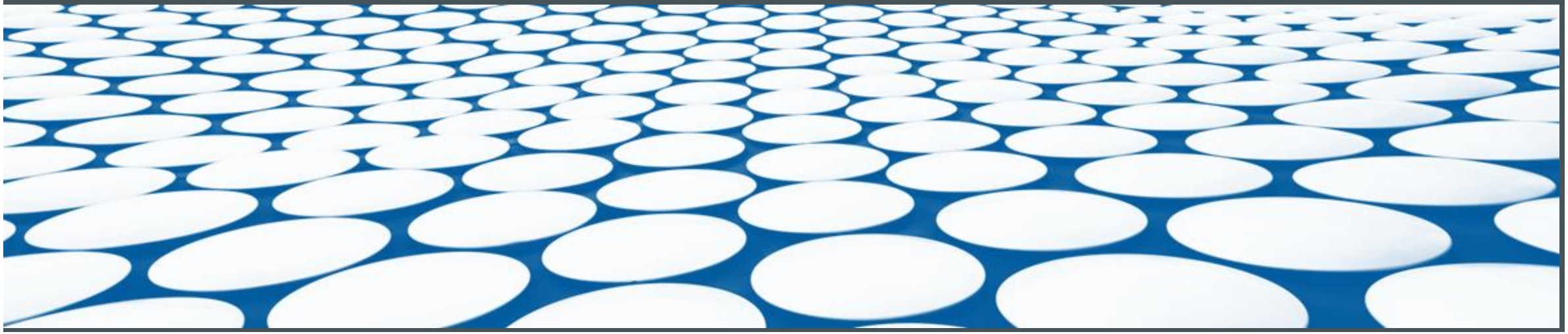
GROUP #4 MIDTERM REPORT

COMPARATIVE ANALYSIS OF DEEP LEARNING OPTIMIZERS ON THE KMNIST DATASET



1. Introduction

Investigate and compare the performance of three popular deep learning optimizers:

- Adam (Adaptive Moment Estimation)
- RMSprop (Root Mean Square Propagation)
- AdamW (Adam with Weight Decay)

2. Dataset Description

The Kuzushiji-MNIST (KMNIST) dataset [5] consists of:

- 70,000 grayscale images of Japanese Hiragana characters
- Standard split: 60,000 training samples and 10,000 test samples
- 28×28 pixel resolution (784 input dimensions when flattened)
- 10 balanced classes representing different characters

Preprocessing included:

- Normalization to center pixel values around zero
- Conversion to PyTorch tensors
- Batch loading via PyTorch DataLoader

3. Model Architecture

The neural network architecture consists of:

- Input Layer: 784 neurons (flattened 28×28 images)
- Hidden Layers:
  - First hidden layer: 128 neurons with ReLU activation
  - Second hidden layer: 64 neurons with ReLU activation
- Output Layer: 10 neurons (one per class) with Softmax activation

4. Methodology

A. Baseline Evaluation

Initial comparisons used fixed hyperparameters:

- Learning Rate (LR): 0.001
- Batch Size (BS): 64
- Epochs: 5
- Loss Function: CrossEntropyLoss

B. Hyperparameter Tuning

Random search was performed over:

- Learning Rates: [0.0005, 0.001, 0.005]
- Batch Sizes: [32, 64, 128]
- Three configurations were randomly sampled and evaluated per optimizer.

C. Model Validation

5-fold cross-validation was conducted using each optimizer's best configuration from tuning. This provided robust performance estimates by:

- Splitting data into 5 folds (80% training, 20% validation)
- Training and evaluating on each fold
- Averaging metrics across folds

5. Results – Baseline Performance

Optimizer	Test Accuracy (%)	Test Loss	Training Time (s)
Adam	77.05	1.6890	66.61
RMSprop	77.66	1.6842	54.83
AdamW	77.39	1.6868	56.73

