**Abstract**

This project compares two deep learning classifiers for the Kuzushiji-MNIST dataset: a Naive Convolutional Neural Network (CNN) based on the CNN from HW4, and a Stochastic Optimization Plain Convolutional Neural Network (SOPCNN) architected in this paper: [arXiv:2001.08856](https://arxiv.org/abs/2001.08856). Both models are designed to classify grayscale images, but differ in terms of their architectures and optimization methods. The comparison focuses on the effectiveness of regularization techniques, hyperparameter choices, and training methodologies, with performance evaluated using confusion matrices and other metrics.

**Introduction**

The Kuzushiji-MNIST dataset [2], comprising 28x28 grayscale images of Japanese characters, is a challenging classification task requiring effective feature extraction. In this project, we employ two models—a CNN and an SOPCNN—to tackle this challenge. The CNN leverages a traditional convolutional approach, while the SOPCNN integrates stochastic optimization and advanced regularization techniques, including dropout and data augmentation, to achieve robust generalization and performance improvements [1]. The reason we use the SOPCNN architecture, is because it is one of the more accurate architectures for the regular MNIST dataset, and incorporates many different methods not seen in Brandon’s naive CNN for HW4.

**Methods**

**Data Preprocessing**

* **CNN Model:** 
  + The CNN used the basic preprocessing steps—converting the images to tensors and normalizing them to have a mean of 0.5 and a standard deviation of 0.5. We want to normalize the data here because without it, the model converges more quickly during training. When different features have different ranges, gradient descent can "bounce" and slow convergence.
* **The SOPCNN:**
  + Data augmentation is a proven technique to enhance the accuracy of image classifiers, particularly in convolutional neural networks. In the SOPCNN model, Yahia Saeed Assiri implements various augmentations to enrich the training dataset. [1]
  + This strategy aligns with earlier pioneering work: LeCun et al. enhanced LeNet5’s performance by applying affine transformations, including translation, scaling, squeezing, and shearing [3]. These methods are mirrored in our implementation of the SOPCNN model through random scale, shear, and shifting.
  + Krizhevsky’s improvements to AlexNet [4] for the 2012 ImageNet competition [5] included augmentations like mirroring and color adjustments; while color adjustments are omitted in the SOPCNN approach, SOPCNN still applies the normalization based on the mean and standard deviation of the KMINST dataset for additional variability.
  + This augmentation pipeline introduces substantial data diversity, enriching your model’s ability to learn robust features across a broader spectrum of image variations.
  + This helped to create a more diverse dataset, which improved the model's generalization ability. It also normalizes it with a calculated mean and standard deviation, to get a more accurate result of the distribution.

**Model Architectures**

* **CNN**: The CNN model uses four convolutional layers with ReLU activation, followed by max-pooling to regularize the network. The architecture aims to strike a balance between model complexity and generalization.
* **SOPCNN**:
  + The SOPCNN model also consists of multiple convolutional layers, but applies additional regularization, inspired by Assiri's work, with a particular emphasis on dropout after every convolutional block and customized early stopping criteria to avoid overfitting. The model uses stochastic optimization techniques to determine the best parameters.
  + dropout is a method first introduced by Hinton et al. [4] This technique has demonstrated effectiveness across diverse neural networks, producing improved results in tasks such as handwritten digit recognition (MNIST), image classification (CIFAR-10), and speech recognition (TIMIT). [6]
  + In dropout, certain hidden unit activations are randomly set to zero with a fixed probability during training. When evaluating the model, all activations are restored, and the output is scaled according to the dropout rate. This process effectively simulates an ensemble of smaller networks, helping to reduce overfitting by discouraging feature detectors from relying too heavily on each other. Dropout thus acts as a form of robust “bagging,” where each mini-network’s unique feature patterns strengthen the overall model’s performance. [5]

**Hyperparameter Selection**

* **CNN**:
  + **Epochs**: Trained for 100 epochs.
  + **Learning Rate**: 0.04.
  + **Batch Size**: 1024, to leverage computational resources effectively.
* **SOPCNN**:
  + **Epochs**: Trained for up to 1000 epochs, with early stopping to determine the optimal point.
  + **Learning Rate**: 0.004, lower than the CNN to facilitate more refined updates during training. And more closely relate to the original paper
  + **Batch Size**: 1024, similar to the CNN.

**Training Methodologies**

* **Stopping Criteria**:
  + Both models used early stopping based on validation loss to avoid overfitting.

**Results**

The results were evaluated using confusion matrices for both models. The SOPCNN outperformed the traditional CNN, as shown below:

* **CNN Model**: Achieved an accuracy of approximately 93.75%, with confusion between similar character classes.
* **SOPCNN Model**: Reached 98.58% accuracy, benefiting from more robust regularization techniques and effective data augmentation.

**Discussion & Conclusion**

* **Strengths**:
  + **CNN**: The simpler architecture led to faster training and lower computational overhead. Suitable for environments with resource constraints.
  + **SOPCNN**: Demonstrated superior generalization due to effective stochastic optimization, dropout, and extensive data augmentation, making it better suited for complex image classification tasks.
* **Weaknesses**:
  + **CNN**: Limited by overfitting and less effective regularization, which reduced its performance on validation data.
  + **SOPCNN**: The additional complexity, in terms of dropout and data augmentation, made the training longer and required more computational resources.
* **Best Stopping Point**: The customized early stopping in the SOPCNN, influenced by Assiri's paper, helped to prevent overfitting while achieving the best possible validation performance.

**References**

1. Assiri, Y. S. (2020). *Stochastic Optimization of Plain Convolutional Neural Networks with Simple Methods.* [arXiv:2001.08856](https://arxiv.org/abs/2001.08856)
2. Kuzushiji-MNIST Dataset: [GitHub Repository](https://github.com/rois-codh/kmnist)
3. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 86(11):2278–2324, 1998.
4. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. *Improving neural networks by preventing co-adaptation of feature detectors.* arXiv preprint arXiv:1207.0580, 2012.
5. DeVries, T. and Taylor, G. (2017). *Improved Regularization of Convolutional Neural Net*works *with Cutout.*
6. *Brownlee J (2019) A Gentle Introduction to Dropout for Regularizing Deep Neural Networks. In: Machine Learning Mastery.* https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks.