Performance Comparison of Adam, RMSprop, and AdamW Optimizers on an ANN using KMNIST Dataset

Prepared By: Group 7

- Hozefa Patel [N01686385]
- Jenil Pancholi [N01665133]
- Dev Ariwala [N01664568]

Course – Advanced Deep Learning

Course Code – 5500

Section Code – ONA

Submitted to – Prof. Hossein Pourmodheji

Table of Contents

Introduction

Deep learning (DL) has revolutionized various fields, enabling significant advancements in image classification, natural language processing, and more. This project aims to compare the performance of three popular optimization algorithms—Adam, RMSprop, and AdamW—on an Artificial Neural Network (ANN) using the Kuzushiji-MNIST (KMNIST) dataset. We aim to identify the most effective optimizer for this specific classification task by analyzing training, validation, and test results.

Dataset Description

The Kuzushiji-MNIST (KMNIST) dataset is a drop-in replacement for the MNIST dataset but contains images of cursive Japanese (Kuzushiji) characters. The dataset consists of 70,000 grayscale images, each of size 28x28 pixels, divided into 60,000 training images and 10,000 test images. Each image belongs to one of ten classes representing different Kuzushiji characters.

Deep Learning Model

The ANN used in this project is a multi-layer perceptron with the following architecture:

- **Input layer:** 28x28 neurons (flattened)
- **Hidden layers:**
    - First layer: 512 neurons with ReLU activation, followed by Batch Normalization and Dropout
    - Second layer: 256 neurons with ReLU activation, followed by Batch Normalization and Dropout
    - Third layer: 128 neurons with ReLU activation, followed by Batch Normalization and Dropout
    - Fourth layer: 64 neurons with ReLU activation, followed by Batch Normalization
- **Output layer:** 10 neurons with Softmax activation

Optimizer Algorithms

Adam (Adaptive Moment Estimation)

- Combines the advantages of two other extensions of stochastic gradient descent, namely, AdaGrad and RMSprop.
- Computes adaptive learning rates for each parameter.
- Parameters: **learning_rate = 0.001, beta1 = 0.9, beta2 = 0.999, epsilon = 1e-8.**

RMSprop (Root Mean Square Propagation)

- Adapts the learning rate for each parameter by dividing the learning rate by an exponentially decaying average of squared gradients.
- Parameters: **learning_rate = 0.001, alpha = 0.99, epsilon = 1e-8.**

AdamW (Adam with Weight Decay)

- Variant of Adam that decouples the weight decay from the gradient update.
- Parameters: **learning_rate = 0.001, beta1 = 0.9, beta2 = 0.999, epsilon = 1e-8, weight_decay = 0.01.**

Solutions, Findings, and Results

Training, Validation, and Testing

The model was trained and evaluated using the specified optimizers with a fixed learning rate of 0.001. The performance metrics were recorded for each optimizer.

