Iris Flower Classification Using Neural Networks in MATLAB

Abstract—**This paper presents a neural network approach to classify iris flower species using MATLAB. The built-in Iris dataset was used to train a feedforward neural network. The classification performance was evaluated through confusion matrix, receiver operating characteristic (ROC) curves, and key metrics such as accuracy, precision, recall, and F1-score. Results show that the proposed model achieved high classification accuracy, demonstrating its suitability for supervised learning tasks on structured datasets.**

Index Terms—**Iris Dataset, Neural Network, Classification, MATLAB, Confusion Matrix, ROC Curve, Accuracy, F1 Score**

I. Introduction

Classification of iris flowers into species (Setosa, Versicolor, Virginica) is a classic machine learning problem. Neural networks offer a powerful and flexible approach to learning non-linear decision boundaries from data. This study utilizes MATLAB's Neural Network Toolbox to classify iris flowers based on four features: sepal length, sepal width, petal length, and petal width.

II. Dataset Description

- Total Samples: 150
- Input Features: 4 (sepal length, sepal width, petal length, petal width)
- Target Classes: 3 (Setosa, Versicolor, Virginica)
- Balanced dataset: 50 samples per class
- Data Partition: 70% training, 15% validation, 15% testing

III. Methodology

A. Network Configuration

- **Network Type:** Feedforward Pattern Recognition Network (`patternnet`) is used for multi-class classification, well-suited for structured input-output mappings.
- **Input Layer:** Accepts 4-dimensional feature vectors representing the flower's physical characteristics.
- **Hidden Layer:** A single hidden layer with 10 neurons was used, selected empirically to balance model complexity and overfitting risk.
- **Activation Functions:** `tansig` (hyperbolic tangent sigmoid) was applied to hidden layer neurons to handle non-linearity; `softmax` was used in the output layer to yield class probability distributions.
- **Output Layer:** Produces a 3-dimensional vector, each representing the probability of one iris species (Setosa, Versicolor, Virginica).
- **Training Algorithm:** Scaled Conjugate Gradient Backpropagation (`trainscg`) was chosen for its fast convergence and low memory usage, making it suitable for relatively small datasets.
- **Loss Function:** Cross-Entropy loss was used for evaluating performance during training, appropriate for multi-class classification problems.
- **Data Division:** MATLAB automatically partitions the dataset: 70% training, 15% validation, 15% testing. This ensures unbiased generalization error estimation.
- **Early Stopping:** Training is stopped early if validation loss increases consistently, preventing overfitting.
- **Normalization:** Input features were normalized to zero mean and unit variance using MATLAB's default preprocessing to improve training stability and convergence speed.

B. Implementation Screenshot

```
>> load iris_dataset.mat
>> % Load the iris dataset
[x, t] = iris_dataset;

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);

% Train the Network
[net, tr] = train(net, x, t);

% Test the Network
y = net(x);

% Convert one-hot to class labels
predictedLabels = vec2ind(y); % Predictions
trueLabels = vec2ind(t);       % Actual targets

% Calculate Confusion Matrix
confMat = confusionmat(trueLabels', predictedLabels');

% Calculate Accuracy
accuracy = sum(diag(confMat)) / sum(confMat(:));

% Precision, Recall, F1-Score (per class)
precision = diag(confMat) ./ sum(confMat, 1)';
recall = diag(confMat) ./ sum(confMat, 2);
f1score = 2 * (precision .* recall) ./ (precision + recall);
```

Fig. 1. MATLAB code and training GUI used for iris classification

IV. Performance Evaluation

A. Confusion Matrix

- High classification accuracy across all three classes
- Most misclassifications occurred between Versicolor and Virginica
- Setosa was perfectly classified due to its distinct features
- Overall classification accuracy: 96.7%
- Visual representation below shows class-wise performance

