NLU course project - Assignment 1 - LM

Daniele Della Pietra (258646)

University of Trento

daniele.dellapietra@studenti.unitn.it

1. Introduction

This work develops and regularizes an LSTM language model on Penn Treebank (PTB), reporting test perplexity (PPL) for each incremental change. In Part 1.A I start from a simple 1-layer LSTM and apply two modifications in sequence: dropout after the embedding and before the output linear layer, and switching the optimizer from SGD to AdamW. In Part 1.B I introduce three further techniques over the LSTM: weight tying between embedding and decoder, variational (locked) dropout, and non-monotonically triggered Averaged SGD (NT-ASGD)[1]. Models are trained with early stopping on dev PPL. All configurations satisfy the requirement PPL < 250, and each block shows a clear stepwise improvement.

2. Implementation details

2.1. Data, batching and model

I use the Penn Treebank (PTB) train, validation, and test splits. Each line is tokenized on whitespace and terminated with the special token <eos>. The vocabulary is built only on the training corpus and includes <pad> and <eos>; no <unk> token is used (PTB has no OOV under this setup).

The architecture (LM_model) is a single-layer LSTM language model composed of an embedding layer, the recurrent layer, and a linear decoder to the vocabulary. For Part 1.A I enable dropout after the embeddings and before the decoder. For Part 1.B I add: *weight tying* by setting decoder weights equal to the embedding matrix (which enforces emb_dim $=$ hidden_dim); *variational (locked) dropout*, implemented by sampling a mask of shape $[B, 1, H]$ once per sequence and reusing it across time steps; and *NT-ASGD*, which evaluates using averaged weights when validation stops improving.

2.2. Training loop and optimization

Input-to-hidden matrices use Xavier-uniform initialization, hidden-to-hidden matrices are orthogonal, and biases are zero. Gradients are clipped with ℓ_2 norm 5.

Part 1.A. I train three incremental configurations:

- **LSTM + SGD** (no dropout): learning rate 0.5, hidden/embedding $= 256$.
- **LSTM + SGD + Dropout**: $p = 0.3$ at both dropout points, learning rate 0.5, hidden/embedding $= 512$.
- **LSTM + AdamW + Dropout**: $p = 0.6$, learning rate 10^{-3}, hidden/embedding $= 1024$, and a larger batch for stability.

Part 1.B. I keep a compact model (hidden/embedding $=$ 256), train with SGD (learning rate 1.3), batch size 32, and add regularizers in sequence:

- **LSTM + WT** (no dropout).
- **LSTM + WT + Variational Dropout**: $p = 0.5$ on embeddings and pre-decoder activations.
- **LSTM + WT + Variational Dropout (NT-ASGD)**: when patience expires, switch to ASGD with $t_0 = 0$ and $\lambda = 0$, and evaluate using the averaged weights while restoring raw weights afterwards to continue training.

2.3. Early stopping and logging

Early stopping monitors dev PPL. The best checkpoint is retained and evaluated on the test set. I log epoch-level losses and PPL to WANDB. The forward pass returns logits with shape $[B, V, L]$ to match the loss signature with targets of shape $[B, L]$.

3. Results

Tables 1 and 2 report the best test PPL within each block together with the number of trainable parameters. All required runs satisfy PPL < 250.

3.1. Part 1.A (incremental baseline improvements)

Replacing the vanilla RNN with an LSTM produces a strong baseline (147.01 PPL, 5.7M parameters). Adding two dropout points and increasing width reduces test PPL to 110.96 (12.4M), consistent with dropout mitigating overfitting as capacity grows. Scaling further and switching to AdamW gives the best result at 104.24 (28.8M), likely due to decoupled weight decay and more robust curvature adaptation on the wider model.

With smaller models, AdamW *without* dropout underregularized and plateaued above the dropout-enabled SGD variant. Retaining dropout and increasing width restored stability and accuracy, so the final AdamW setting keeps dropout.

