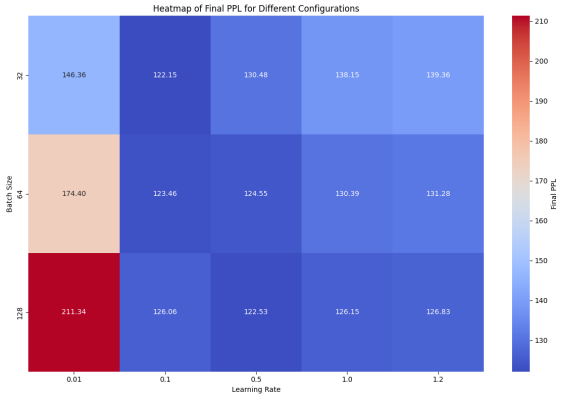

Figure 2: *Heatmap of test PPL in LSTM network with dropout layers*

Finally, the optimization algorithm was switched from Stochastic Gradient Descent (SGD) to AdamW [4] to improve convergence dynamics. With AdamW employed, a focused investigation into lower learning rate values (ranging from 0.0001 to 0.01) was performed (Figure 3), resulting in the overall best test PPL of **120.57**, observed at the lowest tested learning rate value of 0.0001.

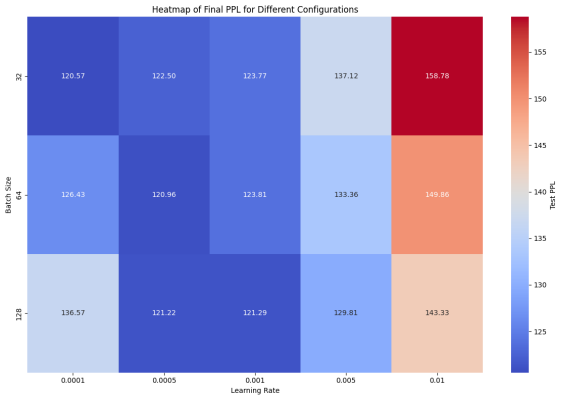

Figure 3: *Heatmap of test PPL in LSTM network with dropout layers and ADAMW optimizer*

As depicted in Figure 4, the validation PPL reveals a sharp initial decline, stabilizing by epoch 50. This trajectory aligns with the optimized hyperparameters identified in this phase. However, analysis of the training and validation loss curves in Figure 5 shows a persistent gap between the two metrics, indicative of overfitting.

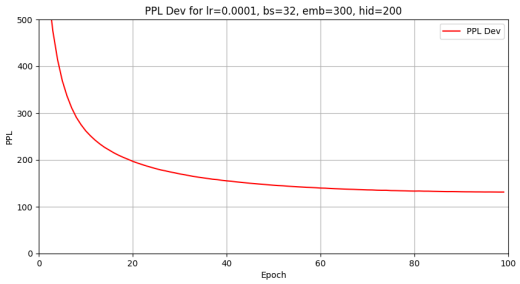

Figure 4: *Plot of validation PPL in LSTM network with dropout layers and ADAMW optimizer*

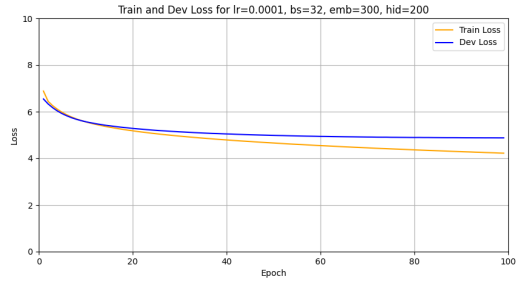

Figure 5: *Plot of loss function for training and validation set in LSTM network with dropout layers and ADAMW optimizer*

Therefore, to further enhance performance and generalization, a series of advanced regularization techniques were incrementally applied to the refined LSTM network (using SGD). First, the weights of the input embedding matrix and the output projection layer were tied [5]. Hyperparameter tuning was conducted by evaluating some learning rate values (from 0.01 to 3) and the previous batch sizes, yielding a new best test PPL of **114.80**, achieved with a learning rate of 3 and a batch size of 32.

