NORTHWESTERN UNIVERSITY | SCHOOL OF PROFESSIONAL STUDIES

A.1 First Research/Programming Assignment: MNIST Classification

MS DSP 458 - Artificial Intelligence and Deep Learning

Sachin Sharma

October 07, 2024

Abstract

This research explores the impact of varying neural network architectures on the classification of handwritten digits using the MNIST dataset. A series of five experiments were conducted, each focusing on a dense neural network with a single hidden layer. The experiments progressively increased the number of hidden nodes and analyzed how these variations influenced model performance. Key insights were drawn from the hidden nodes' activation values, providing an understanding of their role in feature extraction. Additionally, the study incorporated dimensionality reduction techniques, such as Principal Component Analysis (PCA), and feature selection via Random Forests to compare their effects on model efficiency and accuracy. Results indicate that while increasing the number of hidden nodes initially improves accuracy, there is a point of diminishing returns. Dimensionality reduction was found to improve computational efficiency without a significant loss in performance, highlighting the trade-off between model complexity and computational cost.

Introduction

The classification of handwritten digits is a key problem in machine learning, often used as a starting point to explore neural network architectures. This study utilizes the MNIST dataset, which contains 70,000 grayscale images of handwritten digits (0-9), to investigate the performance of dense neural networks with a single hidden layer. By varying the number of hidden nodes and incorporating dimensionality reduction techniques such as Principal Component Analysis (PCA), we aim to understand how network architecture and feature reduction impact classification accuracy and computational efficiency.

Literature Review

The MNIST dataset has been a cornerstone for evaluating neural network architectures. Akmaljon Palvanov and Young Im Cho (2018) compared Capsule Networks

(CapsNet), ResNet, and CNNs, finding CapsNet to be superior but more complex than our single-layer approach.

While MNIST has been a standard for decades, early OCR advancements, like the perceptron (Rosenblatt, 1958), paved the way for modern neural networks. Techniques like PCA (Jolliffe, 2002) and Random Forest (Breiman, 2001) have also been used for dimensionality reduction. Our study builds upon these foundations, focusing on simple architectures and assessing performance across various configurations.

Recent studies have explored more advanced techniques, such as transfer learning (Yosinski et al., 2014) and generative adversarial networks (GANs) (Goodfellow et al., 2014), to improve MNIST classification. However, our research aims to demonstrate the effectiveness of simpler models, highlighting their potential for real-world applications.

Methods

The experiments were conducted on a Mac OS machine with GPU support to accelerate computation. The MNIST dataset, consisting of 60,000 training images and 10,000 test images, was used for training, validation, and testing. To ensure reliable evaluation, 5,000 images were set aside for validation. Each image, originally 28x28 pixels, was flattened into 784-dimensional vectors and normalized with pixel values between 0 and 1 to be compatible with our neural network models.

An exploratory data analysis (EDA) was performed to better understand the variations in handwritten digits and their distribution across the dataset, revealing potential challenges in classification due to inconsistencies in how digits are written (e.g., variations in the number ‘7’ or ‘3’). The relatively balanced distribution of digits provided a stable foundation for training.

Our research design employed multi-layer perceptrons (MLPs) built using the Keras and TensorFlow frameworks. The models were sequential dense layers with ReLU activation functions in the hidden layers and softmax in the output layer to classify the 10-digit classes. We experimented with different hidden layer sizes and applied dimensionality reduction techniques such as Principal Component Analysis (PCA) and Random Forest feature selection to reduce the input features and evaluate their impact on classification performance.

The RMSprop optimizer was used for model training, with Sparse Categorical Cross-Entropy as the loss function, suitable for multi-class classification. The best model weights were saved based on validation accuracy, and performance metrics, including accuracy and loss, were computed on both validation and test sets.

Dimensionality reduction was explored by applying PCA to retain 95% of the variance, reducing the number of input features. Another experiment used Random Forest to identify the 70 most important features for classification. These techniques helped assess the balance between reducing complexity and maintaining model performance. See Appendix A for a detailed table outlining the methodology of each experiment.

Results

This section summarizes the results from our experiments, including accuracy, loss, and confusion matrices. All referenced graphs and figures can be found in the appendix.

In **Experiment 1**, we started with a simple neural network consisting of a single hidden node and 784 input features, representing the pixel data from 28x28 images of digits. After training the model for 30 epochs, it achieved a test accuracy of 38.17%, which, while better than random guessing (10%), highlighted significant limitations. The confusion matrix (Appendix B) indicated substantial misclassification, particularly between digits ‘1’ and ‘9,’ which the model failed to predict. Moreover, the boxplot analysis of activation values (Appendix E) showed a considerable overlap between activation outputs for different digits, revealing the model's difficulty in differentiating between classes. Despite its limitations, this experiment provided a useful baseline for evaluating more complex models.

In **Experiment 2**, we increased the hidden layer to include two nodes, maintaining the same input dimensions. This modification significantly improved the model's performance, with the test accuracy increasing to 65.71%. The confusion matrix (Appendix B) demonstrated much better predictions across digit classes compared to Experiment 1, although some issues, particularly with misclassifying '3,' persisted. Scatterplots of activation values (Appendix E) also indicated that this model was better at clustering digit classes, suggesting an enhanced ability to separate the data. Nonetheless, certain misclassifications still occurred, as seen in Appendix C, showing that while the model improved, further refinement was needed to achieve higher accuracy.

Experiment 3 involved a substantial increase in model complexity, as we expanded the hidden layer to 128 nodes. This adjustment resulted in a dramatic improvement in performance, with the model reaching a test accuracy of 97.53% after 30 epochs and a minimal test loss of 0.0873. The confusion matrix (Appendix B) revealed near-perfect classification, with most predictions correctly aligning with actual digit labels. However, a few misclassifications remained, particularly between digits '4' and '9,' which are visually similar. The activation value visualizations (Appendix E) demonstrated that the model effectively separated digit classes, although the increased complexity introduced challenges in interpretability. Overall, this experiment showcased the power of deeper neural networks for classification tasks.

In **Experiment 4**, we applied Principal Component Analysis (PCA) to reduce the input dimensions from 784 to 154, retaining 95% of the variance. Using this reduced input, we trained a neural network with 85 hidden nodes, achieving an impressive test accuracy of 97.65%. This result surpassed that of the deeper network from Experiment 3, showing that dimensionality reduction could improve performance while simplifying the model. Although the accuracy and loss curves (Appendix D) suggested potential overfitting, the model's strong performance indicated the effectiveness of PCA in maintaining predictive power with fewer features. This experiment underscored the potential of combining dimensionality reduction with neural networks to achieve high accuracy without the computational burden of large input spaces. This experiment yield the most higher result in comparison to all our experiments, which we have highlighted in table in Appendix A.

In **Experiment 5**, we introduced a Random Forest classifier for feature selection, reducing the input data to the 70 most important pixels. This new feature set was fed into a neural network with 85 hidden nodes, which achieved a test accuracy of 93.68% after 30 epochs. Although the accuracy was slightly lower than in Experiment 4, the confusion matrix (Appendix B) showed that the model performed reasonably well with significantly fewer input features. The heatmap of pixel importance (Appendix E) confirmed that the most important pixels were concentrated near the center of the digit images, reinforcing the utility of feature selection. While this model performed admirably, the training and validation curves (Appendix D) suggested potential overfitting, indicating room for further optimization, such as applying regularization techniques to improve generalization.

In summary, the results of our experiments, presented in **Appendix A**, show that increasing the number of hidden units generally improves model performance, significantly enhancing accuracy and reducing misclassification rates. Moreover, effective dimensionality reduction techniques, such as PCA and Random Forests, can maintain high accuracy with fewer features, suggesting a path forward for optimizing neural network architectures. These findings set the stage for further research into more complex architectures, advanced feature selection methods, and hybrid models combining deep learning with traditional techniques.

Conclusions

The experiments conducted in this study demonstrate the power and flexibility of neural networks for digit classification, highlighting the impact of model architecture and feature selection on performance. Starting with a simple neural network architecture, we observed that even a model with a single hidden node could achieve accuracy above random chance, though its limitations were clear. By progressively increasing the complexity of the network, particularly by adding more hidden nodes, we were able to significantly enhance model performance, achieving near-perfect accuracy in some cases from **38%** to **99%**. However, this improvement came with trade-offs, such as increased potential for overfitting as the number of parameters grew.

Dimensionality reduction techniques, particularly Principal Component Analysis (PCA) and Random Forest-based feature selection, proved highly effective in maintaining high model accuracy while reducing the input feature space. These methods not only improved model efficiency but also highlighted that feature selection can preserve essential information without compromising predictive performance. The success of PCA, in particular, showed that reducing the dimensionality of input data can be a valuable tool in improving both model performance and interpretability.