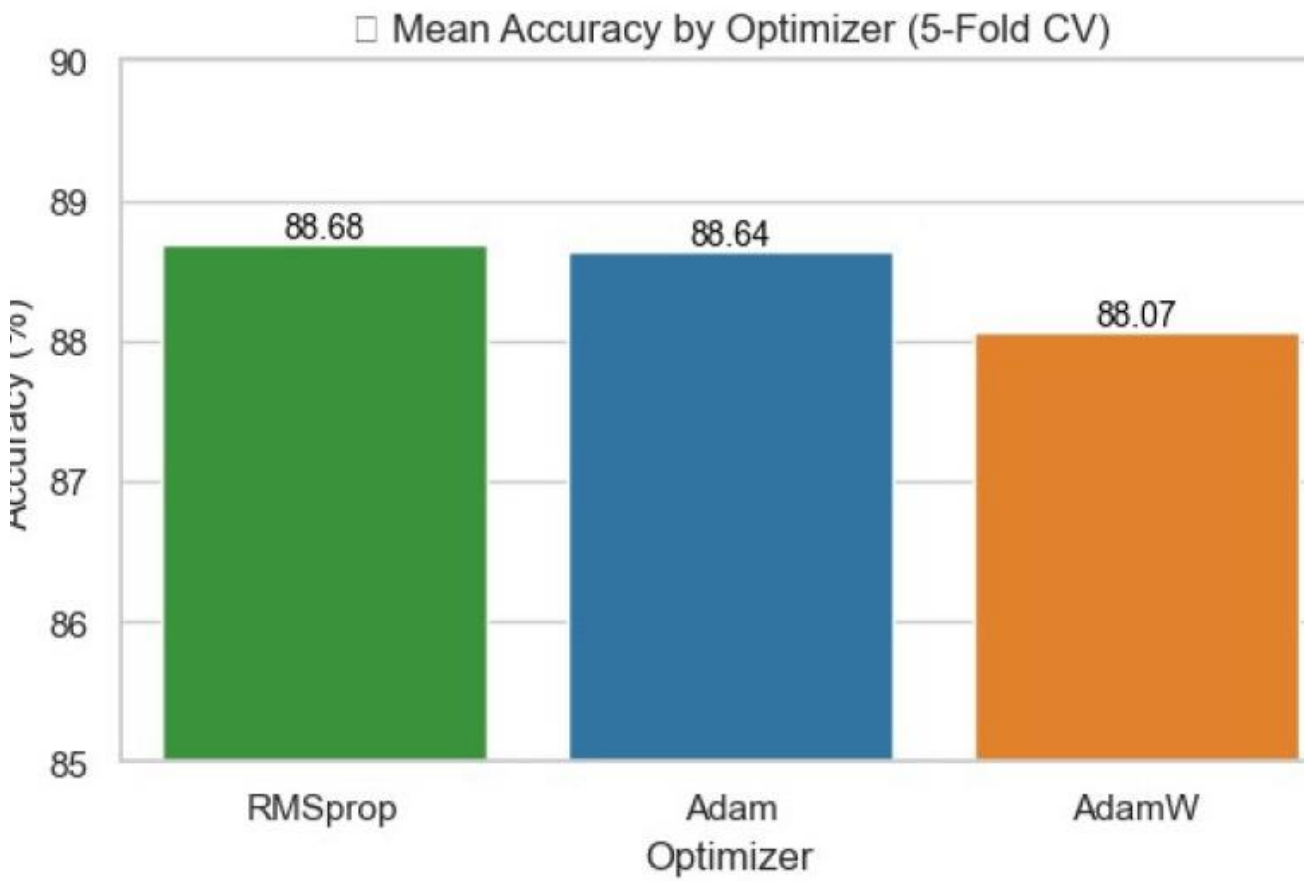
5. Results – After Hyperparameter Tuning

Optimizer	Best LR	Best BS	Accuracy (%)	Loss	Time (s)
Adam	0.0005	32	76.60	1.6962	64.80
RMSprop	0.0005	32	78.01	1.6806	168.32
AdamW	0.0010	128	77.91	1.6835	89.01

