**Model Performance Analysis Report**

**Introduction**

The objective of this study was to evaluate the performance of different neural network architectures on the KMNIST dataset. We experimented with multiple model configurations by varying the number of hidden layers, activation functions, optimizers, and weight decay. The models were assessed based on their validation accuracy and final test accuracy.

**Key Observations**

**Comparison of Model Configurations**

| **Hidden Layers** | **Activation** | **Optimizer** | **Weight Decay** | **Best Validation Accuracy** | **Test Accuracy** |
| --- | --- | --- | --- | --- | --- |
| [128, 64] | ReLU | Adam | 0.0005 | 95.30% | 88.83% |
| [128, 128, 64] | ReLU | SGD | 0.0000 | 84.03% | 71.26% |
| [256, 128, 64] | Sigmoid | RMSprop | 0.0005 | 90.03% | 78.81% |
| [128, 64, 32] | ReLU | Nesterov | 0.0005 | 93.52% | 86.38% |

**Interpretation of Results**

* The best performing model had **128 and 64 hidden units**, used **ReLU activation**, the **Adam optimizer**, and **L2 weight decay of 0.0005**. This model achieved a validation accuracy of **95.30%** and a test accuracy of **88.83%**.
* The model with **SGD optimizer and no weight decay** performed the worst, indicating that proper regularization is crucial for generalization.
* The model with **sigmoid activation** (instead of ReLU) showed a lower performance, suggesting that ReLU is better suited for this dataset.
* **Weight decay (L2 regularization) improved model performance**, especially for the best-performing configuration.

**Confusion Matrix Analysis**

A confusion matrix was generated for the best model. The results indicate that:

* Most misclassifications occurred between visually similar characters.
* The model generally performed well across all classes, but certain classes had slightly higher misclassification rates.

**Future Improvements**

1. **Experimenting with deeper architectures** – Adding more hidden layers may improve feature learning.
2. **Fine-tuning the learning rate** – Optimizers like Adam and RMSprop are sensitive to learning rate adjustments.
3. **Data augmentation** – Adding rotation, noise, and scaling to training images can further improve generalization.
4. **Batch normalization** – Normalizing activations between layers may lead to faster convergence.

**Conclusion**

The study highlights the effectiveness of Adam optimization and ReLU activation in neural networks for the KMNIST dataset. Proper regularization (L2 weight decay) significantly improves model generalization, and future work can focus on fine-tuning hyperparameters and incorporating advanced techniques to further boost accuracy.