References

Palvanov A, , Cho YI. **Comparisons of Deep Learning Algorithms for MNIST in Real-Time Environment**. IJFIS 2018;18:126-134. https://doi.org/10.5391/IJFIS.2018.18.2.126

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386–408. https://doi.org/10.1037/h0042519.

Jolliffe, I. T. (2002). Principal component analysis. Springer Science & Business Media.

Breiman, L. Random Forests. *Machine Learning* **45**, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14). MIT Press, Cambridge, MA, USA, 3320–3328.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14). MIT Press, Cambridge, MA, USA, 2672–2680.

Keras. 2022. “MNIST digits classification dataset.” *Keras*. https://keras.io/api/datasets/mnist/

Javier Martinez Ojeda. 2022. “Digit Classification with Single-Layer Perceptron. *Github*. https://towardsdatascience.com/digit-classification-with-single-layer-perceptron-9a4e7d4d9628

Appendix A – Accuracy Results From Experiments

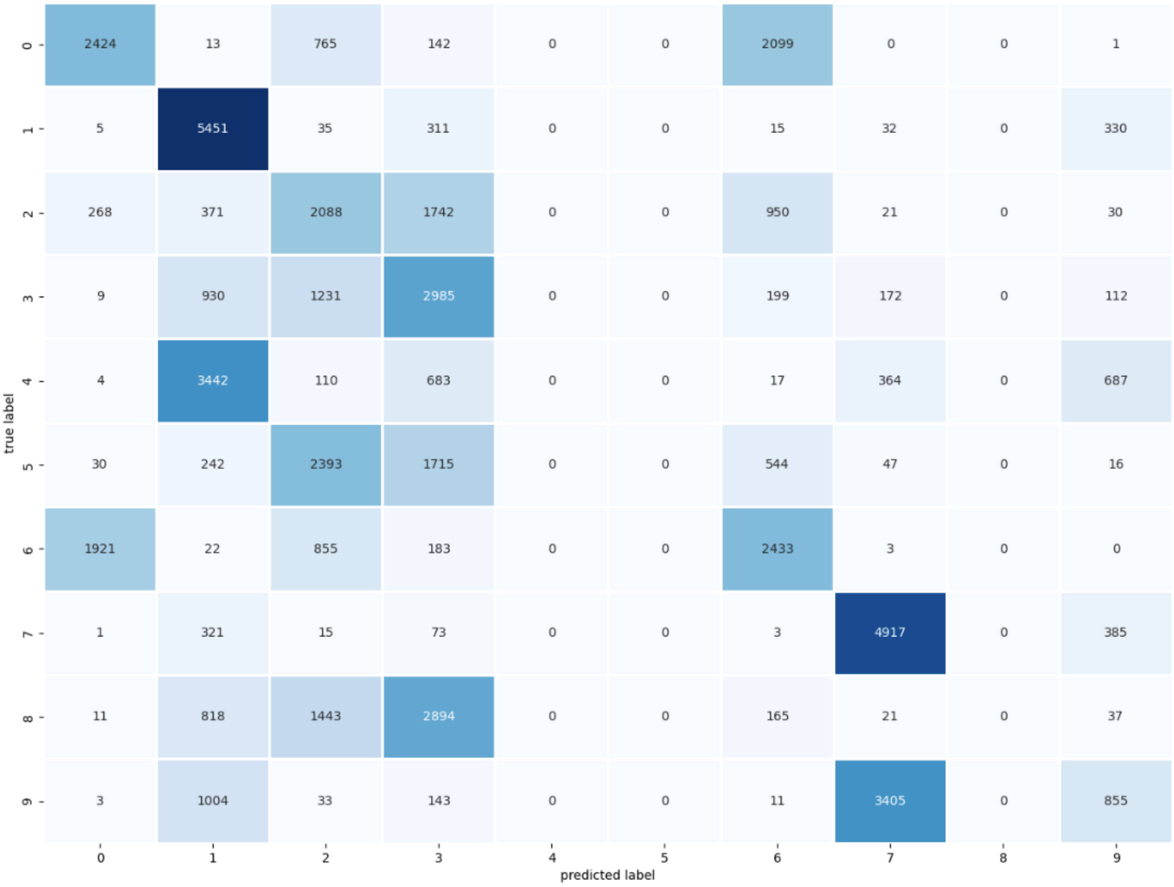
The table below displays the training, validation, and testing accuracy of each of the 5 neural network models developed for classification of MNIST images as digits.

Experiments	Input Dimensions	Hidden Layer	Output Layer	Dimensionality Reduction	Training Accuracy	Test Accuracy	Test Loss
1	784 (flattened 28x28)	1 node	10 nodes	None	0.3846	0.3817	1.5491
2	784 (flattened 28x28)	2 nodes	10 nodes	None	0.6595	0.6571	1.0712
3	784 (flattened 28x28)	128 nodes	10 nodes	None	0.9862	0.9753	0.0873
4	154 (after PCA)	85 nodes	10 nodes	PCA (95% variance, 154 features)	0.9946	0.9765	0.0824
5	70 (after Random Forest)	85 nodes	10 nodes	Random Forest (70 features)	0.9544	0.9368	0.2241

Appendix B – Confusion Matrices Resulting from Experiments

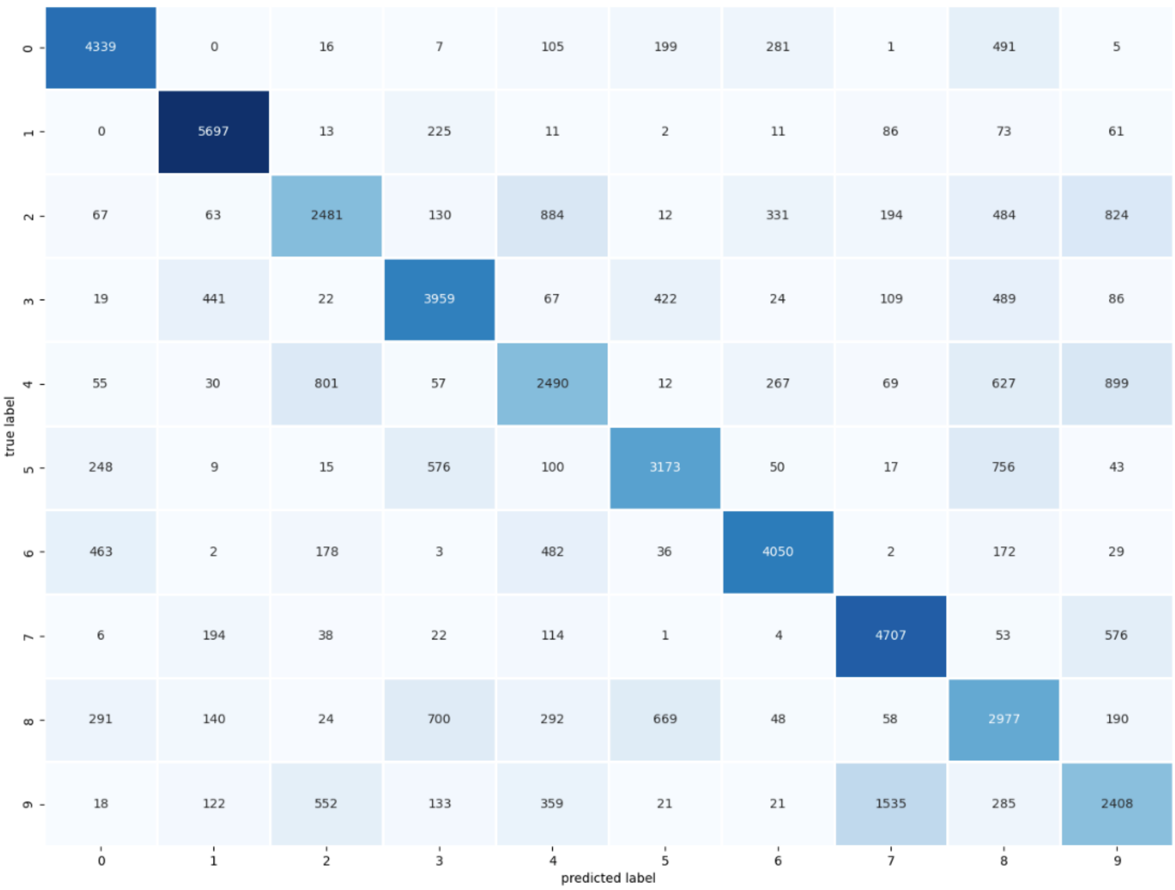
The images below display the confusion matrices resulting from the application of each MNIST neural network classification model the testing dataset. For ease of interpretation, these confusion matrices are color coded as heat maps.

Experiment 1:


true label
predicted label

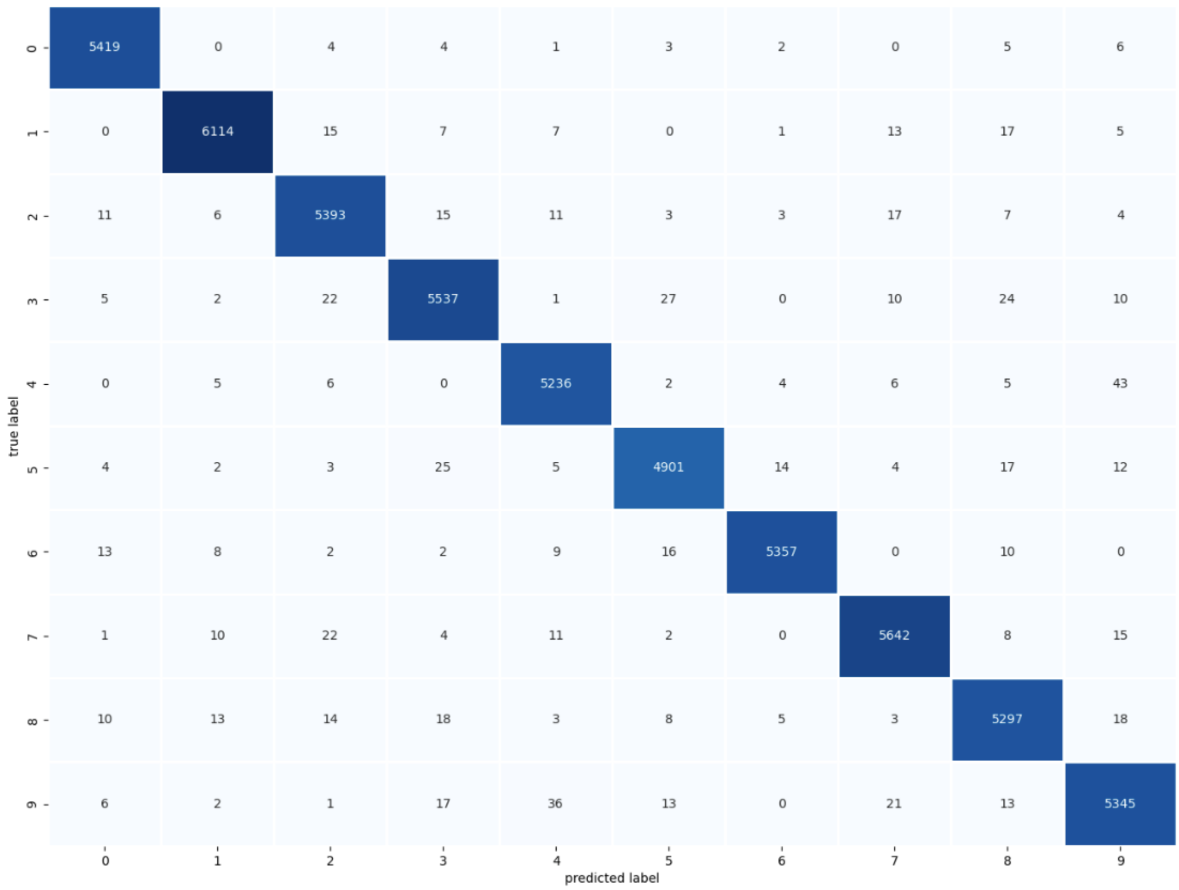
	0	1	2	3	4	5	6	7	8	9
0	2424	13	765	142	0	0	2099	0	0	1
1	5	5451	35	311	0	0	15	32	0	330
2	268	371	2088	1742	0	0	950	21	0	30
3	9	930	1231	2985	0	0	199	172	0	112
4	4	3442	110	683	0	0	17	364	0	687
5	30	242	2393	1715	0	0	544	47	0	16
6	1921	22	855	183	0	0	2433	3	0	0
7	1	321	15	73	0	0	3	4917	0	385
8	11	818	1443	2894	0	0	165	21	0	37
9	3	1004	33	143	0	0	11	3405	0	855

Experiment 2:



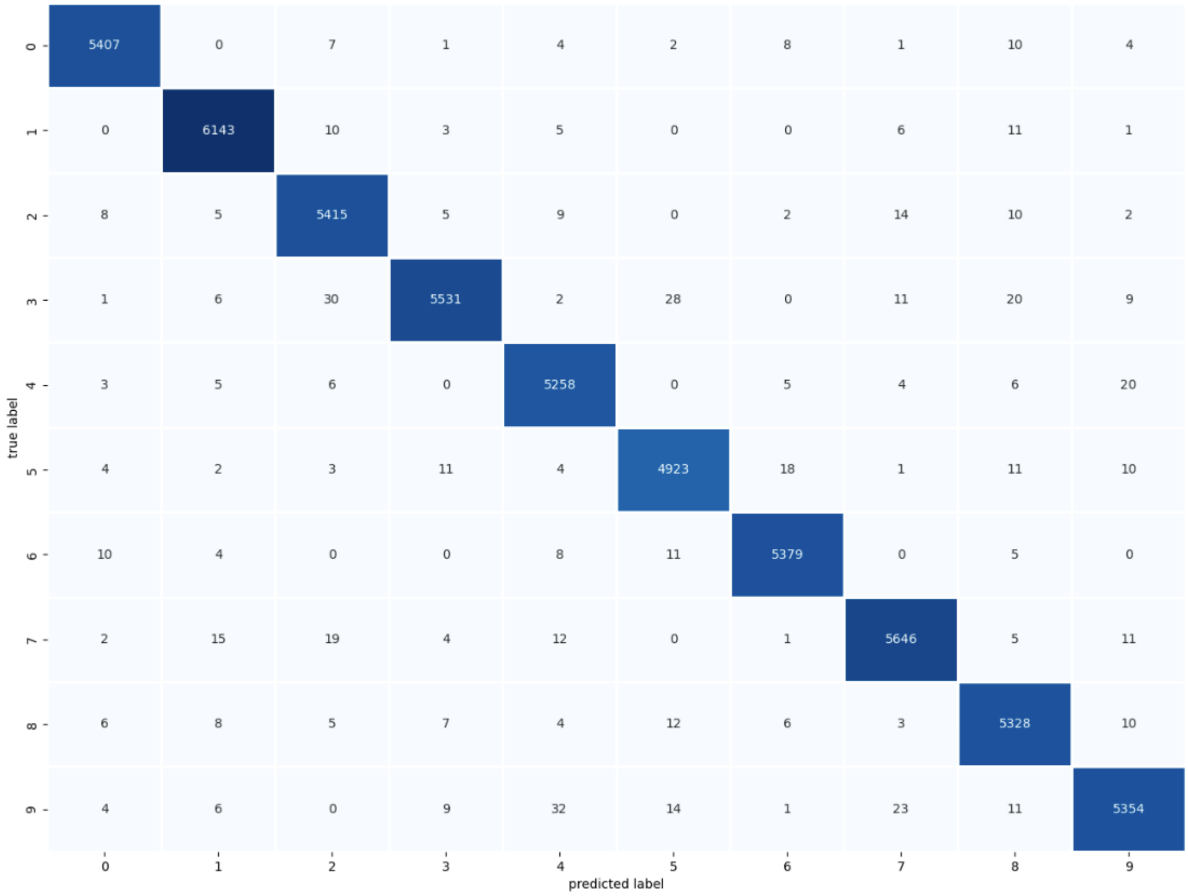
true label \ predicted label	0	1	2	3	4	5	6	7	8	9
0	4339	0	16	7	105	199	281	1	491	5
1	0	5697	13	225	11	2	11	86	73	61
2	67	63	2481	130	884	12	331	194	484	824
3	19	441	22	3959	67	422	24	109	489	86
4	55	30	801	57	2490	12	267	69	627	899
5	248	9	15	576	100	3173	50	17	756	43
6	463	2	178	3	482	36	4050	2	172	29
7	6	194	38	22	114	1	4	4707	53	576
8	291	140	24	700	292	669	48	58	2977	190
9	18	122	552	133	359	21	21	1535	285	2408

Experiment 3:



