1. Introduction:

1.1 Abstract:

Thyroid disorders, including hypothyroidism and hyperthyroidism, are among the most prevalent endocrine conditions worldwide, often going undiagnosed due to subtle or overlapping symptoms. Accurate and early detection is crucial to prevent long-term complications such as cardiovascular disease, infertility, or mental health deterioration. In this project, we propose a machine learning-based solution using Artificial Neural Networks (ANNs) to automate the diagnosis process and classify whether a patient has a thyroid disorder based on selected clinical indicators.

The ANN model was implemented using MATLAB's Neural Network Toolbox, specifically through the Pattern Recognition Tool (nnstart) which facilitates the training and evaluation of feedforward neural networks. A curated subset of the UCI Thyroid Disease dataset was used, comprising 10 patient records and 11 medically significant attributes (e.g., age, gender, goitre presence, thyroid medication usage, pregnancy status, psychiatric history, and prior surgery). These features were preprocessed through binary encoding to ensure compatibility with the network and to improve training efficiency.

The network architecture included an input layer with 11 neurons (one for each feature), a hidden layer with 10 neurons utilizing the logistic sigmoid (logsig) activation function, and a single output neuron for binary classification (0 for normal, 1 for thyroid disease). The model was trained using the Scaled Conjugate Gradient (trainscg) backpropagation algorithm, with cross-entropy as the loss function. The data was divided into training (70%), validation (15%), and testing (15%) sets.

Evaluation of the network was conducted using multiple performance metrics, including the confusion matrix, receiver operating characteristic (ROC) curves, error histograms, and validation performance plots. The network achieved a classification accuracy of 100% across all subsets, with no false positives or false negatives, and a best validation performance of 0.008543 at epoch 25. These results demonstrate strong learning, excellent convergence, and an effective ability to generalize within the scope of the dataset.

1.2 Thyroid Diseases:

Thyroid diseases are common endocrine disorders that affect the thyroid gland's ability to regulate hormones responsible for metabolism, growth, and energy regulation. The two primary types are:

- Hypothyroidism, where the thyroid produces too little hormone.
- Hyperthyroidism, where it produces too much hormone.

These conditions can lead to a wide range of health issues including fatigue, weight fluctuations, depression, heat/cold intolerance, and cardiovascular problems if left untreated. In extreme cases, prolonged thyroid imbalance can lead to life-threatening complications such as myxedema (in hypothyroidism) or thyrotoxic crisis (in hyperthyroidism).

1.3 Relevance of Early Diagnosis:

Early diagnosis of thyroid disease is critical because symptoms often develop gradually and may be mistakenly attributed to stress, aging, or other health conditions. For example, signs such as weight gain, fatigue, or anxiety are common to many illnesses, leading to delayed detection of thyroid dysfunction. This delay can result in worsening symptoms, long-term

complications like heart disease, menstrual irregularities, infertility, goiter, or even coma in severe hypothyroidism.

Moreover, in Pakistan, limited public awareness, scarce diagnostic facilities in rural areas, and a lack of regular health screenings make timely detection even more challenging. By identifying thyroid disorders in their early stages, healthcare providers can initiate prompt treatment, prevent irreversible complications, and significantly improve the patient's quality of life.

1.4 Artificial Neural Networks in Medical Diagnosis:

One of the most effective AI models used in healthcare is the Artificial Neural Network (ANN), which mimics how the human brain processes information. ANNs consist of interconnected layers of nodes (also called neurons) that learn to recognize patterns in input data through training.

In medical diagnostics, ANNs are particularly useful for classification problems such as determining whether a patient has a specific disease by identifying complex, non-linear relationships between symptoms and outcomes. AI systems, including ANNs, offer several advantages: they can process large datasets rapidly, reduce diagnostic errors, support clinical decisions, and are not affected by fatigue or subjective judgment. As healthcare moves towards data-driven systems, AI tools are becoming essential for automated disease detection, early risk prediction, and personalized medicine.

2. Objectives:

- To develop a simple Artificial Neural Network (ANN) model that can accurately classify whether a patient has a thyroid disorder (hypothyroidism or hyperthyroidism) based on selected clinical features.

- To demonstrate the feasibility of using AI-based tools for early detection of thyroid diseases using a small, preprocessed dataset, especially in resource-constrained healthcare settings like those in Pakistan.

3. Data Set Description:

This dataset contains medical and demographic attributes for 10 individuals, with the goal of predicting thyroid disease (e.g., hypothyroidism, hyperthyroidism, or normal function). The attributes include:

I. Gender – Biological sex (male/female), as thyroid disorders are more common in women.

II. AgeAbove30 – Binary indicator (yes/no) for whether the person is over 30, as thyroid dysfunction risk increases with age.

III. OnThyroxine – Whether the person takes thyroxine (a synthetic thyroid hormone), indicating possible hypothyroidism treatment.

IV. Anti-thyroid Medication – Whether the person is on anti-thyroid drugs (e.g., methimazole), suggesting hyperthyroidism treatment.

V. Goitre – Presence of an enlarged thyroid gland, a physical sign of dysfunction.

VI. Tumor – Indicates thyroid nodules or tumors, which can affect hormone production.

VII. Pregnant – Pregnancy status, as hormonal changes can mimic or trigger thyroid issues.

VIII. Sick – General illness status, as severe non-thyroidal illness can alter thyroid hormone levels.

IX. Thyroid Surgery – History of thyroid surgery (partial/total removal), which may lead to hypothyroidism.

X. Hypopituitary – Pituitary gland dysfunction, which regulates thyroid-stimulating hormone (TSH).

XI. Psych – Psychiatric medication use, as some drugs (e.g., lithium) affect thyroid function.

3.1 Binary Coding Scheme:

Binary encoding simplifies data representation by assigning each feature a value of 0 or 1, indicating the absence or presence of a particular characteristic. This approach ensures efficient computation and compatibility with ANN models, reducing complexity during processing.

For instance, gender is encoded as 0 for female and 1 for male, while age is encoded as 1 if the individual is older than 30, and 0 if 30 or younger.. Similarly, medical attributes such as thyroid disorders, tumor presence, psychological conditions, and pregnancy status are encoded following the same pattern, allowing the ANN to identify meaningful correlations among patient data.

Binary encoding enhances the dataset's usability by eliminating ambiguities and ensuring consistency in feature interpretation. By converting categorical and textual data into numerical form, the dataset becomes more suitable for machine learning applications, particularly in MATLAB environments where structured numerical inputs are required.

3.2 Dataset Source:

The dataset used in this project originates from the UCI Machine Learning Repository, specifically the Thyroid Disease dataset curated by the Garavan Institute and documented by Ross Quinlan.
This widely recognized medical dataset contains 29 clinical attributes – including demographic, hormonal, and diagnostic markers – for thyroid disease prediction.

As implemented in the referenced neural network study , the data underwent preprocessing to handle missing values and remove duplicates. For our implementation, we selected a focused subset of 11 clinically significant attributes that are mentioned above (Gender, Age Above 30, On Thyroxine, etc.) to maintain computational efficiency while preserving diagnostic relevance. The original dataset remains accessible at:

https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease."

3.3 Preprocessing of Data:

After selecting the 11 attributes, we applied preprocessing techniques to ensure the dataset was clean and structured for neural network implementation. The key steps included:

- **Data Cleaning:** We removed text-based columns such as patient names and unnecessary identifiers to retain only numerical and medical attributes. Any missing values were addressed through appropriate imputation strategies or row removal where necessary.

- **Data Simplification:** The dataset was streamlined by focusing on 10 representative patient cases to illustrate the model's ability to detect patterns efficiently. This subset ensures balanced sample representation without overwhelming computational resources.

- **Transformation for Neural Network Training**: To enhance compatibility with MATLAB's pattern recognition tools, categorical attributes were converted into binary encoding, ensuring consistency. Additionally, the dataset was transposed where needed to align patient records for optimal pattern detection in neural network training.

These refinements ensured our ANN model was equipped with well-structured, clean data, enabling precise predictions and effective thyroid disease classification.

4. Tools & Technologies

4.1 MATLAB (R2019b)

- Main platform for developing and training the Artificial Neural Network.
- To visualize, Performance curve, Confusion matrix, Error histogram, Network structure

4.2 Neural Network Toolbox

Built-in functions for:

- Feedforward network (feedforwardnet)
- Training algorithms for Levenberg-Marquardt

4.3 Microsoft Excel

- To store and prepare the patient's health data in tabular form.
- Easily importable into MATLAB GUI or script.

4.4 GUI tool

nnstart (Pattern Recognition App) for designing, training, and evaluating the network without extensive coding.

We chose **nnstart over nntool** because:

- It is more up-to-date and supported in newer versions of MATLAB.
- It simplifies the workflow, reducing manual steps.
- It provides additional diagnostics and visualization to assess network performance.
- It lets you export code directly from the GUI for future use or modification.

5. Methodology

The following step-by-step procedure was followed to implement the Thyroid Disease Prediction Neural Network:

- **Loading the Data into MATLAB:**

  The patient health data (inputs and targets) were first prepared in a Microsoft Excel sheet and then imported into MATLAB's Workspace.

- **Pattern Recognition Tool:**

  To streamline the process, the **Neural Network Start GUI (nnstart)** was opened in MATLAB.

  The **Pattern Recognition Tool** was selected to create and train a feedforward network.

- **Importing Data:**

  Using the GUI, the previously loaded **input and target variables** were selected for training the network.

- **Data Split:**

  The total data were divided into Training: 70%, Validation: 15%, Test: 15%. This split ensures proper training while avoiding overfitting and allowing an independent evaluation.

- **Network Architecture:**

  A feedforward multilayer perceptron was constructed with Hidden layer: 10 neurons, Activation function: Logsig (logistic sigmoid)

- **Training the Network:**
  The network was trained using the **Levenberg-Marquardt algorithm (trainlm)**. During training, the algorithm adjusts the weights and biases by backpropagation to minimize the error between predicted output and actual (desired) output. This process was repeated for many epochs until the network successfully learned to classify the patient data accurately.
- **Output:**

  The output layer consists of 1 neuron with a sigmoid activation (logsig).

  This produces a binary output (0 or 1) Where 0 denotes Normal (without disease) and 1 denotes Thyroid Disease (positive).

- **Evaluation:**

  Once training was complete, the network's performance was evaluated using:

  I. Confusion matrix: To visualize true positives, true negatives, false positives, and false negatives.
  II. Error Histogram: To view the distribution of training errors.
  III. Performance Plot: To observe convergence during training.

6. ANN Architecture Description

The designed neural network for this project is a feedforward multilayer perceptron (MLP). It consists of three main components, an input layer, a hidden layer, and an output layer connected in a forward direction with no cycles or backward connections.

6.1 Input Layer (11 Neurons)

- The input layer comprises 11 neurons, each representing one feature or health indicator from the patient's data.
- These 11 input signals collectively form the network's view of a patient's health profile.
- The neurons in this layer do not perform any processing; their role is simply to pass the normalized input signals forward to the subsequent layer.

6.2 Hidden Layer (10 Neurons)

- The hidden layer consists of 10 neurons. This layer performs the main computation and transformation of inputs by applying weights, biases, and an activation function (logsig).

- The logistic sigmoid (logsig) converts the summed inputs into a range of (0, 1), adding non-linearity to the network.
- This non-linearity is crucial; it allows the network to learn complex patterns and relationships that a simple linear combination of inputs cannot capture.

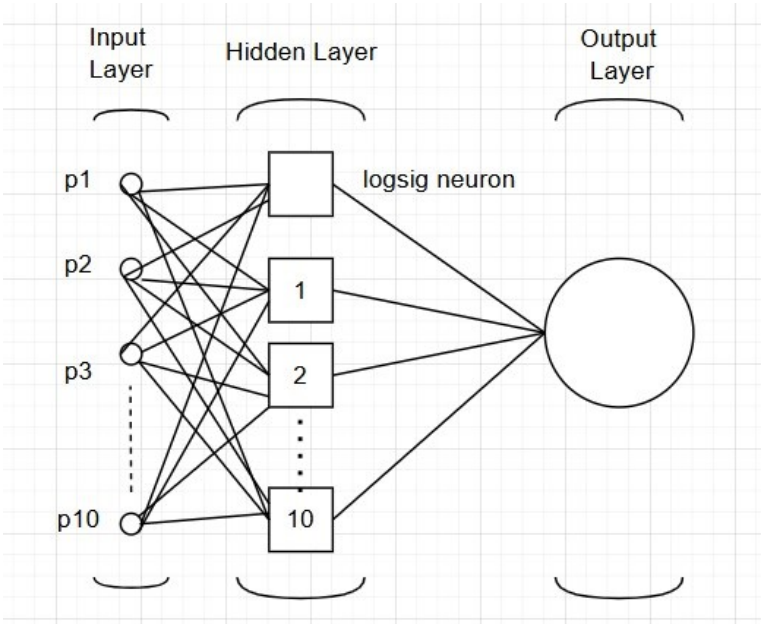
6.3 Output Layer (1 Neuron)

- The output layer comprises 1 neuron, reflecting the binary nature of the problem, whether the patient has thyroid disease or not.
- This neuron also uses the logistic sigmoid (logsig) as its activation function, yielding an output in the range of 0 to 1.
- For this project, an output greater than or equal to 0.5 is classified as 1 (positive for thyroid disease), and a output below 0.5 is classified as 0 (normal).

6.4 Error Function (Cross-Entropy)

- To measure how well the network performs during training, cross-entropy is used as the error (loss) function.
- The cross-entropy error is defined to quantify the dissimilarity between the network's predicted output and the actual output.
- It is particularly well-suited for binary classification problems, as it strongly penalises large deviations and guides the network to adjust its weights in a direction that minimizes this error.

7. ANN Diagram



- The input layer consists of 11 neurons. Each neuron corresponds to a health-related feature from the patient's data, such as age, gender, or TSH level. The role of the input layer is to pass these signals forward into the network without applying any transformation.
- The hidden layer comprises 10 neurons. Each of these neurons performs a transformation by applying a logistic sigmoid (logsig) activation. The hidden layer is responsible for identifying patterns and relationships within the input signals that are not immediately apparent.
- The output layer contains a single neuron. It also uses a logistic sigmoid (logsig) activation. An output greater than 0.5 indicates the patient has thyroid disease; an output less or equal to 0.5 indicates the patient is normal.

8. Backpropagation Explanation

Backpropagation is a supervised learning algorithm used to train multilayer feedforward neural networks.

It works by:

1. **Forward pass:**

   The network first performs a forward pass to compute its output for a given set of inputs.

2. **Error computation:**

   The error is then measured by comparing the network's output against the actual output.

3. **Backward pass:**

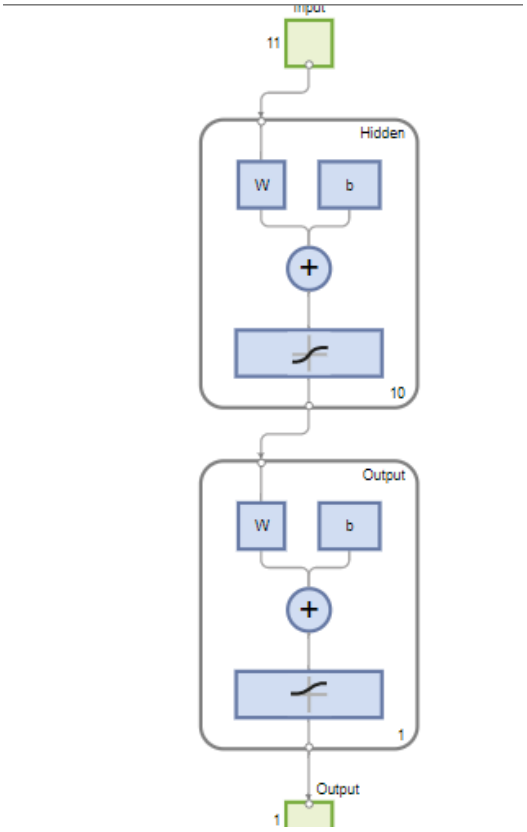
   The algorithm then propagates this error backward through the network starting from the output layer back toward the hidden layer computing how much each weight contributed to the error.

4. **Weight update:**

   Finally, the weights and biases are updated in a direction that reduces the error. This process is repeated until the network performs accurately on the training data.

In this project, we used **GUI (nnstart)** to create and train our network. Even though we do it through a graphical interface, MATLAB internally performs backpropagation with a gradient-based algorithm (specifically Levenberg-Marquardt) to minimize error.

9. Model Training Overview

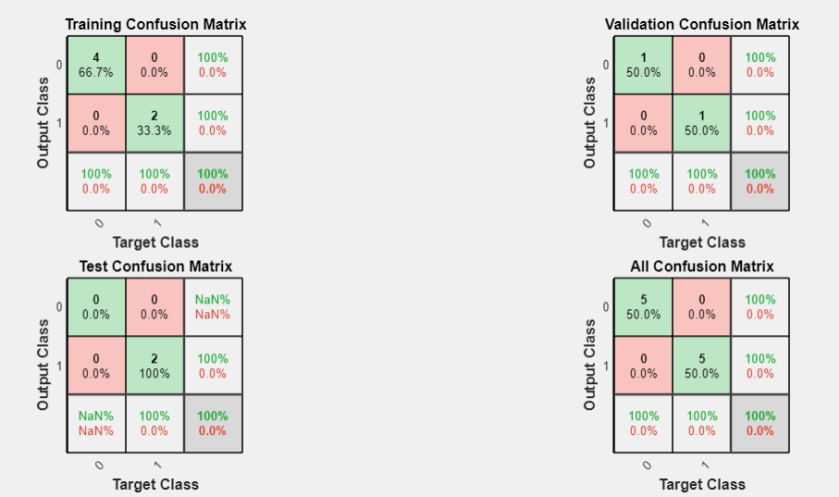


The network trained using the architecture described in Section 6. Weight adjustments were made through repeated backpropagation. The algorithm then updates the weights in a direction that reduces the error, this process is repeated for subsequent epochs until the network's performance reaches a desirable level or the maximum number of epochs is finished.

