SINGLE-LAYER NEURAL NETWORK CLASSIFICATION MODELS FOR HANDWRITTEN DIGIT IMAGES BASED ON MNIST DATA

Apoorv Saraogee
Northwestern University, Artificial Intelligence and Deep Learning
October 21. 2022

Abstract

Handwriting data digitization is useful for many applications such as digital note-taking, recognizing postal mailing addresses and recognizing handwritten forms. Due to the complexity of imaging data, connectionist network models or neural networks have been gaining significant interest in recent years. This study analyzes a dataset of handwritten numeric digits as part of the standard benchmark MNIST dataset using various single hidden layer neural network models – single node, double node, many nodes, preprocessing inputs using principal component analysis (PCA) and a model with ranking inputs from a random forest analysis. Neural network models with 128 nodes in the hidden layer performed the best with 97% testing accuracy compared to 31%, 67%, 13% and 93% for the single node, double node, PCA and random forest models respectively.

1. Introduction

Handwritten documents have a lot of challenges for digitalization due to variable spacing, legibility, inherent noise from common variations, distortions and writing instrument. It is an unresolved problem to the satisfaction of the community and attractive to solve automatically due to the laborious nature of manual digitalization (Chiney et al., 2021). This study's central research topic is to analyze a dataset of handwritten digits derived from the National Institute of Standards and Technology used as a standard benchmark to compare machine learning models called MNIST which has 70,000 images (28x28pixels each). This study used 60,000 images for the training set and 10,000 images for the test set with 5,000 images used as a holdback from the training set for validation. Five different experiments were conducted with a single hidden layer and 10 nodes in the output layer corresponding to each of the digits (0 to 9). Number of nodes was varied between

1, 2 and 128 in the hidden layer of the neural network with input nodes based on the number of pixels in an image to elucidate the role of the activation values. Finally, number of input nodes was reduced as deforming input images was reported to have the best results with MNIST (Meier et al., 2011) and would also take less time for training. Principal components analysis and random forest classifier models were used to reduce the image dimensions from 784 pixels to 154 components and top 70 pixels respectively. Research questions will include comparing the performance of each of the experiments on the holdout validation dataset with the accuracy score. Other questions will look at the clustering of activation values in the hidden layer according to output class.

2. Literature Review

A variety of neural network classification models have been used on the MNIST dataset (Baldominos et al., 2019) including single-hidden layer perceptrons (Bettilyon, 2018; Meier et al., 2011), single-layer convolutional neural networks (Mcdonnell et al., 2015) and multi-layer neural networks (Bettilyon, 2018; Chen et al., 2018). The state of the art models are convolutional neural networks and have been shown to have less than 1% error rate in the current literature with error rates as low as 0.2%. Older models with linear classifiers were included with error rates ranging from 7.6-12% but recently have also been shown to have lower than 1% error rates as well. Also of note is that models have been trained with and without data preprocessing steps for the input, including PCA. Future studies would likely include the EMNIST dataset which was released in 2017 as MNIST which was released in 1998 is considered non-challenging today (Baldominos et al., 2019).

This study uses some of the same parameter settings as other studies such as input data preprocessing. Multiple layers did lead to overfitting and the single layer had the best performance and was subsequently chosen for the experiments in this study (Bettilyon, 2018). While some researchers only used 5 units (Ardi, 2020), other studies had notably used more units with 800 units(Meier et al., 2011) or as much as 2048 units (Bettilyon, 2018) in their single hidden layer network. This study does vary the number of units to verify this increase in model performance. This study also compares a linear classifier based on logistic regression with connectionist models with the first experiment with a single hidden layer and single node to test model performance.

3. Methods

This research will be conducted on the MNIST dataset of 70,000 handwritten digit images (28x28 pixels) using flattened and normalized 1D arrays as input. This corresponds to 784 features for the input and one-hot encoded classification outputs with 10 features for each digit from 0 to 9. Analysis is done in a Jupyter notebook using kernels for Python3 run locally with models in the keras and sklearn packages for modeling. The keras package is also used to load the MNIST dataset which is prepackaged as part of keras. Real handwritten digit images are used in the classification models with 10,000 images used for testing using accuracy scores. The remaining 60,000 images are used to train the neural networks with 5,000 images as a holdout for validation. Five different experiments were conducted with different neural network architectures as shown below:

- Experiment 1: 784 inputs, single node in hidden layer – same as logistic regression
- Experiment 2: 784 inputs, two nodes in hidden layer
- Experiment 3: 784 inputs, 128 nodes in hidden layer
- Experiment 4: 154 inputs with 95% variance from PCA, 128 nodes in hidden layer

- Experiment 5: top 70 inputs from random forest, 128 nodes in hidden layer

The key objectives include looking at activation function behavior with increasing nodes in the hidden layer using a boxplot and scatterplot as well as comparing the performance with data preprocessing steps (PCA/random forest) using the accuracy score on the test dataset. Confusion matrices were also constructed for each experiment to compare results.