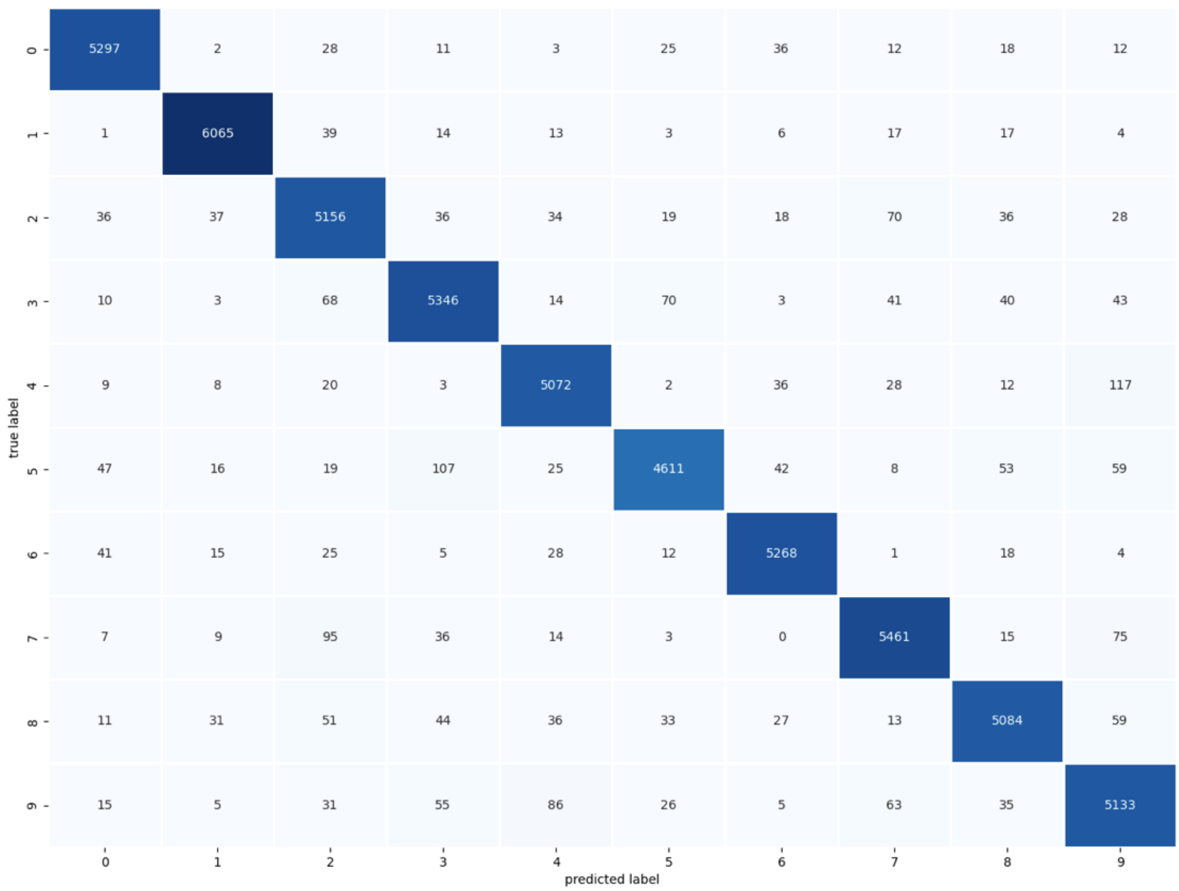
	0	1	2	3	4	5	6	7	8	9
0	5419	0	4	4	1	3	2	0	5	6
1	0	6114	15	7	7	0	1	13	17	5
2	11	6	5393	15	11	3	3	17	7	4
3	5	2	22	5537	1	27	0	10	24	10
4	0	5	6	0	5236	2	4	6	5	43
5	4	2	3	25	5	4901	14	4	17	12
6	13	8	2	2	9	16	5357	0	10	0
7	1	10	22	4	11	2	0	5642	8	15
8	10	13	14	18	3	8	5	3	5297	18
9	6	2	1	17	36	13	0	21	13	5345

Experiment 4:



	0	1	2	3	4	5	6	7	8	9
0	5407	0	7	1	4	2	8	1	10	4
1	0	6143	10	3	5	0	0	6	11	1
2	8	5	5415	5	9	0	2	14	10	2
3	1	6	30	5531	2	28	0	11	20	9
4	3	5	6	0	5258	0	5	4	6	20
5	4	2	3	11	4	4923	18	1	11	10
6	10	4	0	0	8	11	5379	0	5	0
7	2	15	19	4	12	0	1	5646	5	11
8	6	8	5	7	4	12	6	3	5328	10
9	4	6	0	9	32	14	1	23	11	5354

Experiment 5:

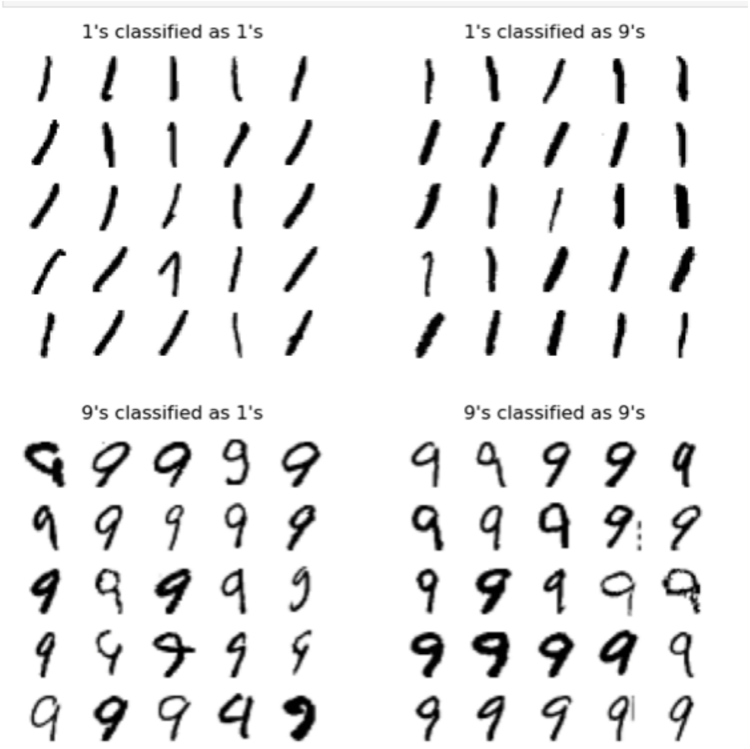


	0	1	2	3	4	5	6	7	8	9
0	5297	2	28	11	3	25	36	12	18	12
1	1	6065	39	14	13	3	6	17	17	4
2	36	37	5156	36	34	19	18	70	36	28
3	10	3	68	5346	14	70	3	41	40	43
4	9	8	20	3	5072	2	36	28	12	117
5	47	16	19	107	25	4611	42	8	53	59
6	41	15	25	5	28	12	5268	1	18	4
7	7	9	95	36	14	3	0	5461	15	75
8	11	31	51	44	36	33	27	13	5084	59
9	15	5	31	55	86	26	5	63	35	5133

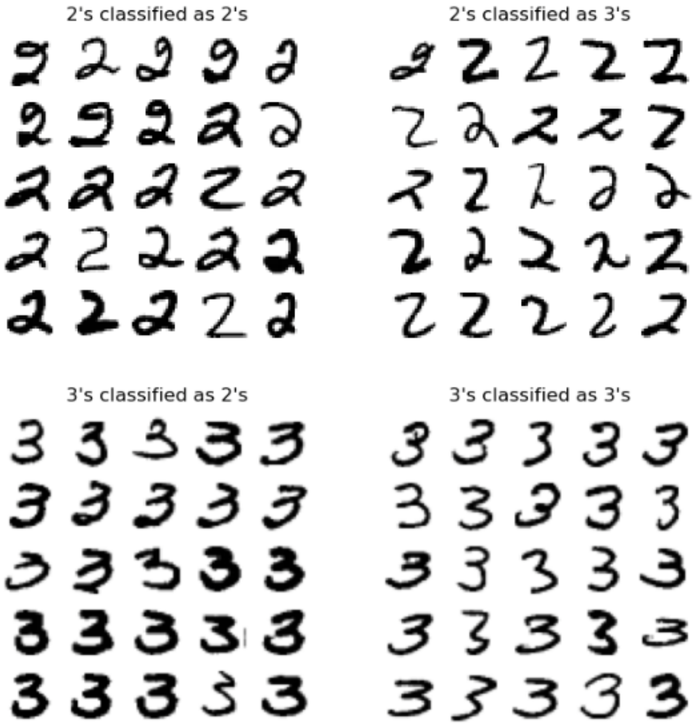
Appendix C – Most Commonly Misclassified Digit Combinations Resulting from Experiments

The images below display examples of some of the most frequently misclassified digits resulting from each of the MNIST image digit classification models.