Next, variational dropout [5] was added to the LSTM, employing the same dropout mask across all time steps for a given sequence. The tuning process involved evaluating learning rate values from 0.5 to 4, the standard batch sizes, and specific dropout rates for both the embedding layer (0.1 and 0.15) and the output layer (0.35 and 0.4). The optimal configuration in this phase resulted in a test PPL of **95.69**, achieved with a learning rate of 3.7, a batch size of 32, an embedding dropout rate of 0.15, and an output dropout rate of 0.4.

As the final step in this optimization sequence, the SGD optimizer was replaced by NT-ASGD (Non-monotonically Triggered Averaged Stochastic Gradient Descent), a variant of averaged SGD where the averaging trigger is determined by a non-monotonic condition rather than a user-defined schedule [5]. This implementation involved setting a trigger window size and increasing the patience for early stopping from 3 to 7 epochs. The hyperparameter search space focused on a narrower range of learning rate values (from 3.4 to 4) and a reduced set of batch sizes (32, 64). The embedding dropout rate was fixed at 0.15, while the two previously tested output dropout values were retained. As illustrated in Figure 6, in this final phase the development PPL declined sharply, stabilizing by epoch 20. Notably, the activation of NT-ASGD at epoch 23 coincides with a slight plateau in PPL, reflecting stabilization of generalization performance.

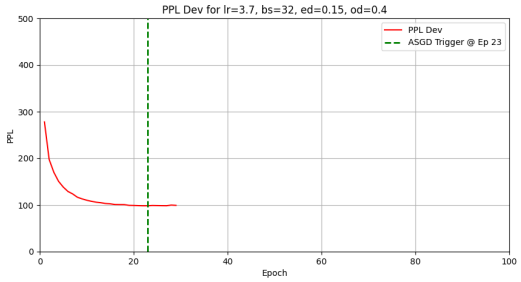

Figure 6: *Plot of validation PPL in LSTM network using weight tying, variational dropout and NT-AvSGD as optimizer*

Analysis of the corresponding training and validation loss curves in Figure 7 shows a reduced gap between the two metrics compared to earlier phases. While minor divergence persists post-epoch 20, the convergence of both losses after NT-ASGD activation indicates effective mitigation of overfitting.

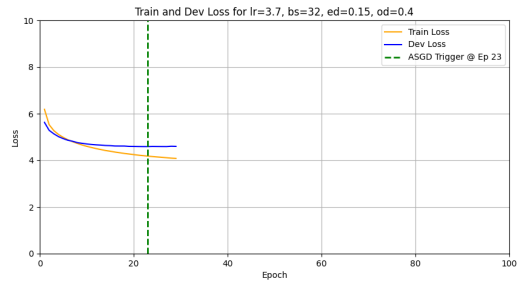

Figure 7: *Plot of loss function for training and validation set in LSTM network using weight tying, variational dropout and NT-AvSGD as optimizer*

This final tuning yielded the overall best test PPL of **91.49** achieved with the same hyperparameter configuration that produced the best performance in the preceding step.

	Technique	**Best Test PPL**
	Baseline RNN	155.87
	LSTM	136.83
+	Dropout layers	122.15
+	AdamW as optimizer	120.57

Table 1: *Summary of the lowest PPL scores obtained for different network configurations during hyperparameter optimization in part 1A.*

	Technique	**Best Test PPL**
	Baseline LSTM	136.83
+	Weight tying	114.80
+	Variational dropout	95.69
+	NT-AvSGD as optimizer	91.49

Table 2: *Summary of the lowest PPL scores obtained for different network configurations during hyperparameter optimization in part 1B.*

4. References

[1] R. DiPietro and G. D. Hager, "Deep learning: Rnns and lstm," in *Handbook of medical image computing and computer assisted intervention*. Elsevier, 2020, pp. 503–519.

[2] Y. Hu, A. Huber, J. Anumula, and S.-C. Liu, "Overcoming the vanishing gradient problem in plain recurrent networks," *arXiv preprint arXiv:1801.06105*, 2018.

[3] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[4] S. M. Zaman, M. M. Hasan, R. I. Sakline, D. Das, and M. A. Alam, "A comparative analysis of optimizers in recurrent neural networks for text classification," in *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2021, pp. 1–6.

[5] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models," *arXiv preprint arXiv:1708.02182*, 2017.