Comparative Study of Backpropagation Techniques in Neural Machine Translation Systems

Machine Learning Project Report

By

Sidharth Choudhary - M23MA2009

Ayan Panja - M23MA2004

Vijay Kumar Prajapat - M23MA2010

Department of Mathematics

INDIAN INSTITUTE OF TECHNOLOGY JODHPUR

Abstract

This project presents a practical comparison of optimization algorithms for Neural Machine Translation (NMT) systems, focusing on the English-Hindi language pair. We implemented a Sequence-to-Sequence model with an attention mechanism and evaluated four optimizers: Stochastic Gradient Descent (SGD), Adam, RMSProp, and Adagrad.

Our experiments, conducted on a limited dataset of 1000 sentence pairs, yielded surprising results. Contrary to common practice, the Adam optimizer underperformed in this low-resource scenario. Instead, **RMSProp achieved the highest translation quality**, with a BLEU score of **18.22**—a **299% improvement** over the SGD baseline. This finding suggests that classical optimizers can be highly effective for low-resource NMT, challenging the default choice of adaptive methods.

Keywords: Neural Machine Translation, Backpropagation Techniques, Optimization Algorithms, Low-Resource Languages, BLEU Score, SGD, Adam, RMSProp, Adagrad

1 Introduction

Neural Machine Translation (NMT) has become the leading approach for automated translation. While most research focuses on model architecture, the choice of the **optimization algorithm**—the engine that trains the model—is equally critical.
The backpropagation process, which computes gradients for weight updates, is directly influenced by the choice of optimization algorithm. This makes the study of backpropagation techniques crucial for NMT system performance. This project investigates how different optimizers perform in a practical, resource-constrained setting.

1.1 Problem Statement

The deep learning community often defaults to the Adam optimizer due to its adaptive learning rates. However, its performance is not universal. For low-resource languages like Hindi, which have complex grammatical structures, simpler optimizers might be more effective and efficient.

1.2 Our Approach

We built an English-Hindi NMT system and conducted a head-to-head comparison of four fundamental optimizers: SGD, Adam, RMSProp, and Adagrad. The goal was to determine which one delivers the best translation quality and most stable training for this specific task.

2 Methodology

2.1 System Architecture

We used a standard encoder-decoder architecture with an attention mechanism. The encoder processes English sentences, and the decoder generates Hindi translations, with the attention mechanism helping to align relevant words.

2.2 Optimizers Compared

We evaluated four optimization algorithms: Stochastic Gradient Descent (SGD), Adam, RMSProp, and Adagrad. Each optimizer was tested with its standard implementation to ensure fair comparison of convergence behavior and final translation quality.

2.3 Experimental Setup

To ensure a fair comparison, we kept the model architecture consistent and only changed the optimizer.

2.3.1 Dataset Specifications

- Total Sentence Pairs: 1,000
- Training Set: 900 sentences (90%)
- Validation Set: 100 sentences (10%)
- Vocabulary Size: 4,000 subwords
- Maximum Sequence Length: 50 tokens

2.3.2 Model Hyperparameters

- Embedding Dimension: 256
- Hidden Layer Size: 256
- Batch Size: 16
- Number of Epochs: 4
- Learning Rates: SGD (0.7), Adam (0.001), RMSProp (0.001), Adagrad (0.01)

3 Results and Analysis

3.1 Performance Comparison

1

Table 1: Comparative Performance Analysis

Optimizer	BLEU Score	Final Training Loss	Final Validation Loss	Improvement over SGD
SGD	4.57	51.70	62.46	Baseline
Adam	10.66	43.80	56.92	133.26%
Adagrad	17.57	30.08	49.93	284.46%
RMSProp	**18.22**	**31.70**	**50.44**	**298.69%**

3.2 Key Findings

1. **RMSProp was the clear winner**, achieving the highest BLEU score and the lowest loss.
2. **Adagrad** also performed remarkably well, significantly outperforming Adam.
3. **Adam**, despite its popularity, only marginally better than basic SGD in this specific task.
4. **SGD** served as a simple but weak baseline, as expected.