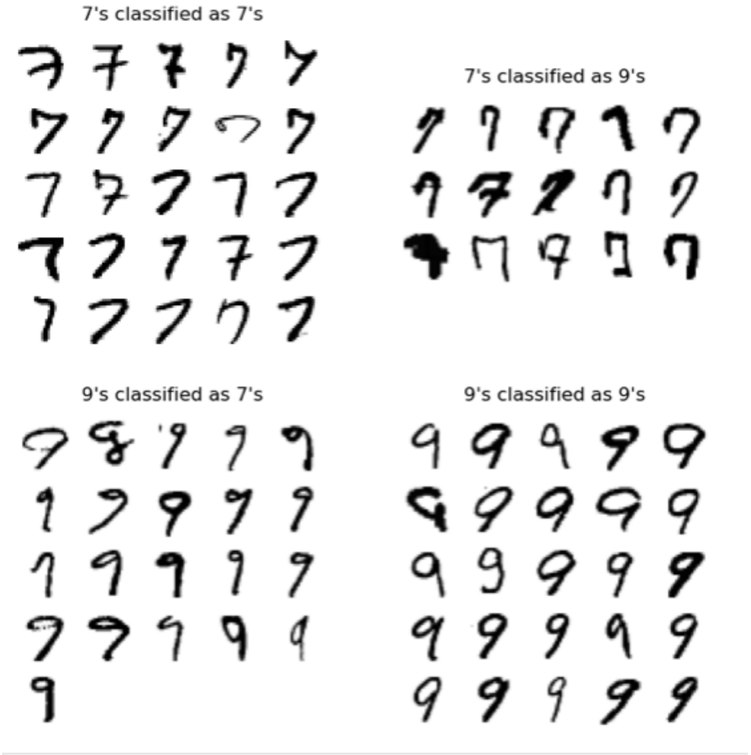
Experiment 1:



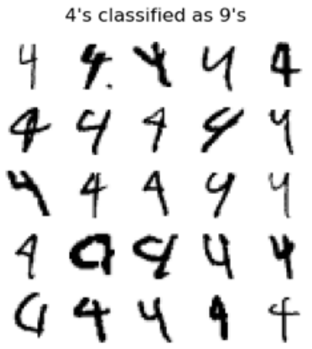
Experiment 2:

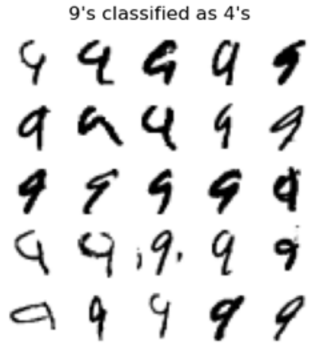


Experiment 3:

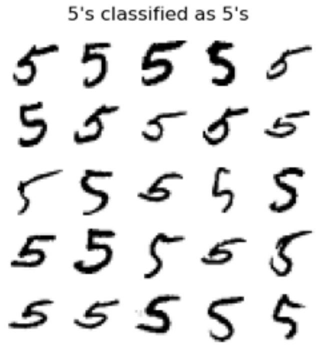


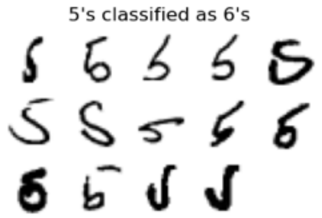
4's classified as 4's

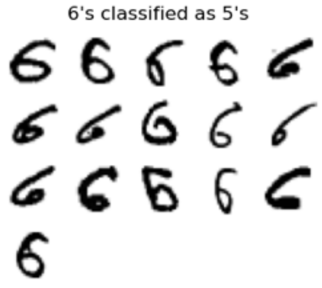

4's classified as 9's

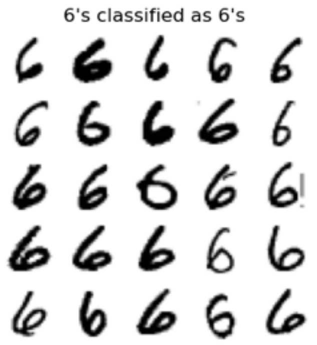

9's classified as 4's

9's classified as 9's

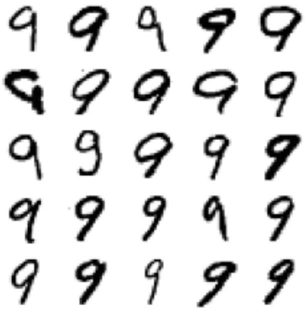

5's classified as 5's


5's classified as 6's

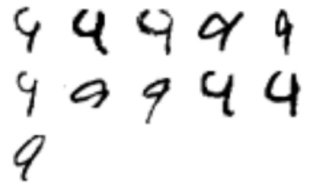

6's classified as 5's


6's classified as 6's

Experiment 4:

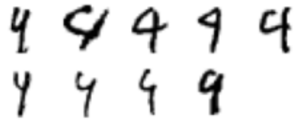


9's classified as 9's



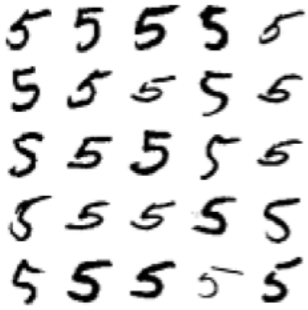
9's classified as 4's

4's classified as 4's

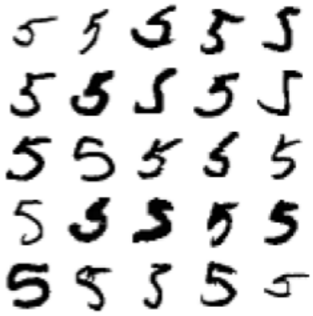


4's classified as 9's

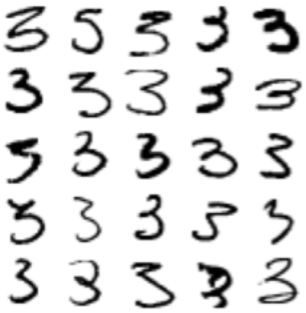
Experiment 5:



5's classified as 5's



5's classified as 3's



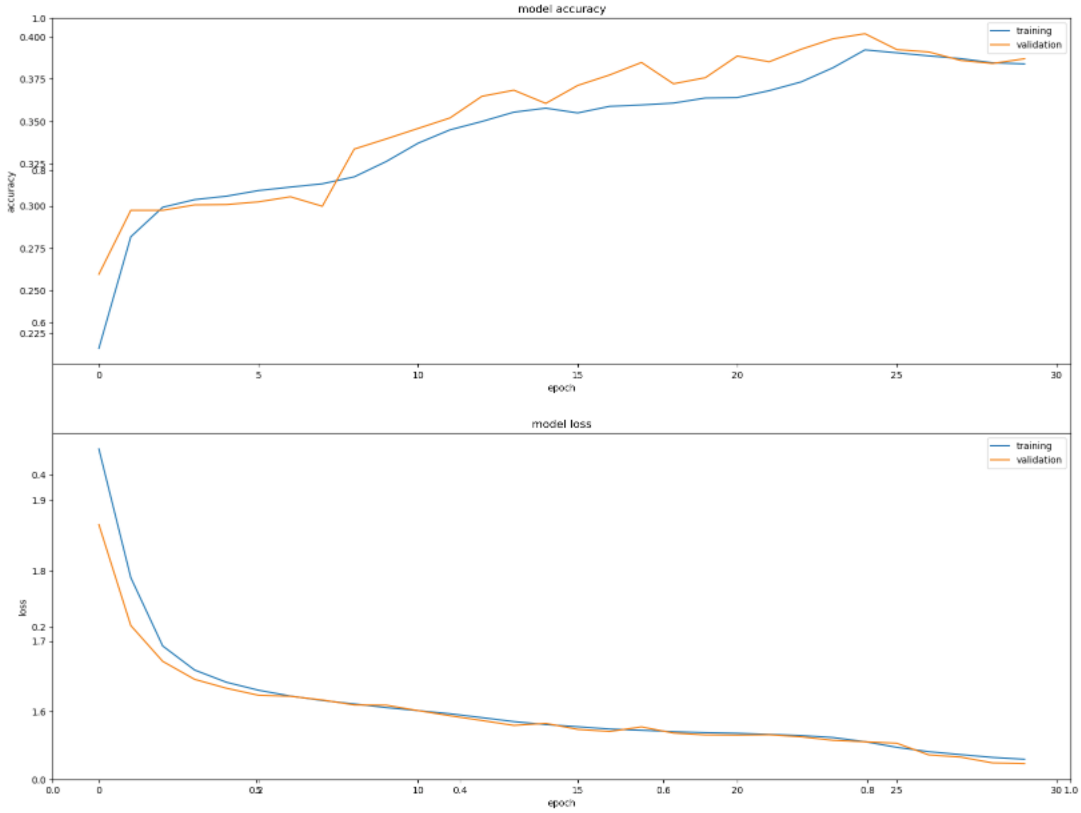
3's classified as 5's

3's classified as 3's

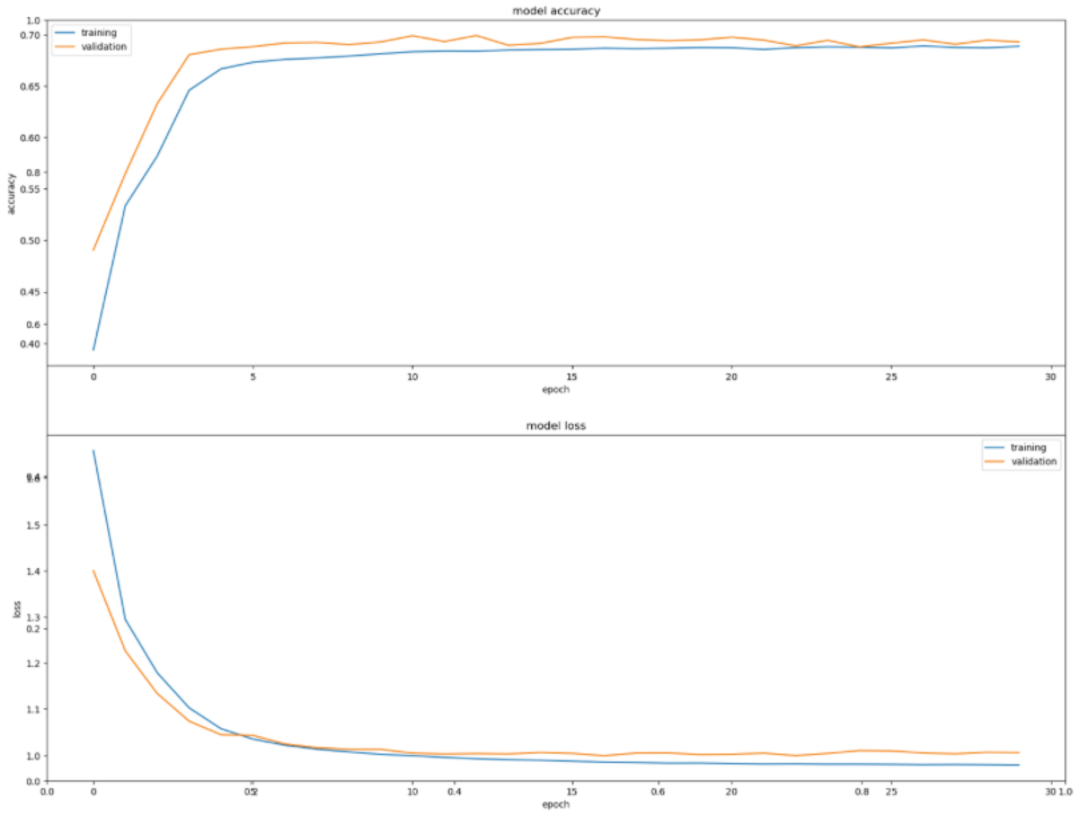
Appendix D – Accuracy and Loss Trends by Epoch Resulting from Experimental Model Fitting

The graphs below display the training and validation accuracies generated throughout each epoch of training each of the MNIST image digit classification models.

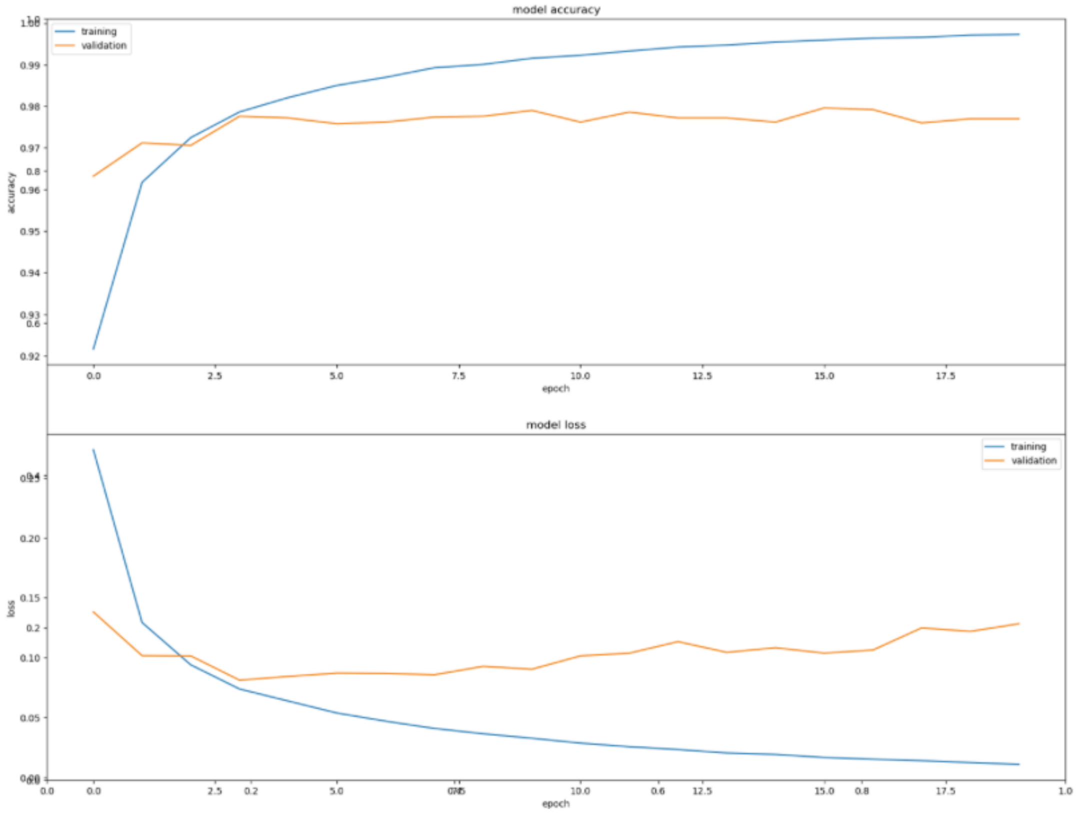
Experiment 1:



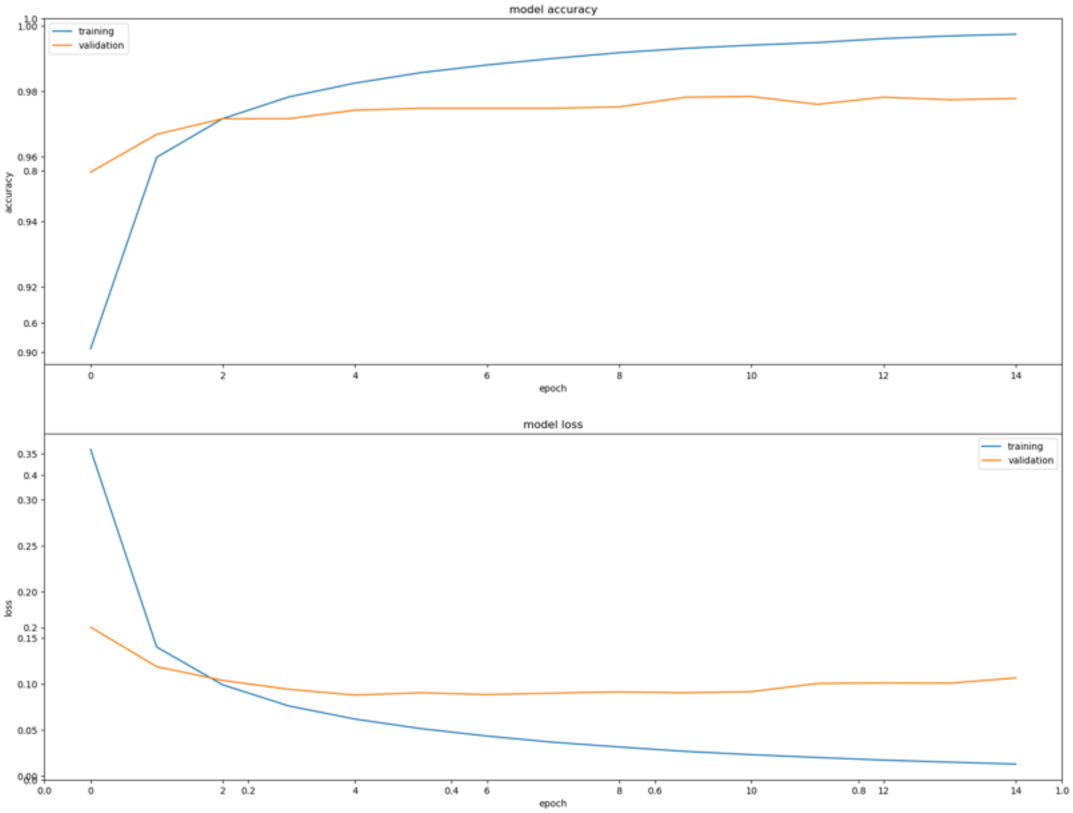
Experiment 2:



