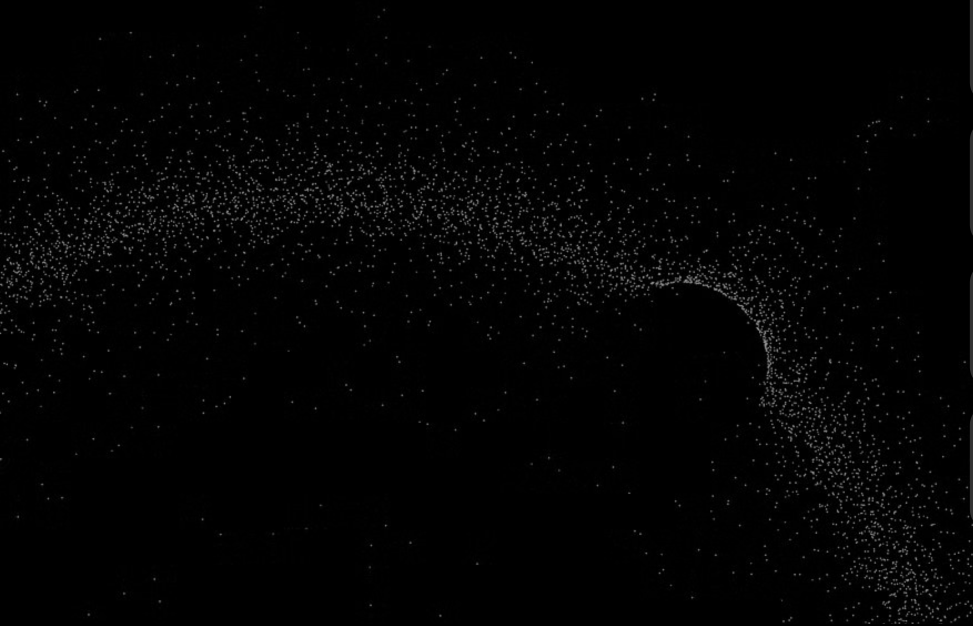
ECG Classification Using Neural Networks

Sai Yadavalli

● Machine Learning ■ May 2025

Content



1 Problem

2 Data

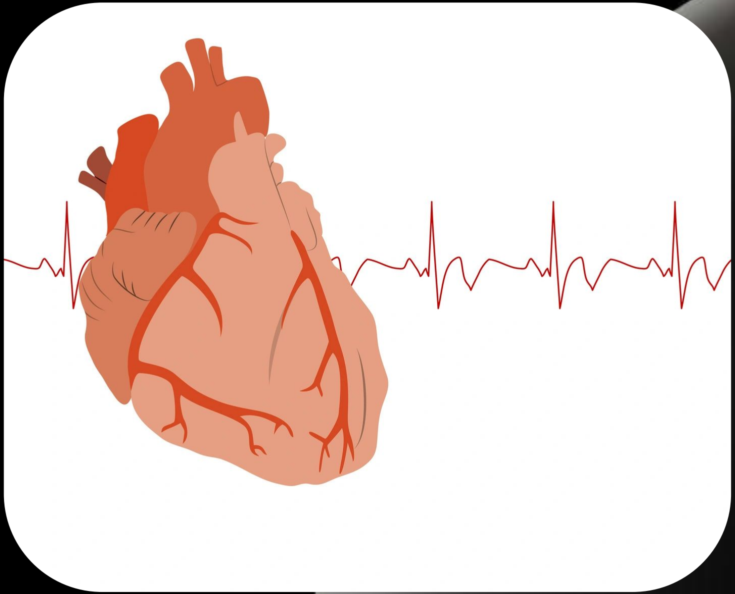
3 Algorithm

4 Training

5 Results

6 Future

Problem



- No way to automatically classify different types of heartbeats from ECG signals
- End Goal: Early, accurate detection of arrhythmias, reducing the burden on healthcare professionals by providing automated diagnostic support.

Data Description

109446

rows/samples (training + validation)

187

Features: 187 values per heartbeat (ECG signal over time)

5

Labels: Normal → 0, Supraventricular ectopic → 1, Ventricular ectopic → 2, Fusion → 3, Unknown → 4

Source

MIT-BIH

Arrhythmia Dataset

PhysioNet

Kaggle

Each sample is a 1D vector representing a fixed-length ECG waveform segment.

Link to Kaggle Data Set

Convolutional Neural Network (CNN) Model

1

Libraries

- Implemented from scratch in NumPy—no deep learning libraries used
- Pandas for data processing
- Scikit-Learn for metrics

2

Implementation

- 1D Convolution layer extracts local ECG patterns
- Max pooling reduces feature dimensionality
- Softmax layer for probability output
- ReLU activation and backpropagation—manually implemented

3

Architecture (Fully Connected)

Conv → Pool → Dense → Output

Training & Testing

Data was split into...