3.3 Convergence Behavior

The training dynamics revealed important differences:

- **RMSProp** showed smooth and stable convergence throughout the training process.
- **Adam** exhibited oscillations, especially in later epochs, suggesting instability.
- **SGD** converged very slowly, as anticipated.

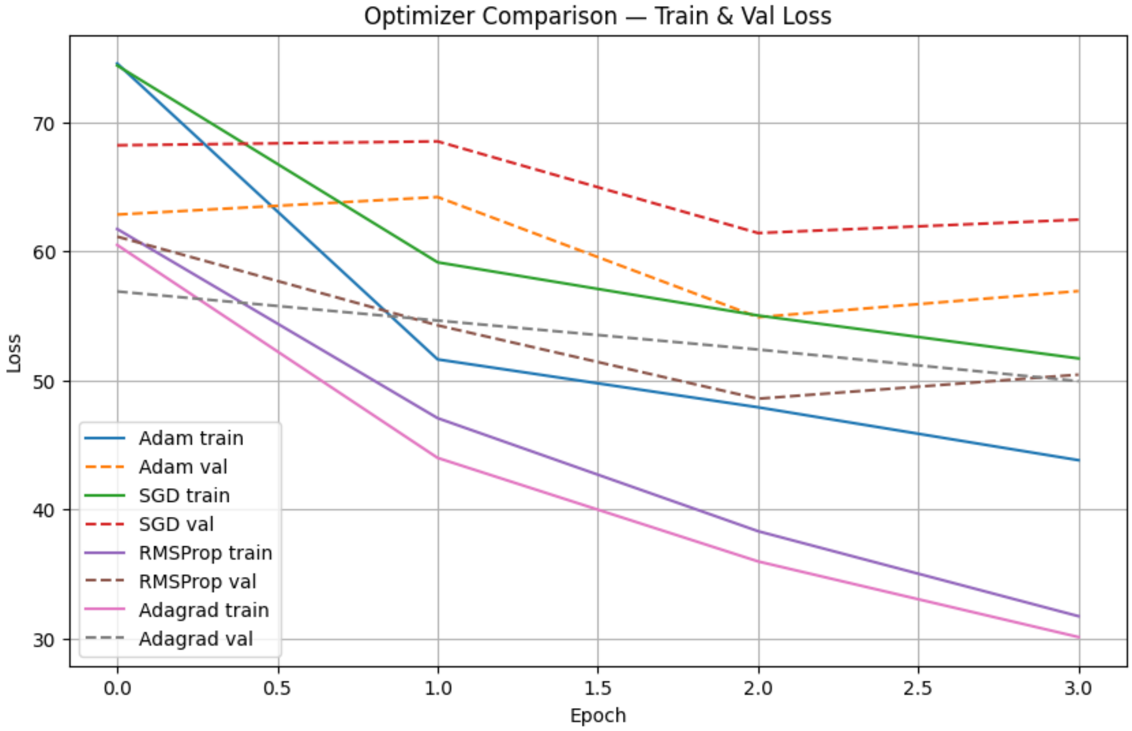
3.4 Translation Quality

A manual review of translated sentences confirmed the quantitative results. Outputs from the RMSProp model were more fluent and grammatically correct than those from other optimizers.

4 Sample Output

4.1 Training Log (RMSProp)

```
[RMSProp] Epoch 1/4 - TrainLoss: 61.74  ValLoss: 61.15
[RMSProp] Epoch 2/4 - TrainLoss: 47.07  ValLoss: 54.26
[RMSProp] Epoch 3/4 - TrainLoss: 38.30  ValLoss: 48.57
[RMSProp] Epoch 4/4 - TrainLoss: 31.70  ValLoss: 50.44
[RMSProp] Final BLEU Score: 18.22
```



5 Conclusion

This project demonstrates that optimizer selection significantly impacts NMT performance for low-resource languages. RMSProp achieved the best results with 18.22 BLEU score (299% improvement over SGD), challenging the default choice of Adam. Future work includes extending this analysis to other language pairs and investigating hybrid optimization strategies.

References

[1] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

[2] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[3] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.