Report on Multi-Stage Classification Model

Sarthak Thakare

May 4,2025

Abstract

This report discusses the design and evaluation of a neural network architecture used for classifying the Iris dataset. The model consists of a two-stage classification pipeline combining a Support Vector Machine (SVM) classifier followed by a Feedforward Neural Network (FFNN) for further classification. Evaluation results in terms of accuracy for different configurations are also presented.

1 Introduction

The task of classifying the Iris dataset is a well-known problem in machine learning and pattern recognition. The Iris dataset consists of 150 samples, each with 4 features (sepal length, sepal width, petal length, and petal width) and a target variable with 3 distinct species. The objective of this report is to build a classification model using a two-stage pipeline consisting of an SVM classifier followed by an FFNN. The model's performance is evaluated using accuracy as the primary metric.

2 Network Architecture

2.1 Support Vector Machine (SVM)

The first stage of the classification pipeline is the Support Vector Machine (SVM). The SVM is used to classify one target class (e.g., species) against the rest. In this case, the model trains three separate SVM classifiers, each targeting one of the three Iris species. The SVM classifier uses a decision function to calculate the likelihood of each data point belonging to the target class.

The SVM model works by finding the hyperplane that best separates the data of one class from the others. This hyperplane maximizes the margin between the two classes.

2.2 Feedforward Neural Network (FFNN)

After the SVM classifies the target class, a Feedforward Neural Network (FFNN) is used to classify the remaining two classes. The FFNN takes the output probabilities from the SVM as inputs and performs further classification to predict which of the other two classes the data point belongs to.

The FFNN consists of:

- An input layer with 4 features (corresponding to the features of the Iris dataset).
- One hidden layer with 4 neurons, using the sigmoid activation function.
- An output layer with 3 neurons, using the sigmoid activation function, to classify the data point into one of three classes.

The architecture of the FFNN is trained using backpropagation, where the weights of the network are updated based on the gradient of the loss function.

2.3 Model Training and Optimization

The model is trained using the following steps:

- The training data is first passed through the SVM to classify the target class.
- The FFNN is then trained using the output of the SVM as input.
- Backpropagation is used to adjust the weights of the FFNN to minimize the loss.
- The learning rate is set to 0.01, and the model is trained for 10,000 epochs.

3 Evaluation and Results

3.1 Training and Testing

The Iris dataset was split into training and testing sets with an 80-20 ratio. The training data was standardized using the `StandardScaler` to improve convergence during training.

Three separate pipelines were trained, each targeting a different species. The accuracy for each of these pipelines was evaluated and recorded. Additionally, the FFNN was also trained independently and its accuracy was compared to that of the SVM+FFNN pipelines.

3.2 Accuracy Results

The accuracy of each pipeline was measured as the percentage of correctly classified data points in the testing set. The following results were obtained:

- **SVM+FFNN Pipeline (Target 0)**: 100.00% accuracy
- **SVM+FFNN Pipeline (Target 1)**: 93.33% accuracy

- **SVM+FFNN Pipeline (Target 2)**: 86.67% accuracy
- **Standard NN**: 100.00% accuracy

The accuracy for each pipeline was calculated by comparing the predicted labels with the true labels in the testing set. The highest accuracy was achieved by the Standard NN classifier, but the SVM+FFNN pipeline also performed very well.

4 Discussion

The results show that both the SVM+FFNN pipeline and the Standard NN classifier performed similarly in terms of accuracy. However, the SVM+FFNN pipeline has the advantage of using a two-stage classification process, which may allow it to perform better for specific types of classification problems. In this case, the FFNN complements the SVM by further refining the classification of the remaining classes.

While the Standard NN performed slightly better in this specific case, the SVM+FFNN pipeline could still be more beneficial for more complex datasets with imbalanced classes or noisy data.

5 Conclusion

In this report, we implemented a classification pipeline that combines an SVM and an FFNN for classifying the Iris dataset. We demonstrated the network architecture and evaluated its performance in terms of accuracy. The results showed that the proposed model achieved high classification accuracy, with the Standard NN classifier achieving the best performance overall. However, the SVM+FFNN pipeline also showed strong results, highlighting its potential in multi-stage classification problems.

Future work could include tuning hyperparameters, exploring different activation functions, or applying the model to other datasets to assess its generalization capabilities.

6 References

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