Results Summary

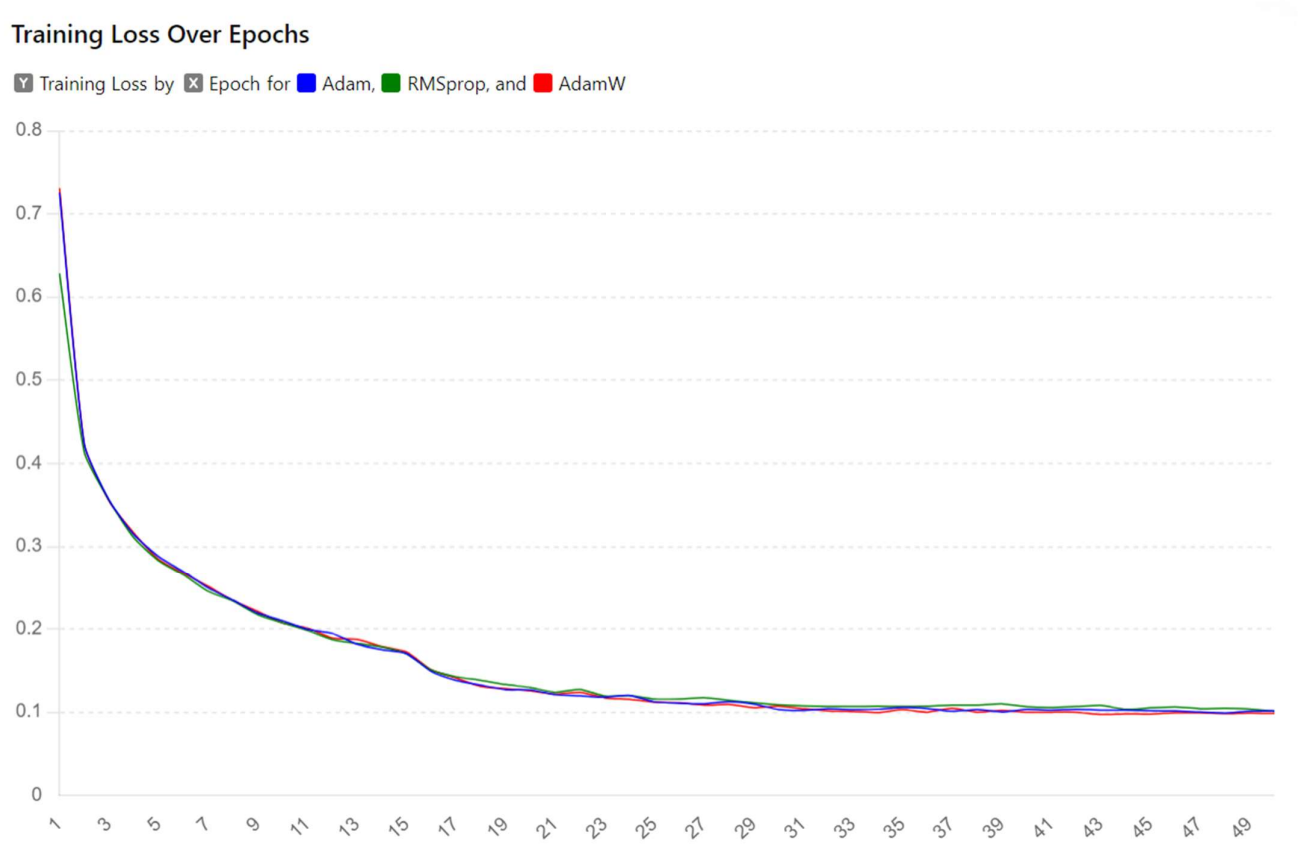
Optimizer	Learning Rate	Training Accuracy	Test Accuracy
Adam	0.001	96.85%	91.86%
RMSprop	0.001	96.93%	91.68%
AdamW	0.001	96.99%	92.02%