3. Results

Accuracy scores for each of the tested experimental models are summarized in Table 1. The testing set accuracy scores for each of the tested models across the 10 digit image classes was highest for Experiment 3 with 97% accuracy compared to 31%, 67%, 13% and 93% for Experiments 1, 2, 4 and 5 respectively. The training/validation scores for Experiments 1-5 were the not significantly different from the test set except for Experiment 4 which saw a marked increase to 97% and 98% for the training and validation sets during training of the neural network. This decrease in model accuracy on the test set is indicative of overfitting.

Increasing number of nodes in hidden layer

Experiments 1, 2 and 3 explored how increasing the number of nodes in the hidden layer helped change model accuracy as the model captures more complex relationships with more nodes. In Experiment 1, we can see some overlap in the activation function values between the classes as shown in the boxplot in Figure 1. The separation of the activation function values between classes becomes greater when another node is added. The clustering of these 2 activation function values is shown in the scatterplot in Figure 2 and shows that increasing the number of nodes would increase the clustering of activation values with less overlap. Experiment 1 was unable to linearly

capture this complexity with the confusion matrix looking almost random as shown in Figure 3. Increasing the number of nodes in the model captures higher order interaction terms and performance. This is also shown in the literature where 2048 nodes produced a final test accuracy of 95% (Bettilyon, 2018).

Data preprocessing steps

Experiments 4 and 5 explored various data preprocessing steps – PCA and random forests respectively. PCA reduces the dimensionality of the input from a vector of 784 to 154 while attempting to capture information from the entire input space (95% variance of the training images was captured). In contrast, random forests reduced the dimensionality of the data by ranking the pixel values and taking the top 70 index values from the input image vector. This is a key difference in these two experiments and shows how using PCA led to overfitting and poor performance close to random with a training accuracy of 13% and confusion matrix shown in Figure 3. Since neural networks fit to the representation in the training and validation datasets, the model performed poorly with the PCA decomposition on the test dataset. In contrast, using ranked indexes worked well as the representation of the input space is the same – normalized pixel values. To get around this issue with PCA, others have used methods using wavelet transformations to generate representations that are more stable (Baldominos et al., 2019).

4. Conclusions

Within this study, Experiment 3 which employs a single hidden layer neural network model with 128 nodes and preserves the entire input space performs the best on the test sets with an accuracy score of 97%. This is quite good as it is comparable to state of the art models which are at ~99%

(Baldominos et al., 2019). It suggests that preserving the input variables and increasing the number of parameters has better performance. However, in more complex use cases, reducing dimensionality of the input is required due to the computational cost of training a neural network with many parameters. In this case, data preprocessing is required to reduce the number of training parameters. In real applications, random forests would be an attractive choice with a test accuracy of 93%. Future studies could explore other data preprocessing steps such as convolutional neural networks to extract sparse features that are highly important (Baldominos et al., 2019).

5. Appendices

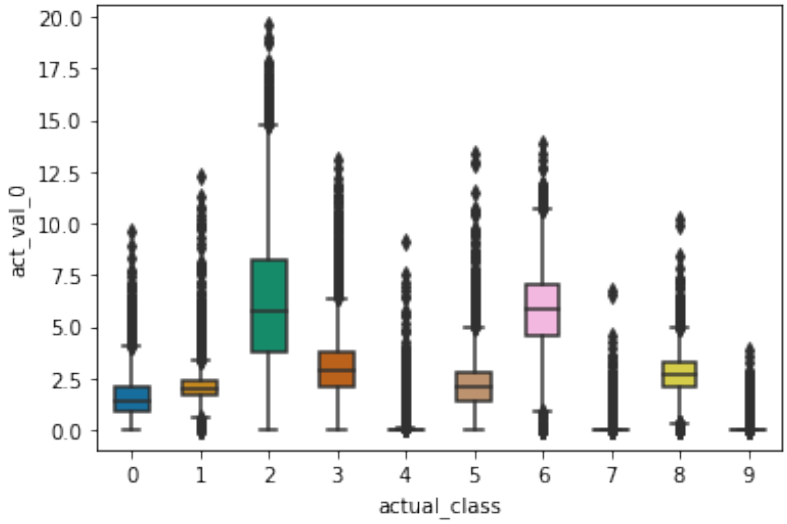


Figure 1. Boxplot of the class distribution of training activation values in the hidden layer for Experiment 1 with each class corresponding to that handwritten digit.

