Language Modelling - Lab 4

Marco Prosperi (257857)

University of Trento

marco.prosperi@studenti.unitn.it

1. Introduction

The first part of the project aimed to improve the baseline RNN performance by incrementally adding features and dynamically adjusting hyperparameters (learning rate, hidden units, embedding size). The model was trained on the Penn Treebank dataset, using perplexity as the main evaluation metric. In the second part, advanced techniques [1] were incorporated to further enhance the language model, including Weight Tying to reduce parameters, Variational Dropout (excluding DropConnect) for regularization, and Averaged Stochastic Gradient Descent (AvSGD) for better optimization. These improvements aimed to enhance generalization and convergence while maintaining low perplexity (less than 250).

2. Implementation details

Part A: Incremental Architecture Enhancement. Building upon the PyTorch implementation from the Lab Repository [2], I systematically enhanced the baseline RNN model through architectural modifications and hyperparameter optimization. Starting with the RNN baseline that yielded high perplexity (173.22), I transitioned to an LSTM architecture, achieving significant improvement (137.31). Regularization through dropout layers further enhanced performance (123.14), while adopting the AdamW optimizer with weight decay produced the best Part A results (105.46). These incremental improvements demonstrated consistent performance gains and established the foundation for advanced techniques.

Part B: Advanced Optimization Techniques. I implemented three sophisticated methods from recent literature [1]. Weight Tying reduced parameter count while improving generalization by sharing embedding and output layer weights. Variational Dropout [3] applied consistent dropout masks across temporal sequences, enhancing training stability for recurrent architectures compared to standard dropout.

AvSGD Implementation Strategy. Non-monotonically triggered AvSGD switches from SGD when performance plateaus, averaging parameter updates for improved convergence. Following the authors' recommendations, I implemented a patience-based trigger mechanism: when validation metrics stagnate over multiple epochs, AvSGD activation occurs with patience reset for early termination. Alternative triggering uses non-monotonic windows inspired by the Salesforce Research repository [4], monitoring validation loss deviations from observed minima.

3. Results

The experimental results are summarized in Table 1. Part A demonstrated systematic performance improvements through architectural enhancements, progressing from RNN baseline (173.22) to LSTM with dropout and AdamW optimization (105.46). Figure 1 illustrates the validation perplexity convergence for the best Part A configuration.

Part B achieved further optimization through advanced techniques. The final model with Weight Tying, Variational Dropout, and AvSGD reached 89.47 perplexity using learning rate 2 and 400 hidden/embedding units. Notably, StepLR scheduling (step size 5, gamma 0.75) proved crucial for convergence stability.

AvSGD Optimization Dynamics. An important observation during AvSGD triggering was temporary training loss increase, stemming from optimization dynamics shifts. When switching from SGD to averaged parameter updates, the model experiences trajectory disruption before stabilizing. The StepLR scheduler compounds this effect through conservative updates, requiring additional convergence steps. This temporary increase reflects transition toward flatter, more generalizable loss landscape minima, as shown in Figure 2.

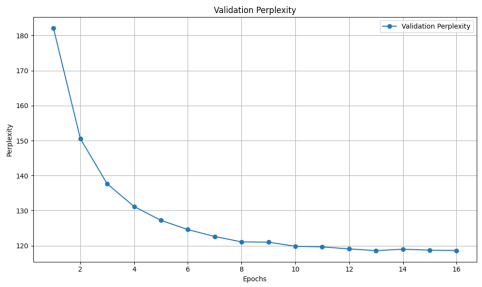

Figure 1: *Validation perplexity of the LSTM model with dropout layers and AdamW.*

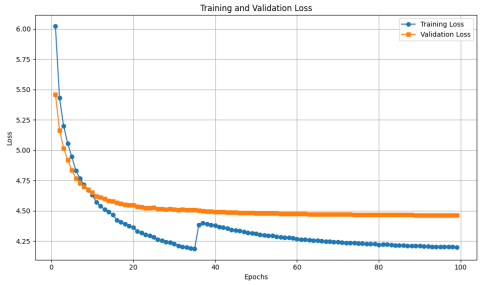

Figure 2: *Trigger point of AvSGD*

4. References

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

[2] UniTrento, "Natural language understanding labs," 2025. [Online]. Available: https://github.com/BrownFortress/NLU-2025-Labs/tree/main

Model	PPL	LR	Hidden	Emb
Part A				
RNN	173.22	0.1	100	100
LSTM	137.31	2	300	300
LSTM + Dropout Layers	123.14	2	300	300
LSTM + Dropout Layers + AdamW	105.46	0.001	250	300
Part B				
LSTM + Weight Tying	95.11	0.001	400	400
LSTM + Weight Tying + Var Dropout	96.68	0.001	400	400
LSTM + Weight Tying + Var Dropout + AvSDG	89.47	2	400	400

Table 1: *Perplexity and hyperparameters of the best models.*

[3] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," 2016. [Online]. Available: https://arxiv.org/abs/1512.05287

[4] SalesforceResearch, "Salesforce lstm reasearch repository," 2018. [Online]. Available: https://github.com/salesforce/awd-lstm-lm