- 80% Training
- 20% Testing/Validation
- Not Standardized

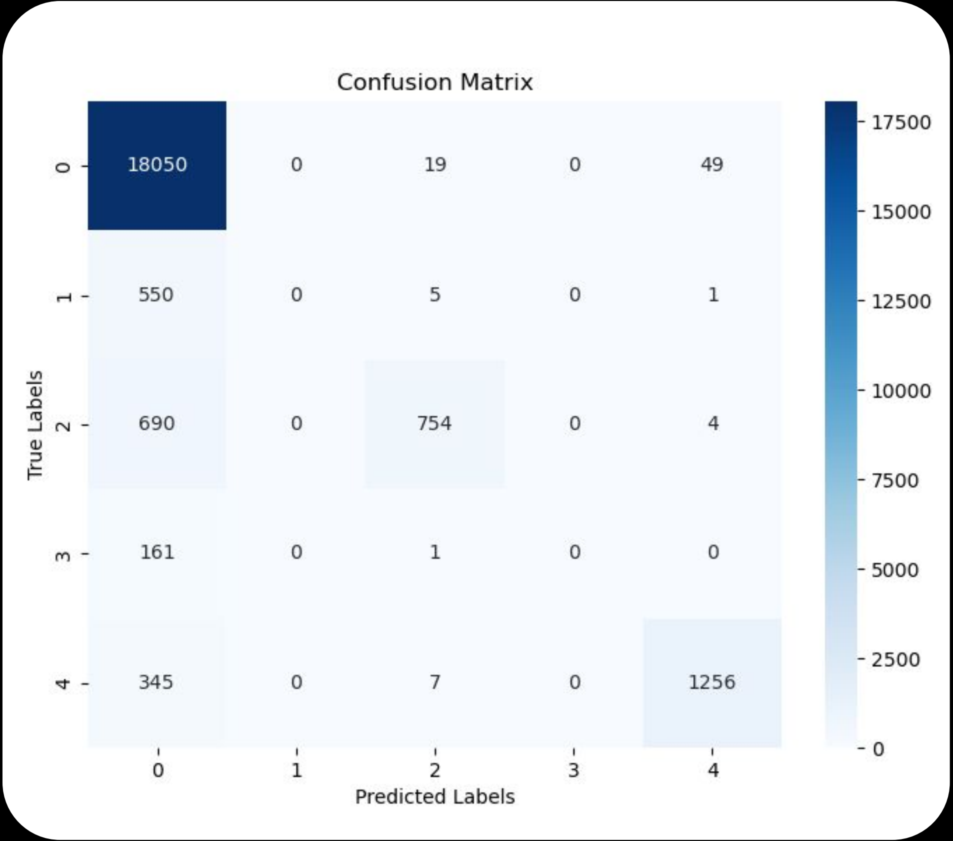
Manual implementation of...

- Forward and backward passes
- Stochastic gradient descent
- Loss: Cross-Entropy

$$H(p,q) = -\sum_{x\in\mathcal{X}} p(x)\,\log q(x).$$

Results & Evaluation

- Evaluated on test set using classification metrics:
  - Accuracy: 91.63%
  - Precision: 88.85%
  - Recall: 91.63%
  - F1 Score: 89.59%