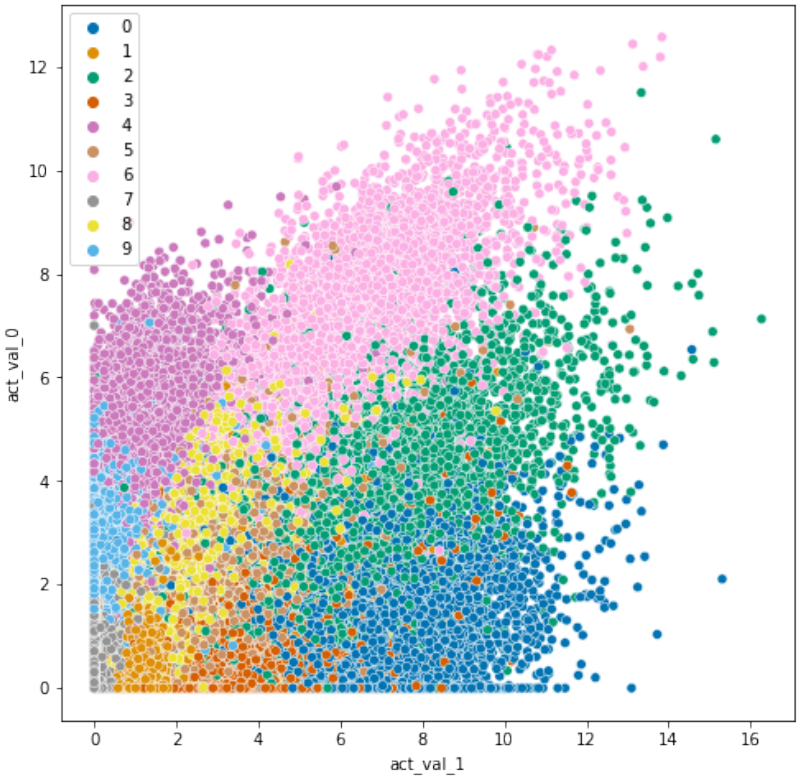
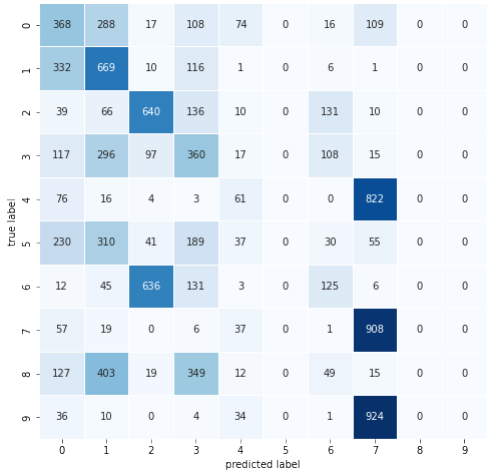
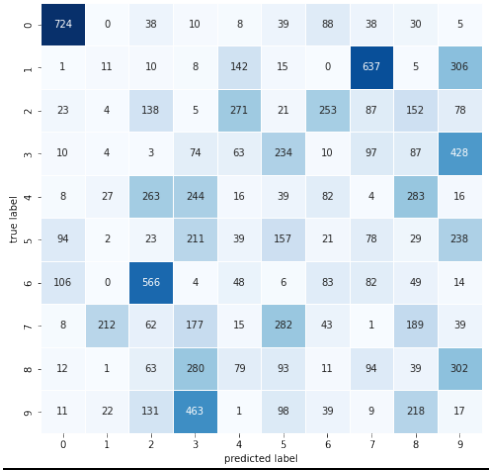


Figure 2. Scatterplot of the class distribution of training activation values in the hidden layer for Experiment 2 with each class corresponding to that handwritten digit as shown in the legend.

