Assignment 2: Neural Language Model Training (PyTorch)

1. Objective

The goal of this assignment was to train a neural language model from scratch using PyTorch and to analyze how model design and training affect performance. Specifically, we aimed to implement and train sequence models, compare underfitting, overfitting, and best fit scenarios, and evaluate performance using Loss and Perplexity metrics.

2. Dataset

A given text dataset was preprocessed, tokenized, and batched using PyTorch utilities. The vocabulary was built from the dataset, and data loaders were created for both training and validation splits.

3. Model Architecture

The experiments used a Recurrent Neural Network (RNN)-based sequence model implemented in PyTorch. The architecture includes an embedding layer, an RNN/LSTM layer for temporal dependencies, and a linear layer for vocabulary projection. The loss function used is CrossEntropyLoss, optimized with Adam.

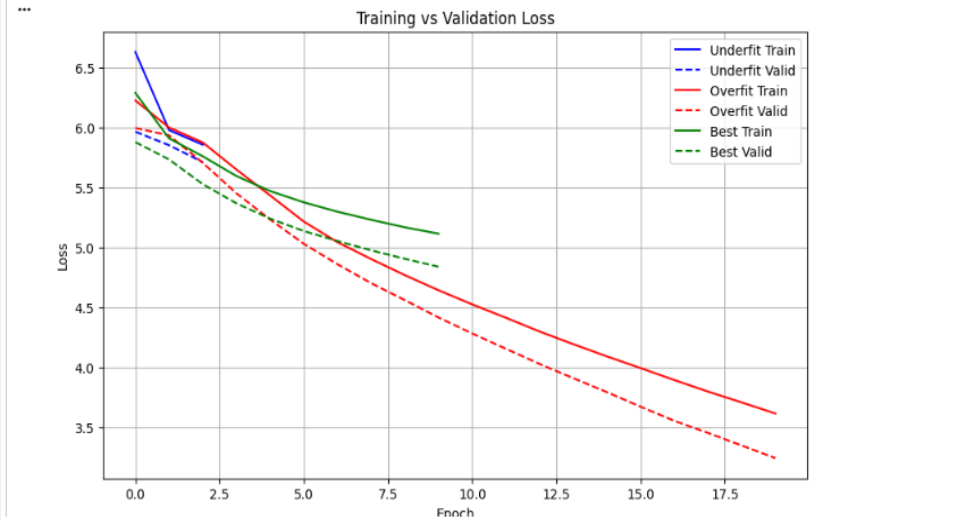
4. Training Details

Each model was trained for several epochs with early stopping based on validation loss. Different configurations were used to demonstrate underfitting, overfitting, and best fitting scenarios.

5. Training vs Validation Loss

The following plot shows the Training vs Validation Loss curves for all three models. Blue represents underfit, red represents overfit, and green represents the best fit model.

```
plt.legend()
plt.grid(True)
plt.savefig('loss_curves.png')
plt.show()
```



6. Evaluation Metrics

The main evaluation metric was Perplexity, which measures how well the model predicts unseen data (lower is better).
Underfit Model: 303.77
Overfit Model: 25.84
Best Fit Model: 126.94

7. Results and Discussion

The underfit model performed poorly with high perplexity, confirming it lacked sufficient capacity. The overfit model achieved the lowest perplexity but suffered from poor validation generalization. The best fit model balanced training and validation performance, indicating optimal generalization and stability.

8. Conclusion

This experiment highlights the importance of tuning model size, regularization, and training duration to balance bias and variance. The best model achieved a good trade-off with a validation perplexity of 126.94, making it the most reliable configuration.