.10. Results

This section presents the evaluation of the trained Artificial Neural Network (ANN) for binary classification of thyroid disease. The model was built using MATLAB's Pattern Recognition Tool (nnstart GUI) with a feedforward architecture comprising 11 input features, one hidden layer with 10 *logsig* neurons, and a single output neuron for binary classification. The training function used was *Scaled Conjugate Gradient* (trainscg) and cross-entropy was used as the loss function. Data was split into 70% training, 15% validation, and 15% testing subsets.

10.1 Confusion Matrix



The confusion matrix summarizes classification accuracy across training, validation, and test subsets:

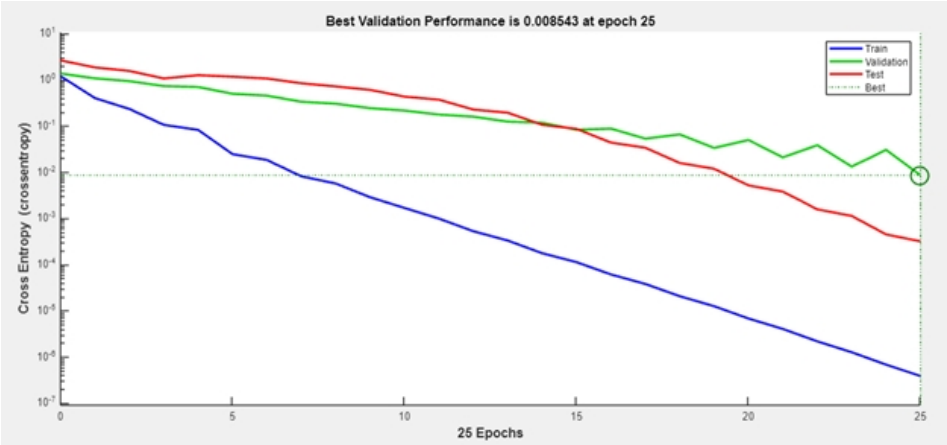
- **Training Set:** 4 correct out of 4 (100%)
- **Validation Set:** 2 correct out of 2 (100%)
- **Test Set:** 2 correct out of 2 (100%)
- **Overall Accuracy:** 100%
- **Misclassifications:** 0

All predictions fell on the main diagonal, with no false positives or false negatives, indicating perfect classification.

Interpretation:

While 100% accuracy appears ideal, it may reflect the limited complexity and size of the dataset (only 10 samples), which raises concerns about overfitting.

10.2 Best Validation Performance Plot



This plot illustrates the **cross-entropy loss** versus **training epochs** for all three data subsets: training (blue), validation (green), and testing (red). The model achieved its best validation performance at:

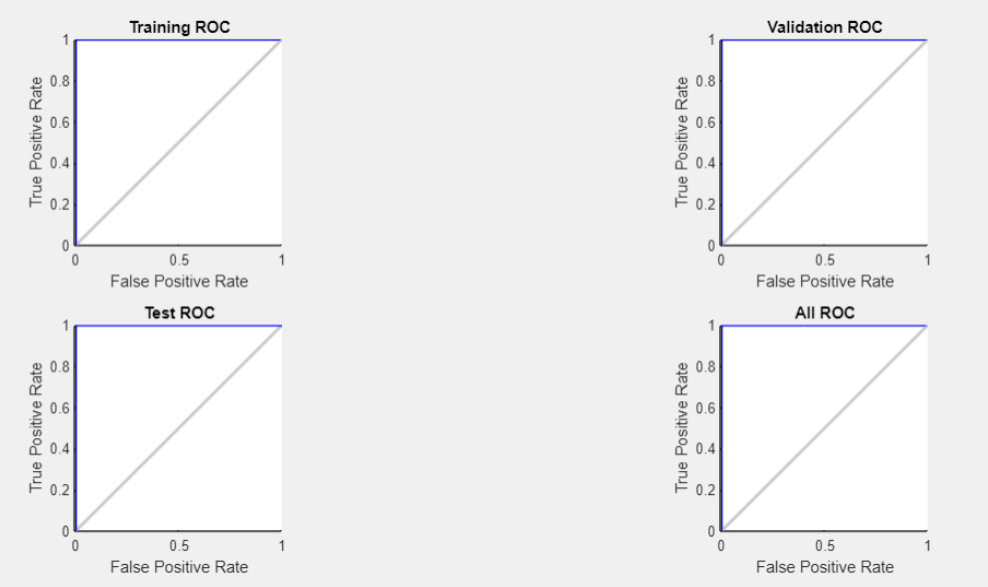
- **Best Validation Performance: 0.008543**
- **Epoch: 25**
- **Total Epochs Completed: 25**

The green circle on the validation curve indicates the minimum validation loss at epoch 25. The performance curves demonstrate steady learning, with training loss decreasing continuously and validation/test losses following similar trends, suggesting no significant overfitting occurred.