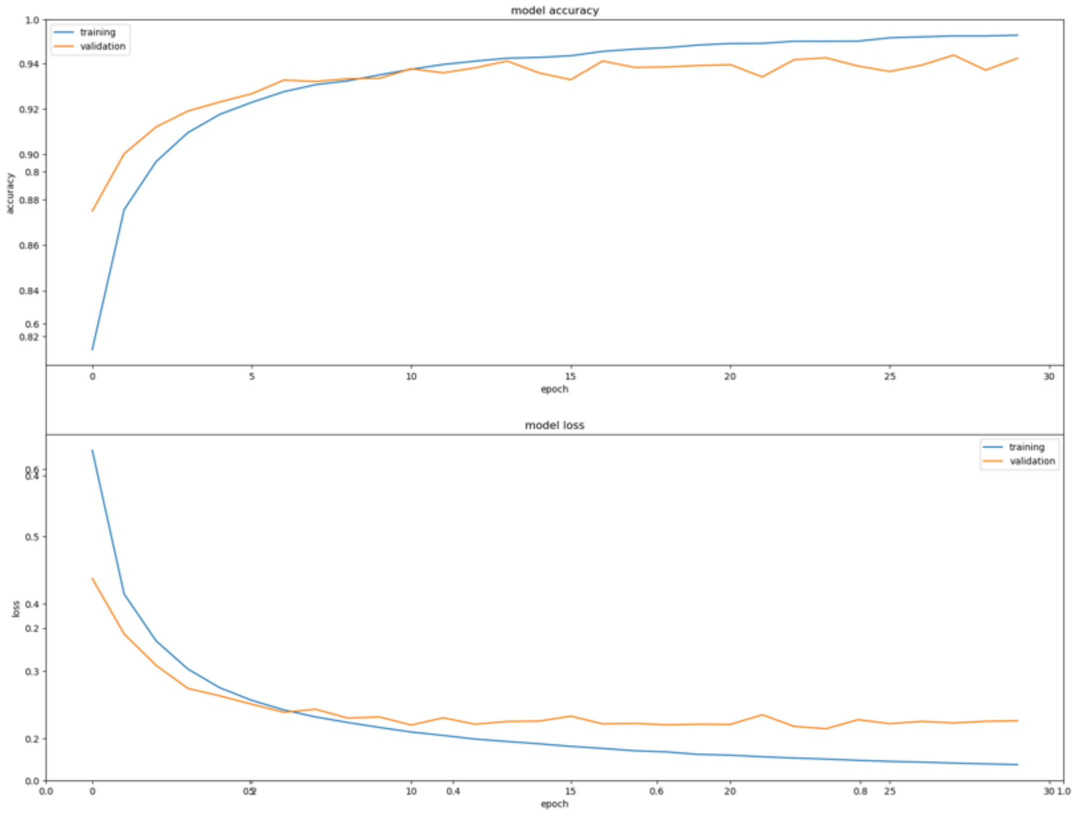
Experiment 3:



Experiment 4:



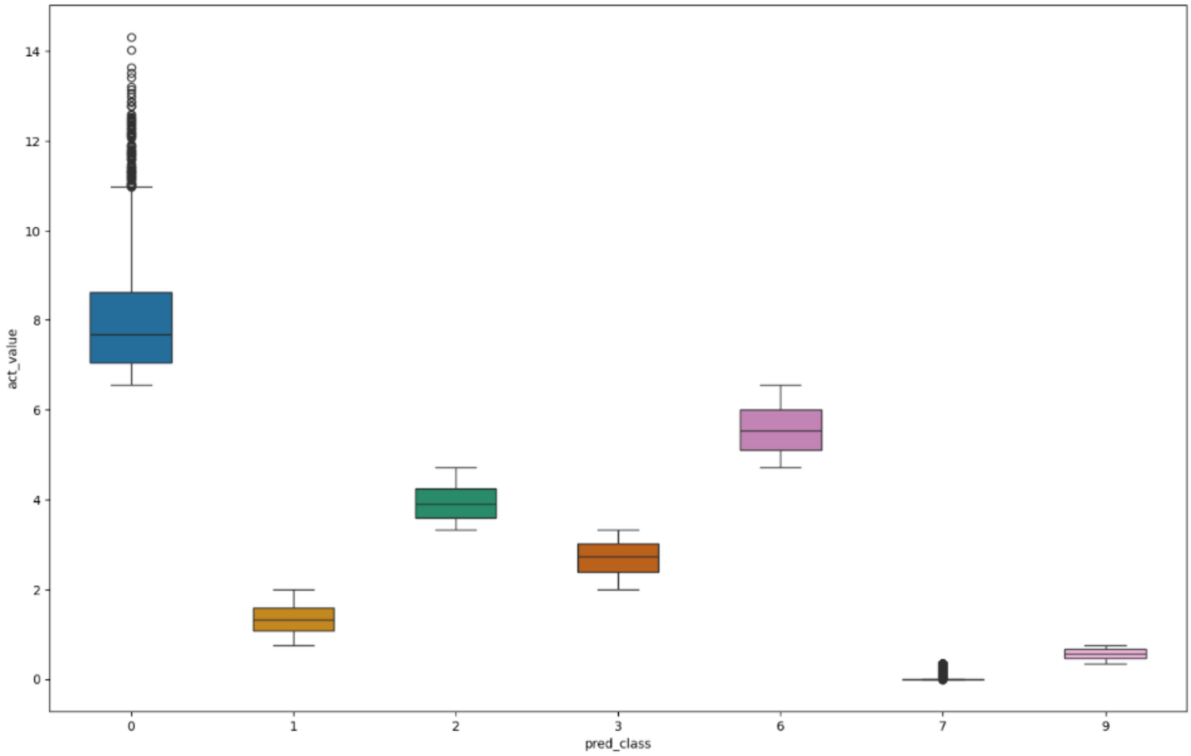
Experiment 5:


model accuracy
training
validation
1.0
0.94
0.92
0.90
0.8
0.88
0.86
0.84
0.6
0.82
accuracy
0
5
10
15
20
25
30
epoch
model loss
training
validation
0.6
0.4
0.5
0.4
0.2
0.3
0.2
0.0
loss
0.0
0
0.2
5
10
0.4
15
0.6
20
0.8
25
30
1.0
epoch

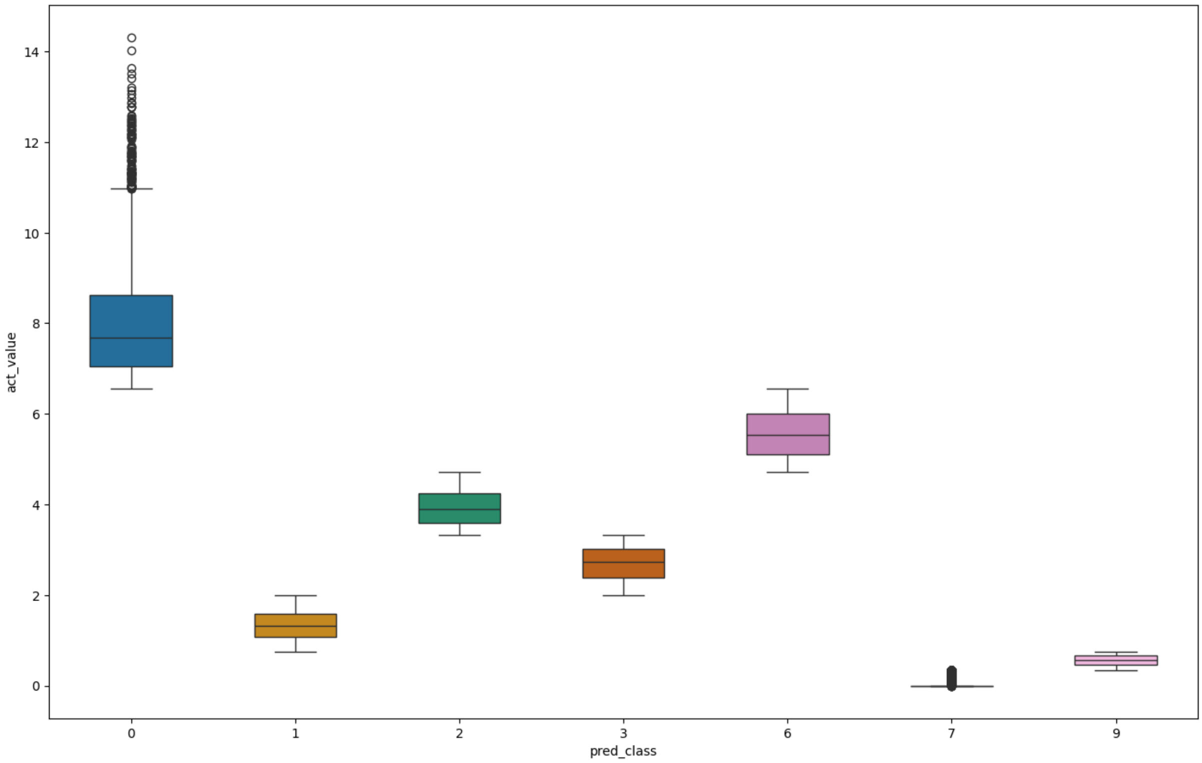
Appendix E – Activation Value Analyses from Experiments

The images below visualize the results of activation value experiments designed to provide insight into features extracted by the neural network model and how well the models are discriminating between digits using these extracted features.