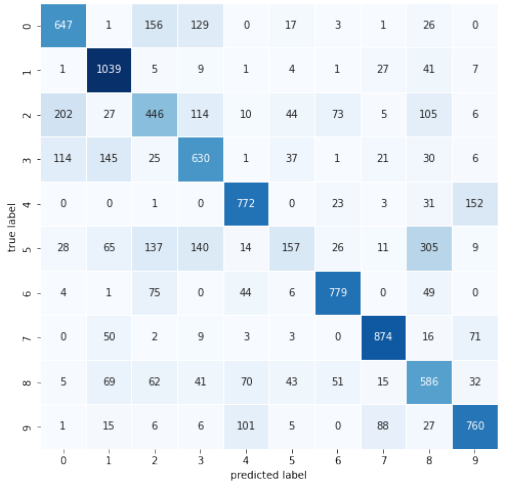
Experiment 1



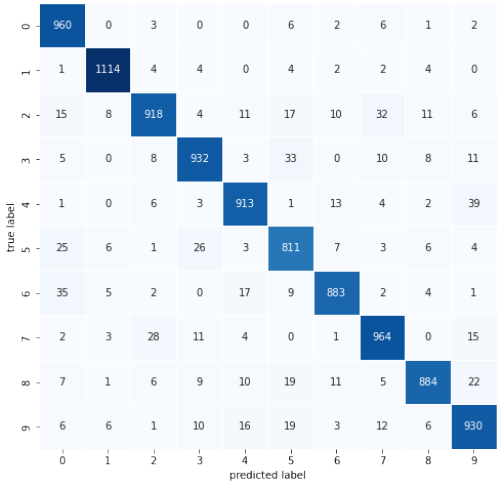
Experiment 4



Experiment 2



Experiment 5



Experiment 3

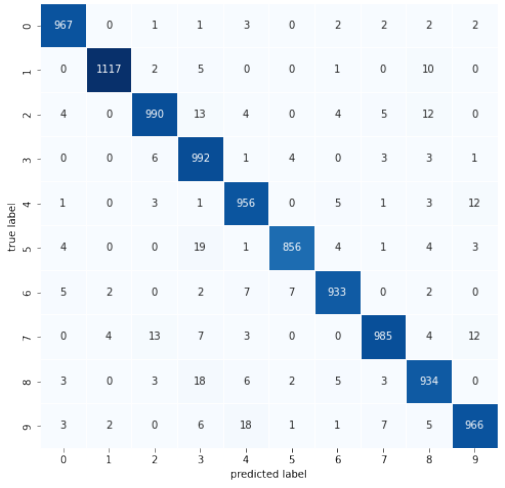


Figure 3. Confusion matrices for each experiment on the test dataset. Experiment 4 looks more random than Experiment 1 except for the classification for the digit 1 which may be simpler.

Table 1: Accuracy scores for each different experiment across test datasets. These look largely the same except for Experiment 4.

Experiment	Training	Validation	Testing
1	0.324	0.315	0.313
2	0.676	0.684	0.669
3	0.972	0.971	0.970
4	0.979	0.973	0.126
5	0.933	0.930	0.931

Supporting Files

- MSDS458_Assignment_01_exp1.html
- MSDS458_Assignment_01_exp2.html
- MSDS458_Assignment_01_exp3.html
- MSDS458_Assignment_01_exp4.html
- MSDS458_Assignment_01_exp5.html

References

Ardi, M. (2020). *Simple Neural Network on MNIST Handwritten Digit Dataset*. Becoming Human: Artificial Intelligence Magazine. https://becominghuman.ai/simple-neural-network-on-mnist-handwritten-digit-dataset-61e47702ed25

Baldominos, A., Saez, Y., & Isasi, P. (2019). A Survey of Handwritten Character Recognition with MNIST and EMNIST. *Applied Sciences*, *9*(25), 3169. https://doi.org/10.3390/app9153169

Bettilyon, T. E. (2018). *How to classify MNIST digits with different neural network architectures*. Medium. https://medium.com/tebs-lab/how-to-classify-mnist-digits-with-different-neural-network-architectures-39c75a0f03e3

Chen, F., Chen, N., Mao, H., & Hu, H. (2018). Assessing Four Neural Networks on Handwritten Digit Recognition Dataset (MNIST). *CHUANGXINBAN JOURNAL OF COMPUTING*.

Chiney, A., Paduri, A. R., Darapaneni, N., Kulkarni, S., Kadam, M., Kohli, I., & Subramaniyan, M. (2021). Handwritten Data Digitization Using an Anchor based Multi-Channel CNN (MCCNN) Trained on a Hybrid Dataset (h-EH). *Procedia Computer Science*, *189*, 175–182. https://doi.org/10.1016/J.PROCS.2021.05.095

Mcdonnell, M. D., Tissera, M. D., Vladusich, T., van Schaik, A., & Tapson, J. (2015). Fast, Simple and Accurate Handwritten Digit Classification by Training Shallow Neural Network Classifiers with the “Extreme Learning Machine” Algorithm. *PLoS ONE*, *10*(8). https://doi.org/10.1371/journal.pone.0134254

Meier, U., Claudiu, D., Cires¸an, C., Gambardella, L. M., & Schmidhuber, J. (2011). Better digit recognition with a committee of simple Neural Nets. *International Conference on Document Analysis and Recognition*. https://ieeexplore.ieee.org/abstract/document/6065510