Report on Multi-Stage Classification Model on Digits Dataset

Sarthak Thakare

May 4, 2025

Abstract

This report presents the design and evaluation of a multi-stage classification model for the 10-class handwritten digits dataset. The architecture integrates a Support Vector Machine (SVM) for binary separation and a Feedforward Neural Network (FFNN) for probability estimation and final classification. The model is trained and tested across all 10 digit classes and benchmarked against a standard FFNN. Results demonstrate that the best SVM+FFNN pipeline outperforms the baseline FFNN by 4.7%.

1 Introduction

The handwritten digits dataset from scikit-learn is a classic multi-class classification task consisting of 1,797 samples and 64 features per sample (8x8 pixel images). The goal is to assign each input to one of 10 digit classes (0–9). We explore a two-stage classification approach that combines the robustness of SVMs with the flexibility of neural networks.

2 Architecture Overview

2.1 Support Vector Machine (SVM)

The first stage of the pipeline is a binary SVM that predicts whether the sample belongs to a particular class (e.g., digit '3') or not. For each target class (0 to 9), a separate SVM is trained. The SVM's output confidence score is treated as the probability of the target class.

2.2 Feedforward Neural Network (FFNN)

The FFNN handles full multi-class classification. It has:

- An input layer with 64 features (flattened pixel values)
- One hidden layer with 4 neurons and sigmoid activation

- An output layer with 10 neurons for 10 classes, also using sigmoid activation

The FFNN is trained using backpropagation and gradient descent over 10,000 epochs.

2.3 Two-Stage Inference

During inference, the model combines the SVM score for the selected class and the FFNN probabilities for the other 9 classes, forming a probability vector over all 10 classes.

3 Training and Optimization

All input features were normalized using StandardScaler. For each class (0–9), the pipeline was trained with that class as the SVM target. The FFNN was trained from scratch for each run. A baseline FFNN model was also trained independently for comparison.

4 Results

Accuracy on the test set (20% split) for each SVM+FFNN pipeline and the standard FFNN is shown below:

- **SVM+FFNN Pipeline (Target 0)**: 85.56%
- **SVM+FFNN Pipeline (Target 1)**: 83.06%
- **SVM+FFNN Pipeline (Target 2)**: 87.22%
- **SVM+FFNN Pipeline (Target 3)**: 88.33%
- **SVM+FFNN Pipeline (Target 4)**: 84.17%
- **SVM+FFNN Pipeline (Target 5)**: 85.56%
- **SVM+FFNN Pipeline (Target 6)**: 84.44%
- **SVM+FFNN Pipeline (Target 7)**: 83.61%
- **SVM+FFNN Pipeline (Target 8)**: 81.39%
- **SVM+FFNN Pipeline (Target 9)**: 86.94%
- **Standard FFNN**: 83.61%

The best SVM+FFNN pipeline (target 3) achieved 88.33% accuracy, outperforming the standard FFNN by 4.72%.

5 Discussion

The two-stage SVM+FFNN model outperformed the standalone FFNN in 9 out of 10 configurations, demonstrating its robustness and flexibility in class-specific refinement. The FFNN alone showed consistent performance but lacked class-specific handling. This hybrid pipeline design is promising for noisy or imbalanced datasets.

6 Conclusion

We presented a modular and scalable classification architecture combining SVM and FFNN to solve a 10-class digit recognition task. The best hybrid pipeline showed a measurable improvement over a standard FFNN. This architecture is easily generalizable to other datasets and adaptable to changing class distributions.

7 References

- Scikit-learn documentation: `https://scikit-learn.org/stable/`
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.