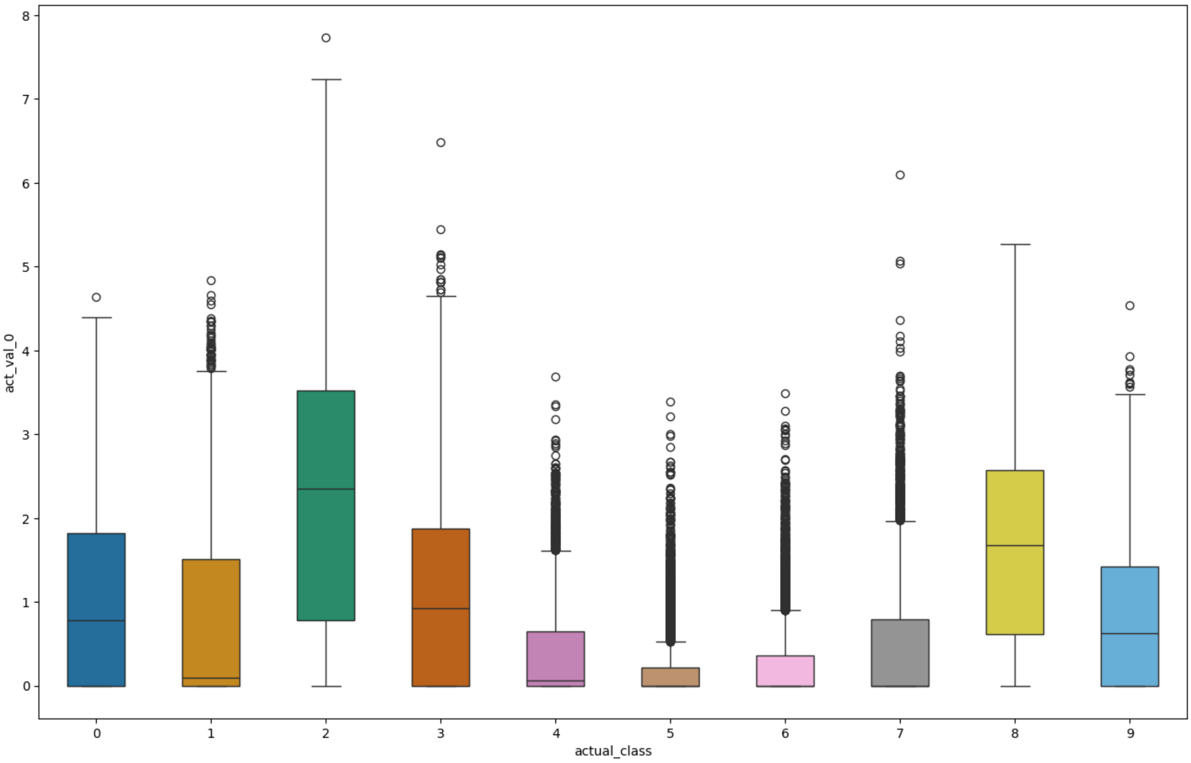
Experiment 1:



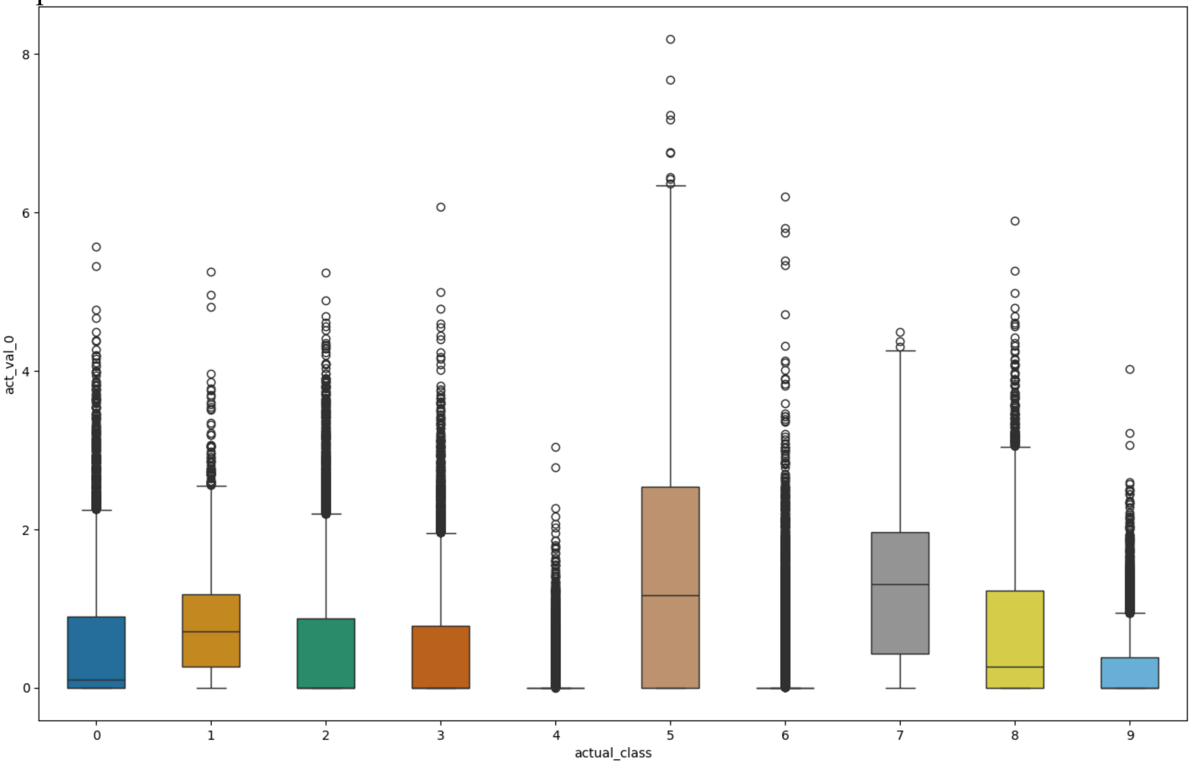
Experiment 2:



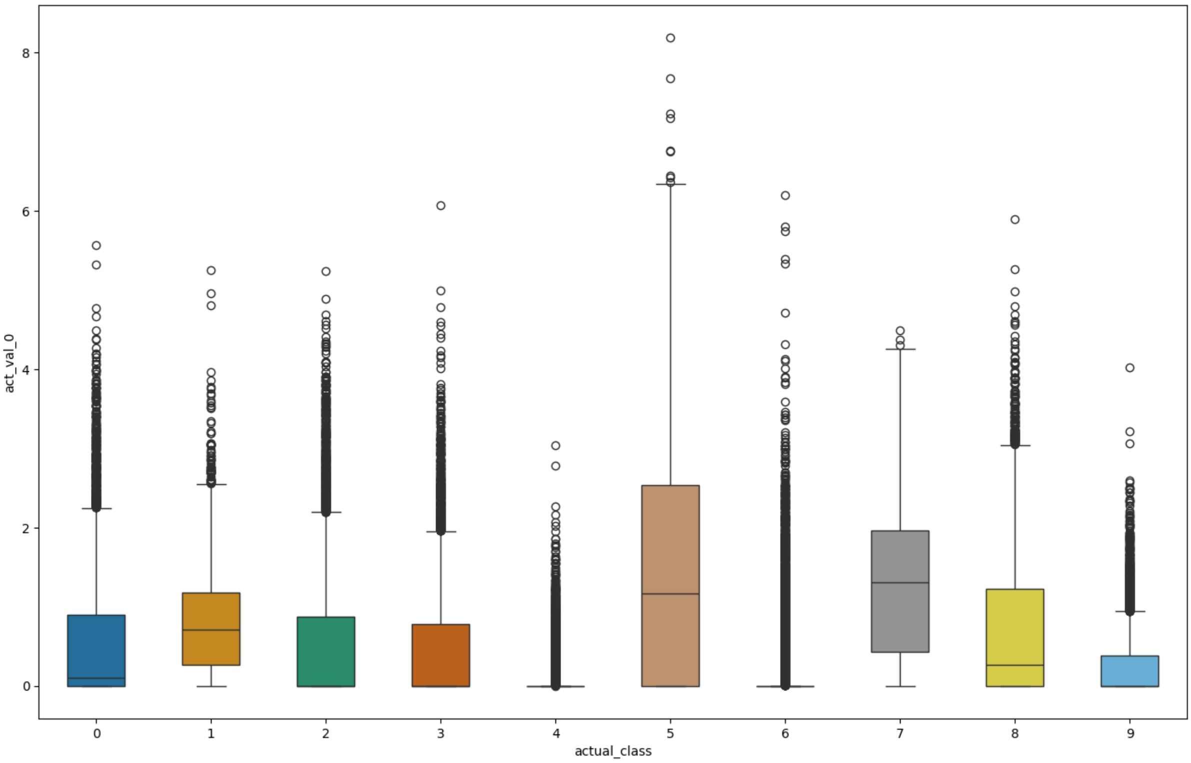
Experiment 3:



Experiment 4:



Experiment 5:


act_val_0
0
2
4
6
8
actual_class
0
1
2
3
4
5
6
7
8
9