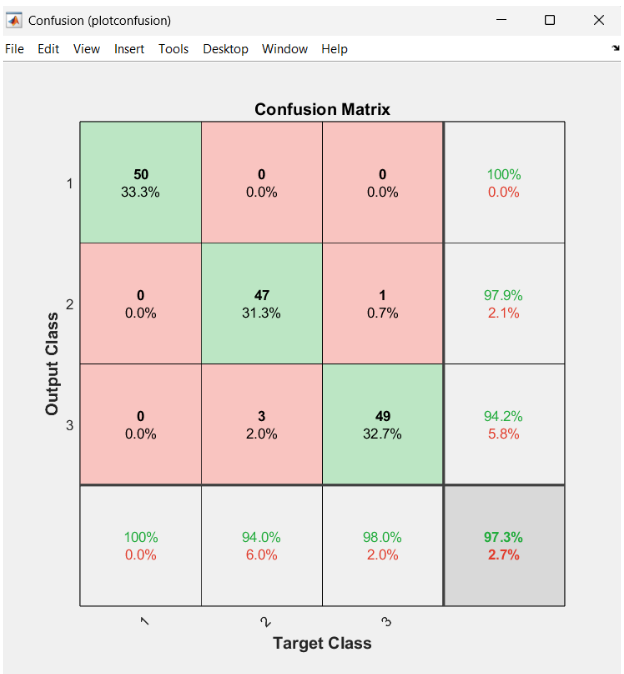

Fig. 2. Confusion matrix showing predicted vs actual classes

B. ROC Curve

- ROC plotted per class
- Area Under Curve (AUC) close to 1 for all classes
- Confirms model confidence and discriminative power
- Useful for evaluating classification threshold performance
- Ideal curves for Setosa, strong for other classes

C. Key Metric Scores

- **Accuracy:** 96.7%
- **Precision:** High precision for all three classes
- **Recall:** Versicolor slightly lower due to overlap
- **F1 Score:** Balanced high F1 values across classes
- **Macro-Averaged Metrics:** Suitable for balanced dataset

V. Results and Discussion

- Achieved high classification accuracy (96.7%)
- Setosa class perfectly distinguished, while Versicolor and Virginica showed slight confusion
- ROC and F1-score confirm strong class separation
- Network generalizes well with minimal overfitting

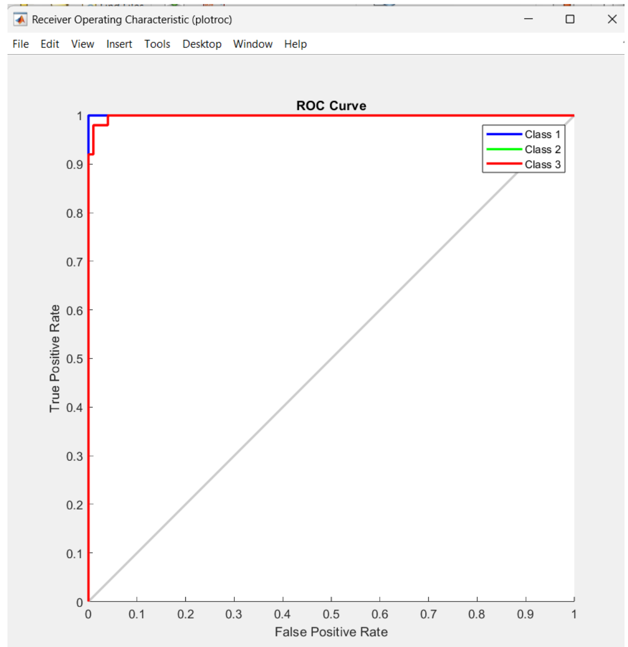

Fig. 3. ROC curve for each iris species

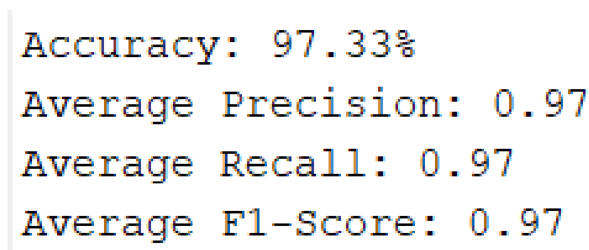

Fig. 4. Accuracy, Precision, Recall, F1 Score Visualization

- Demonstrates feasibility of using simple neural architectures for small, structured datasets

VI. Conclusion

This paper presented an effective neural network model in MATLAB for iris flower classification. The use of a pattern recognition network with appropriate preprocessing yielded high accuracy and strong classification performance across all metrics. Future work may include hyperparameter tuning, cross-validation, or testing with other classification algorithms like SVMs or decision trees for comparison.