3.2. Part 1.B (regularization suite on a compact model)

On the compact 256-dimensional LSTM, weight tying reduces parameters to 3.1M and slightly improves generalization (118.11 PPL), indicating it acts as a helpful inductive bias rather than a bottleneck. Adding variational dropout produces a large gain (93.85 PPL) by stabilizing the noise process across time. Enabling NT-ASGD provides a further, consistent improvement to 90.77 PPL by averaging iterates when validation stalls.

3.3. Takeaways

Improvements are monotonic within blocks and meet the assignment threshold. On PTB, capacity and regularization interact strongly: larger models benefit from dropout plus AdamW, while compact models benefit most from structural regularization (weight tying) and temporally stable noise (dropout), with NT-ASGD offering a late-stage reduction in variance.

Table 1: *Test perplexities for each experiment of part 1.A*

Experiment	#Params	PPL
LSTM + SGD	5.7 M	147.01
LSTM + SGD + Dropout	12.4 M	110.96
LSTM + AdamW + Dropout	28.8 M	104.24

Table 2: *Test perplexities for each experiment of part 1.B*

Experiment	#Params	PPL
LSTM + WT	3.1 M	118.11
LSTM + WT + Vardrop	3.1 M	93.85
LSTM + WT + Vardrop(NT-ASGD)	3.1 M	90.77

4. References

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

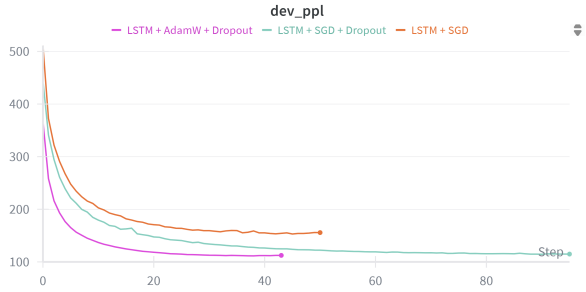

Figure 1: *Validation perplexity on PTB for **LSTM + SGD**, **LSTM + SGD + Dropout**, and **LSTM + AdamW + Dropout**. AdamW+Dropout converges fastest and to the lowest PPL, showing the combined benefit of an adaptive optimizer and regularization.*

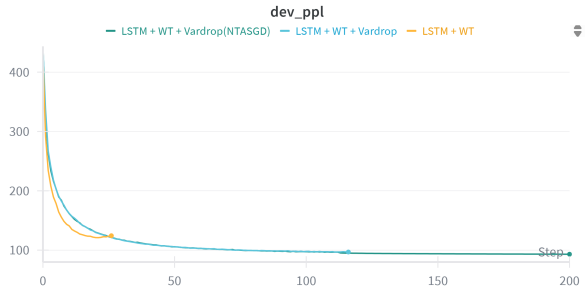

Figure 2: *Validation perplexity for **LSTM + WT**, **LSTM + WT + Variational Dropout**, and **LSTM + WT + Variational Dropout (NT-ASGD)**. Variational (locked) dropout substantially improves generalization over WT; NT-ASGD gives a further late-stage gain.*

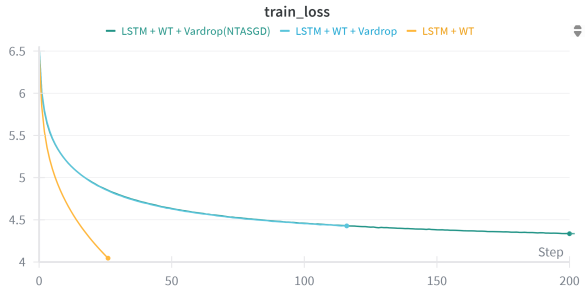

Figure 3: *Training loss for the WT variants. Plain WT attains the lowest training loss but overfits; adding variational dropout (and NT-ASGD) raises training loss yet produces better validation performance (compare to Fig. 2).*