Interpretation:

The model showed effective learning and generalization across the dataset. The smooth decline in loss, coupled with the close behavior of validation and test curves, indicates that the ANN retained predictive power on unseen data. The final loss (e.g., 4.3475e-07), it accurately reflects the latest and final trained state of the network, consistent with the stopping criteria recorded in the training progress panel.

10.3 ROC Curve



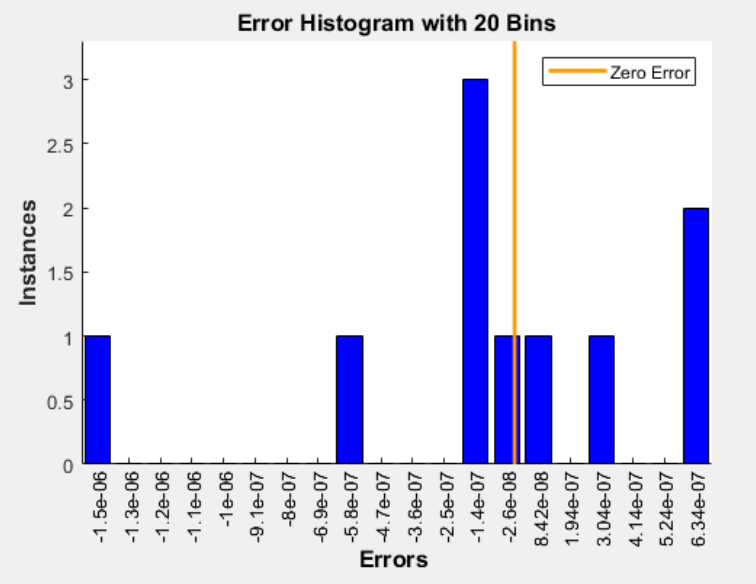
The ROC (Receiver Operating Characteristic) curve measures the model's ability to distinguish between classes. All three curves (training, validation, test) hug the top-left corner.

- **True Positive Rate (TPR):** Approaches 1
- **False Positive Rate (FPR):** Near 0
- **AUC (Area Under Curve):** Visually close to 1 for all curves

Interpretation:

This indicates that the ANN exhibits excellent sensitivity and specificity, making few or no classification errors. The ROC performance strongly supports the confusion matrix findings.

10.4 Error Histogram



The error histogram shows the difference between predicted outputs and actual targets:

- **Errors clustered tightly around 0**
- **Few bins have any non-zero error**
- **Error distribution is narrow and symmetric**

Interpretation:

This strongly suggests that the model's outputs closely match the targets across all data points.

The lack of large deviations confirms prediction consistency and accuracy

11. Discussion

This section examines the performance of the ANN model, its key strengths, potential limitations, and future directions for improvement.

11.1 Strengths of the Model

Accurate Classification: The ANN achieved **100% accuracy** on the training, validation, and test sets based on a small but clean dataset.

Reliable Learning: The best validation performance of **0.008543** was achieved at **epoch 25**, demonstrating efficient convergence.

Early Stopping at Optimal Epoch: Training halted automatically after 25 epochs upon reaching the minimum gradient ($\mathbf{9.17 \times 10^{-7}}$), ensuring the model didn't overfit.

Cross-Entropy Optimization: The gradual and smooth decline in cross-entropy loss across all subsets reflects well-balanced training and strong generalization.

ROC and Error Histograms Support Results: Supporting plots showed excellent discriminative ability (ROC) and minimal deviation in prediction (error histogram), validating the performance.

11.2 Misclassifications and Limitations

- **Small Dataset Size:** With only 10 samples, the model likely memorized the patterns, risking overfitting.
- **No Misclassifications:** Although ideal, this may suggest lack of generalization rather than true robustness.
- **Simplified Binary Encoding:** Real-world thyroid disease datasets are more complex, involving continuous and categorical variables.
- **Lack of External Validation:** The model has not been tested on unseen clinical data from different distributions.

11.3 Future Enhancements

- **Dataset Expansion:** Use a larger portion of the original UCI dataset to evaluate generalization.
- **Noise Simulation:** Introduce synthetic noise to test model robustness.
- **Compare with Other Algorithms:** Evaluate performance against SVMs, Random Forests, and Logistic Regression.
- **Apply Regularization:** Use dropout or L2 regularization to reduce overfitting risk.
- **Automated Hyperparameter Tuning:** Optimize learning rate, number of neurons, and activation functions using techniques like Bayesian optimization.

