CS288 Project 1: LSTM Improvements

NOPPAPON CHALERMCHOCKCHAROENKIT

In Project 1, the concluding phases entail the construction of an LSTM model for next word prediction with a predefined architecture and hyperparameters, followed by an experimental phase to enhance the model's performance. This report documents a series of experiments conducted with aim to improve the model's validation perplexity.

1. HYPERPARAMETER TUNING

A. Layer Size

Reducing LSTM's output dimension from 512 to 128 through a linear layer led to a performance decline, resulting in a validation perplexity of 172.3. I suspect this drop in performance to be attributed to inherent information loss during dimensionality reductions. In contrast, utilizing a 512-unit linear layer significantly improved the validation perplexity. Subsequently, validation perplexity decreased to 141.6 while retaining the rest of original architecture.

B. Batch Size

I initially began training with a batch size of 64 and subsequently decreased it to 32, with the expectation that this adjustment might enhance generalization. Smaller batch sizes often introduce additional randomness into the training process, which can, in turn, contribute to improved perplexity with our validation dataset. However, this change did not yield a significant impact on the validation perplexity.

2. DROPOUT TO THE ACTIVATION OF THE LSTM OUTPUT

Applying dropout to the LSTM output during training serves as a regularization method, boosting model robustness by allowing it to explore different activation nodes. The motivation is by randomly zeroing out some nodes, dropout helps prevent overfitting and generalize better. This technique, with a 0.5 probability dropout rate on the LSTM output, notably improved validation perplexity. In combination with the linear layer, it reduced the perplexity significantly to 118.1.

3. ENSEMBLING

To enhance the model's performance and robustness, I employed an ensemble technique. The approach used involves combining predictions from two networks initialized with different random seeds and taking their average. While this approach slightly worsened the validation perplexity compared to the previous methods (118.1 to 118.6), there is no major decline in performance, and it could be attributed to random variation.

Table S1. Results

Technique	Validation Perplexity
OG Architecture	172.3
OG + Linear Layer	141.6
OG + Linear Layer + Activation Dropout	118.1
OG + Linear Layer + Activation Dropout + 2 Ensembling	118.6

4. SUMMARY

Two key enhancements that notably enhance the validation perplexity of our LSTM model involve employing a Linear Layer (with the same size as the LSTM's output) and applying dropout to the activation. While ensembling doesn't substantially impact the perplexity on the validation set, it leads to an impressive perplexity score of 87.2 on the test set. The common thread among these techniques is their contribution to enhancing model generalization, thereby improving its robustness across a broader spectrum of datasets.