5. Results – 5-Fold Cross-Validation

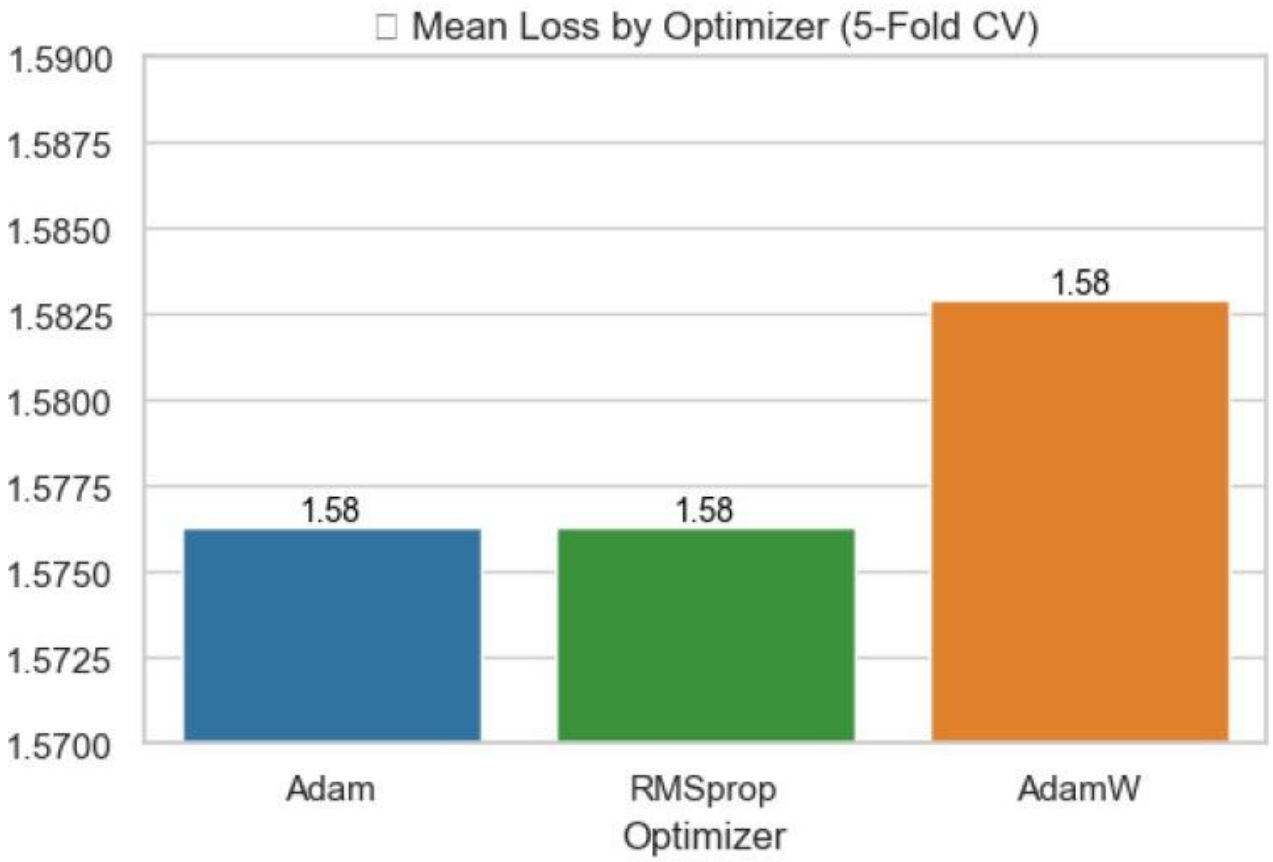
Optimizer	Mean Accuracy (%)	Mean Loss	Mean Time (s)
Adam	88.64	1.5763	70.83
RMSprop	88.68	1.5763	51.87
AdamW	88.07	1.5829	42.30