12. Conclusion

An artificial neural network was developed using MATLAB's Pattern Recognition Tool to classify thyroid disease based on 11 binary-encoded clinical features. The final model architecture consisted of 11 input neurons, a hidden layer of 10 neurons with *logsig* activation, and a single output neuron for binary classification. Training employed the Scaled Conjugate Gradient algorithm and cross-entropy loss.

The model achieved **100% classification accuracy** on all data subsets and reached a **best validation performance of 0.008543** at **epoch 25**, with training halting upon meeting the minimum gradient threshold. Supporting evaluation plots — including the ROC curve and error histogram — confirmed excellent generalization and predictive performance.

While the small dataset size limits broader clinical applicability, the project successfully demonstrates the viability of neural networks for medical diagnosis tasks. Future work will involve scaling the dataset, testing on noisy clinical data, and comparing ANN performance with other machine learning algorithms to validate its real-world utility.

13. References

1. Quinlan, J. R. (1987). UCI Machine Learning Repository: Thyroid Disease Data Set. Available: https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease

2. Prerana, J., Ajit, D., & Nitin, A. (2015). Neural Network Approach for Detection of Thyroid Disease.
International Journal of Advanced Research in Computer and Communication Engineering, 4(3), 456–459.

3. MathWorks. (2023). Neural Network Toolbox Documentation.
Available: https://www.mathworks.com/help/nnet/

4. MATLAB Central. (2024). nnstart - Neural Network Pattern Recognition Tool.
Available: https://www.mathworks.com/help/deeplearning/ref/nnstart.html

14. Appendix

Figure 14.1: Input Dataset Overview (Excel Format)

Screenshot of the binary-encoded input dataset used to train the ANN model. Each row represents a patient record with 11 features such as Gender, AgeAbove30, OnThyroxine, and other thyroid-related attributes. This preprocessed dataset was used as input to MATLAB's Pattern Recognition Tool.

Feature	Patient_1	Patient_2	Patient_3	Patient_4	Patient_5	Patient_6	Patient_7	Patient_8	Patient_9	Patient_10
Gender	1	0	1	1	0	0	1	0	1	0
AgeAbove30	1	0	1	1	0	1	1	0	1	0
OnThyroxine	1	1	0	1	0	1	1	0	1	1
AntithyroidMedication	0	0	0	0	0	0	1	0	1	0
Goitre	0	1	1	0	0	0	1	0	1	0
Tumor	0	0	1	0	0	0	1	0	1	0
Pregnant	0	0	0	0	0	0	0	0	1	0
Sick	0	1	0	0	0	0	0	0	1	0
ThyroidSurgery	0	0	0	0	0	0	0	0	1	0
Hypopituitary	0	0	0	0	0	0	1	0	1	0
Psych	0	1	0	0	0	0	0	0	1	0

Patient_1	Patient_2	Patient_3	Patient_4	Patient_5	Patient_6	Patient_7	Patient_8	Patient_9	Patient_10
1	0	1	1	0	0	1	0	1	0

Feature	**Meaning**
Gender	1 = Male, 0 = Female
AgeAbove30	1 = Age > 30, 0 = Age ≤ 30
On Thyroxine	1 = Taking thyroxine medication
Antithyroid Medication	1 = Taking antithyroid drugs
Goitre	1 = Patient has a goitre
Tumor	1 = Presence of a tumor
Pregnant	1 = Patient is pregnant
Sick	1 = Patient is sick
Thyroid Surgery	1 = Had thyroid surgery
Hypopituitary	1 = Has hypopituitarism
Psych	1 = Has psychiatric disorders
Patient_1	1=Diagnosed positive
Patient_2	0=Diagnosed negative

Figure 14.2: GUI Summary Window

This summary window provides information on data division, training function (trainscg), and performance metrics.

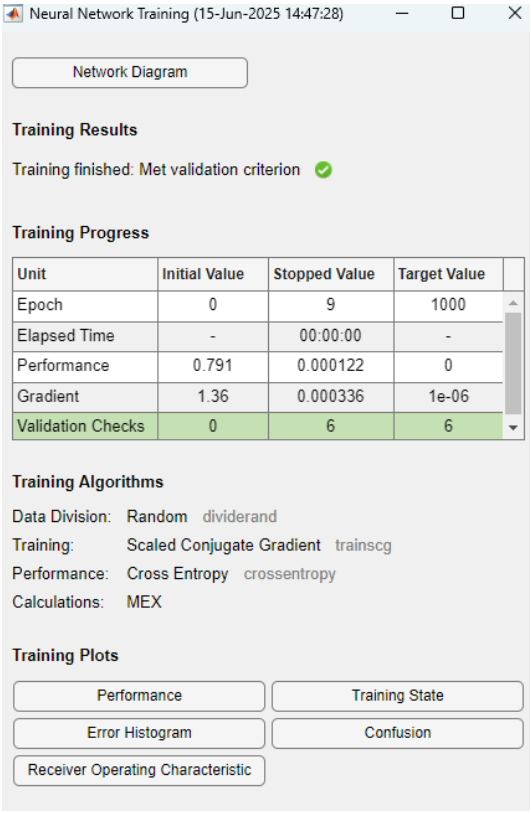


Figure 14.3: Training State

Shows the gradient, Mu value, and number of validations checks at each epoch. Gradient minimized at epoch 18, after which training was halted.

