EEL6814: Project 1

Joseph Madden
Electrical and Computer Engineering Dept.
University of Florida
Gainesville, United States
josephmadden@ufl.edu

***Abstract*—This paper discusses two deep learning classifiers implemented on the KMNIST dataset. The first is a multilayered perceptron classifier and the second is a convolutional neural network classifier. In addition, this paper extends the convolutional neural network using ensemble learning via bagging.**

***Keywords*—machine learning, deep learning, Multilayered Perceptron, Convolutional Neural Network, image recognition, ensemble learning, bagging**

I. Introduction

Deep learning classifiers are the forefront of image classification technologies. Their ability to generalize to wide range of different problem sets with relatively high accuracy makes them unparalleled [1-3]. In this paper, two deep learning classifiers will be trained on the Kuzushiji-MNIST dataset. The first model will be an MLP classifier, and the second model will be a convolutional neural network.

A. Dataset Provided

The provided source data for this project was the Kuzushiji-MNIST dataset (hereby referred to as KMNIST). The KMNIST dataset, similar to the MNIST dataset, contains 70,000 images, each in 28x28 grayscale. There are ten classes, each corresponding to one hiragana character.

The dataset is split into 60,000 training images and 10,000 test images. The GitHub page provides the dataset in both MNIST format and NumPy format. The NumPy format will be utilized for both models. The dataset is available for review at the following link: https://github.com/rois-codh/kmnist.

II. MLP Classifier

ZAYN's work here

III. Convolutional Neural Network

CNNs are the preferred deep learning technique for image classification. The reason for this is their ability to utilize locality in image data. CNNs perform very well on the original MNIST dataset, so it only makes sense to apply one to the KMNIST dataset. The following sections will detail the design, implementation, and results of using a CNN classifier.

A. Design

As with any machine learning model, understanding the hyperparameters of the model is essential. The following are the standard hyperparameters to be considered for any CNN.

- The kernel size for convolutional layers
- The number of kernels (convolutional layers)
- The length of strides (X)
- The pooling size

In addition, there are the hyperparameters that exist for all neural networks. These hyperparameters are:

- Learning Rate
- Number of epochs / stopping criteria
- Batch size
- Activation function (X)
- The number of hidden layers
- The width of hidden layers
- Weight initialization (X)

Out of consideration for the time and scale of this project, it would not be possible to conduct searches for all these hyperparameters. Therefore, some hyperparameters will be held constant and not considered for experimentation. These hyperparameters will instead be selected based on results from prior academic reading and best practices.

- Momentum

The activation function
The weight initialization

B. Search Methodology

The search for the best hyperparameter values will take place in phases. During a phase, only the described values will be experimented on. All other hyperparameters will be held constant for a given training phase. The best hyperparameters for a given phase will be kept as the constant value for that hyperparameter in the following training phase. In the case where a hyperparameter has not been experimented with yet, an explanation will be added for what values were used.

1) Phase one: Learning parameters

The learning rate and the momentum for the training phase will be found using exhaustive search methods. For this reason, the learning rate and the momentum will be kept to small groups of values for the model to test. The choices for the learning rate will be drawn from the set: [0.0001, 0.001, 0.01, 0.1, 1]. The choices for the momentum value will be selected from the set: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9].

In this phase, the convolutional neural__. The number of hidden layers will be kept at one and the width of that layer will be 10 perceptrons.

The results of this experiment

2) Phase two: Hidden Layer Parameters

The number of hidden layers after the convolution phase will be varied from one to five. The model with the best

The kernel size for the convolutional layers

3) Phase three: Convolutional Parameters

C. Unit Price Linear Regressors

IV. Logistic Regressor Classifiers

A. Member Classifier

B. Gender Classifier

C. Decision Tree Classifier

D. Random Forest Classifier

V. Conclusion

VI. References

[1] Z. -Q. Zhao, P. Zheng, S. -T. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
[2] Zeebaree, Subhi & Haji, Lailan & Rashid, Imad & Zebari, Rizgar & Ahmed, Omar & Jacksi, Karwan & Shukur, Hanan. (2020). Multicomputer Multicore System Influence on Maximum Multi-Processes Execution Time. Test Engineering and Management. 83. 14921 - 14931.
[3] Obaid, Kavi & Zeebaree, Subhi & Ahmed, Omar. (2020). Deep Learning Models Based on Image Classification: A Review. International Journal of Social Science and Business. 4. 75-81. 10.5281/zenodo.4108433.