Best Result

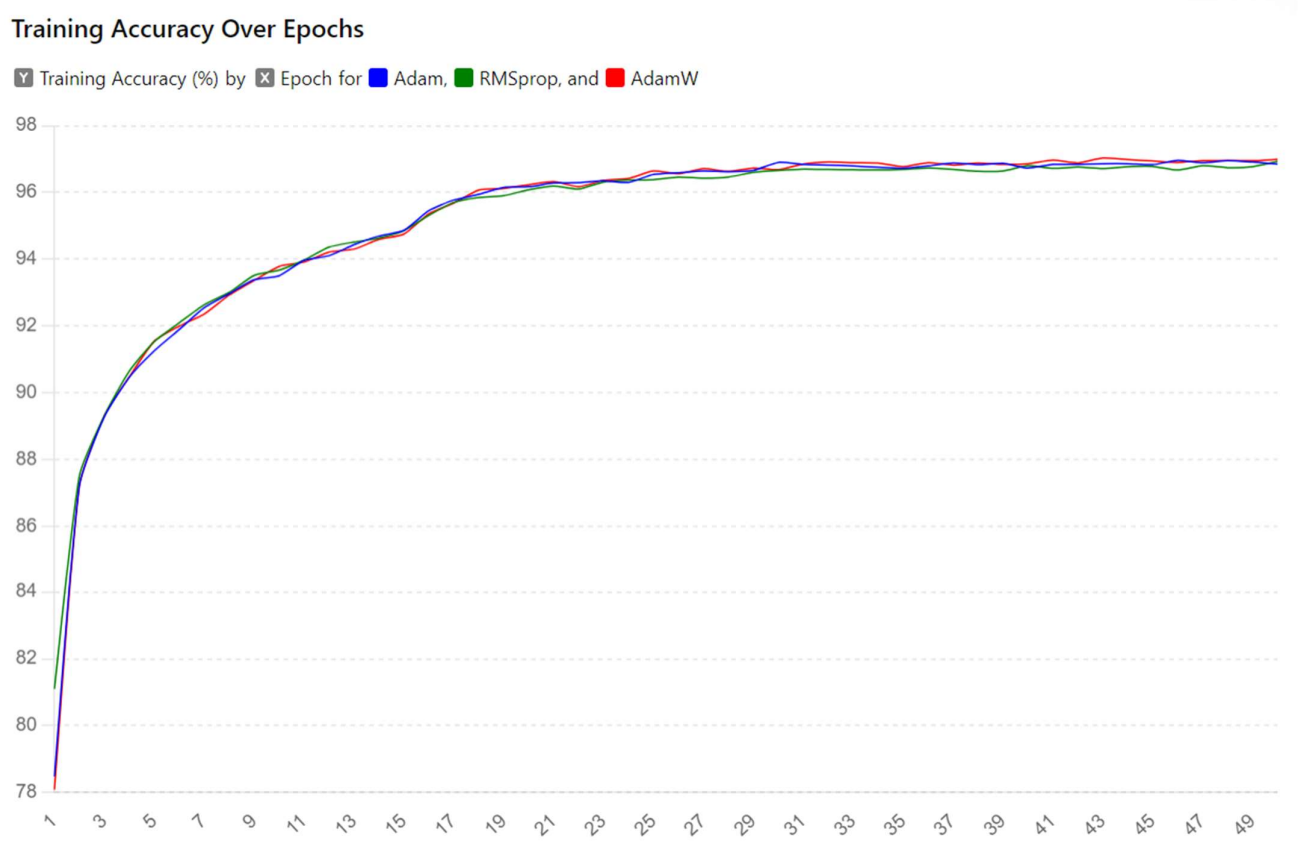
- **Best Test Accuracy:** 92.02%
- **Best Optimizer:** AdamW
- **Parameters:** learning_rate = 0.001

Visual Results

Training Loss Over Epochs



Training Accuracy Over Epochs



Interpretation, Discussion, and Conclusion

The findings suggest that AdamW, with its decoupled weight decay mechanism, provides a slight performance edge over Adam and RMSprop in the context of the KMNIST dataset. Adam and RMSprop also perform well but exhibit slightly lower accuracies compared to AdamW.

The choice of optimizer can significantly impact the training dynamics and final accuracy of a deep learning model. AdamW's effectiveness in handling weight decay separately from the gradient update process may contribute to better generalization performance, as evidenced by the higher test accuracy.

Group Member Contributions

- **Hozefa Patel [N01686385]:** Implemented data loading and preprocessing, designed the model architecture, and wrote the training functions.
- **Jenil Pancholi [N01665133]:** Conducted hyperparameter tuning, trained the model with different optimizers, and recorded results.
- **Dev Ariwala [N01664568]:** Performed evaluation, created visualizations, and compiled the final report.

References

- Adam Optimization Algorithm
- RMSprop: Divide the gradient by a running average of its recent magnitude
- AdamW and Super-Convergence is Now the Best Optimizer
- KMNIST Dataset: GitHub Repository