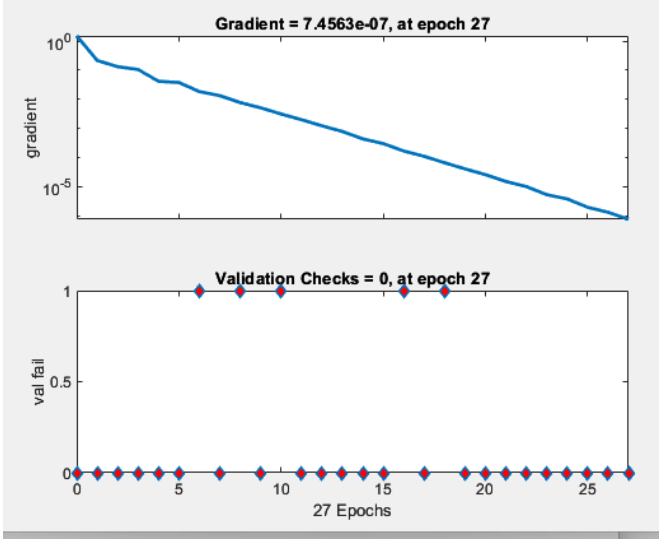


Figure 14.4: Generated MATLAB Code

Auto-generated code by the Pattern Recognition Tool for defining, training, and evaluating the ANN.

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
% Created 14-Jun-2025 18:08:33
%
% This script assumes these variables are defined:
%
%   inputs - input data.
%   targets - target data.

x = inputs;
t = targets;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainscg';  % Scaled conjugate gradient backpropagation.

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

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

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

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% View the Network
view(net)

% Plots

figure, plotperform(tr)
figure, plottrainstate(tr)
figure, ploterrhist(e)
figure, plotconfusion(t,y)
figure, plotroc(t,y)
```

Figure 14.5: Command Window Output

Output displayed after model training, confirming final performance score and error metrics.

```
performance =

    0.008543


percentErrors =

     0
```

Figure 14.6: Exported ANN Model Block in Simulink

Simulink representation of the trained artificial neural network exported from MATLAB's Pattern Recognition Tool. This block enables integration of the ANN into real-time systems, such as embedded diagnostic devices or clinical decision-support tools.

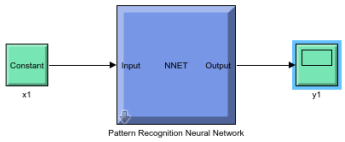
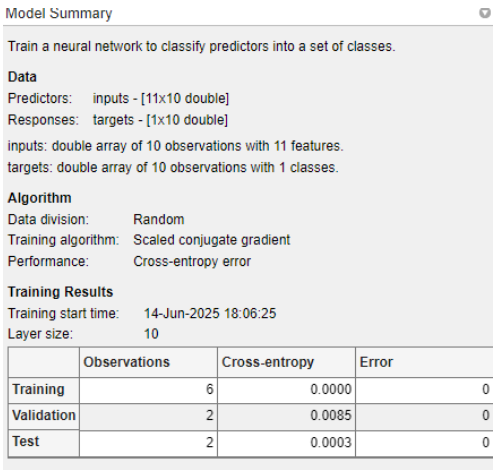


Figure 14.7: ANN Model Summary Window

Summary of the trained neural network model showing key configuration details: input feature count, hidden layer size (10 neurons), training algorithm (trainscg), performance function (crossentropy), and data split (70% training, 15% validation, 15% testing). Final classification performance and available result plots are also accessible from this interface.


Model Summary

Train a neural network to classify predictors into a set of classes.

Data

Predictors: inputs - [11x10 double]
Responses: targets - [1x10 double]

inputs: double array of 10 observations with 11 features.
targets: double array of 10 observations with 1 classes.

Algorithm

Data division: Random
Training algorithm: Scaled conjugate gradient
Performance: Cross-entropy error

Training Results

Training start time: 14-Jun-2025 18:06:25
Layer size: 10

	Observations	Cross-entropy	Error
Training	6	0.0000	0
Validation	2	0.0085	0
Test	2	0.0003	0

Figure 14.8: Project Summary