6. Visualizations - Mean Accuracy Comparison



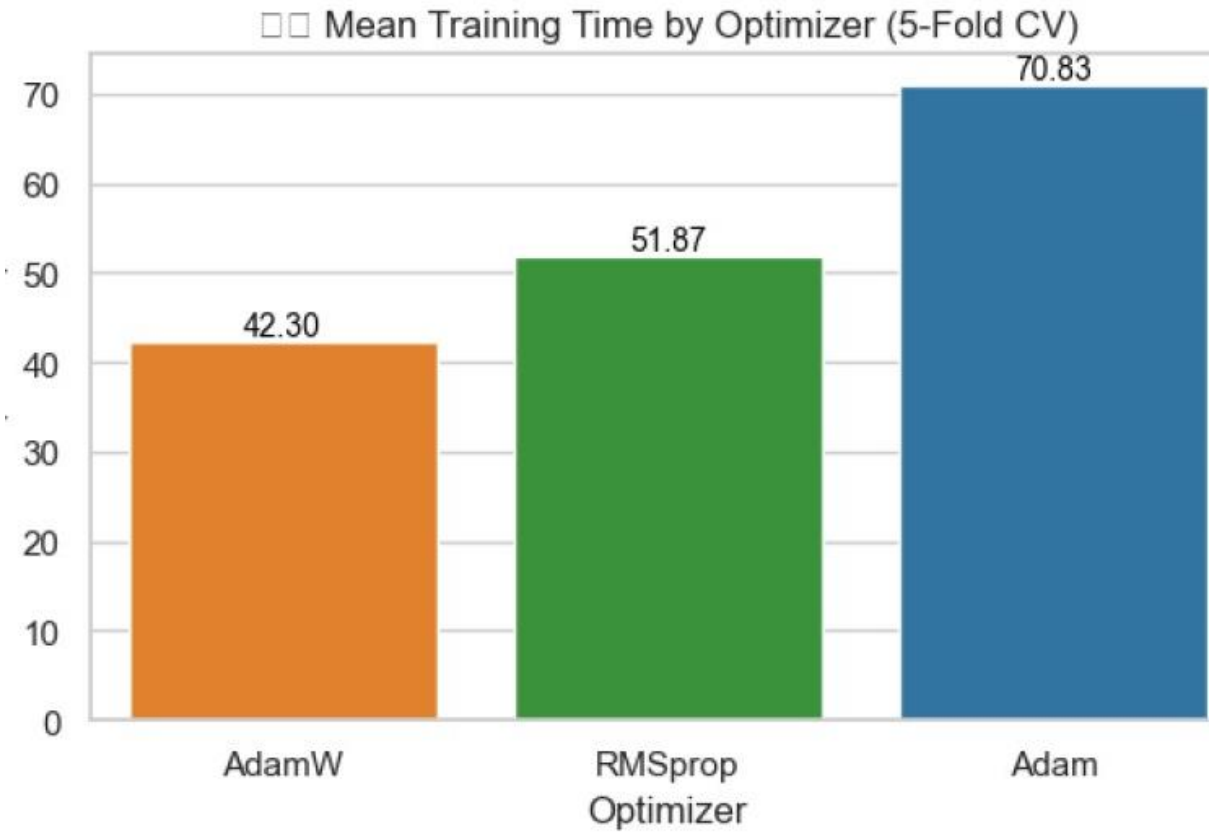
Bar plot showing RMSprop with highest accuracy (88.68%), followed closely by Adam and AdamW

6. Visualizations – Loss Comparisons



Nearly identical loss values across optimizers (~1.576), suggesting similar convergence properties.

6. Visualizations – Training Time Comparison



Clear progression from AdamW (fastest at 42.30s) to RMSprop to Adam (slowest at 70.83s)

7. Interpretation and Discussion

The experimental results reveal several important insights:

- RMSprop Performance: Achieved the highest accuracy in both tuned (78.01%) and cross-validation (88.68%) evaluations. Its slightly better performance may stem from its adaptive learning rate mechanism being particularly suited to the KMNIST dataset characteristics.
- AdamW Efficiency: Demonstrated the fastest training times, being approximately 40% faster than Adam while maintaining competitive accuracy. This makes it attractive for rapid prototyping or resource-constrained applications.
- Consistent Loss Patterns: The nearly identical loss values suggest all optimizers achieve similar final convergence points, though with different training dynamics.
- Hyperparameter Sensitivity: RMSprop showed the largest performance variation during tuning (best accuracy 78.01% vs baseline 77.66%), indicating greater sensitivity to learning rate and batch size configurations.
- Cross-Validation Benefits: The significant accuracy gains during k-fold evaluation (~10% over single test set) highlight the importance of robust validation methods for obtaining reliable performance estimates.

8. Conclusion

Based on comprehensive empirical evaluation:

- RMSprop is recommended as the best overall optimizer for this task, demonstrating:
  - Highest accuracy (88.68% in CV)
  - Best loss values
  - Reasonable training times
- AdamW is ideal when training speed is prioritized, offering:
  - Fastest training (42.30s per fold)
  - Only marginally lower accuracy than RMSprop
- Adam, while competitive, didn't outperform the others in any metric for this particular architecture and dataset.

9. References

[1] Kingma, D.P. & Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv:1412.6980

[2] Hinton, G. (2012). Neural Networks for Machine Learning. Coursera Lecture 6e

[3] Loshchilov, I. & Hutter, F. (2017). Decoupled Weight Decay Regularization. arXiv:1711.05101

[4] Ruder, S. (2016). An Overview of Gradient Descent Optimization Algorithms. arXiv:1609.04747

[5] Clanuwat et al. (2018). Deep Learning for Classical Japanese Literature. arXiv:1812.01718

[6] Goodfellow, I. et al. (2016). Deep Learning. MIT Press

[7] Bergstra, J. & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. JMLR

[8] Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. IJCAI

GROUP 4 MEMBERS

- Jawwad Khalil Ahmed
- Eric Efon
- Daniel Mehta
- Thomas Nash
- Jeffrey Ng