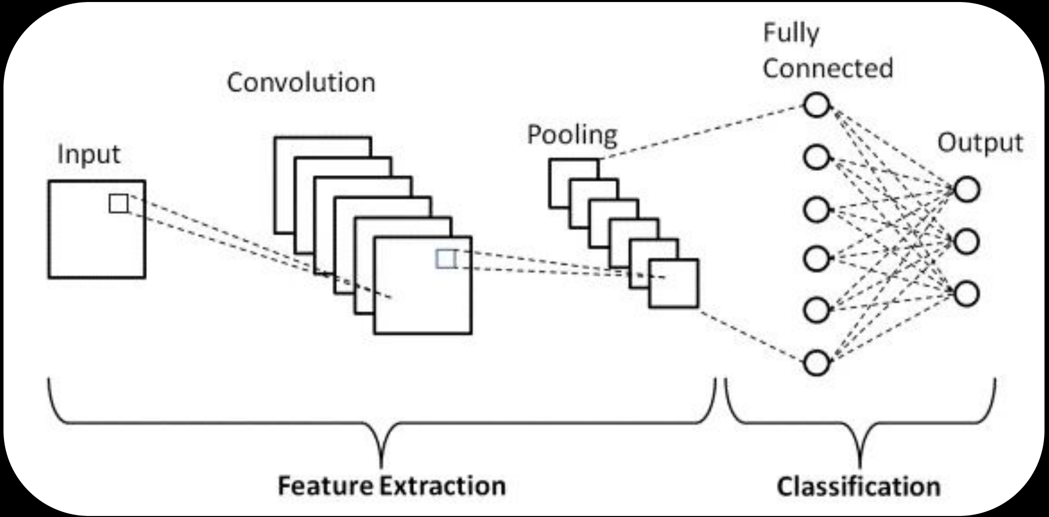


Confusion Matrix

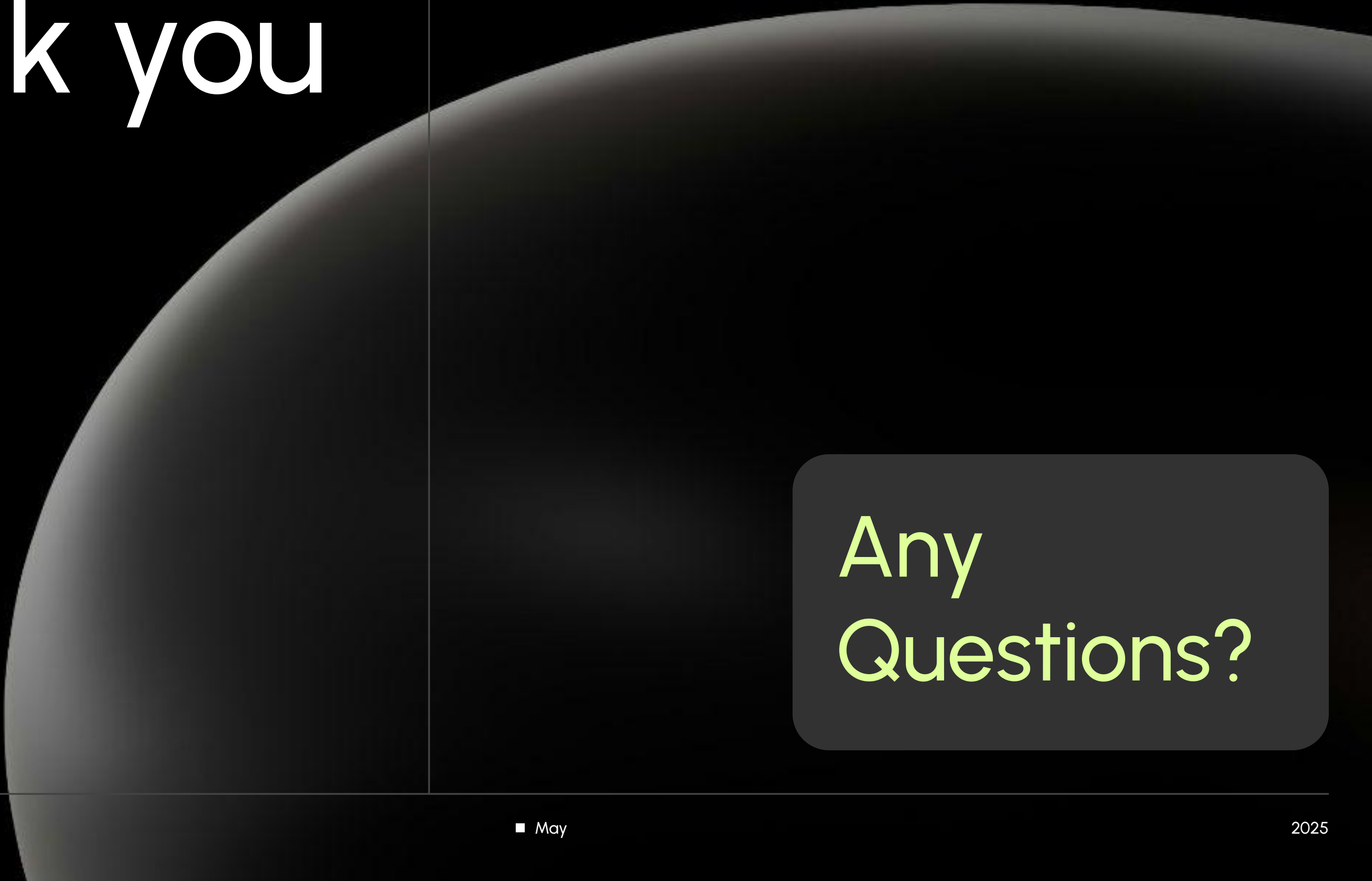
True Labels \ Predicted Labels	0	1	2	3	4
0	18050	0	19	0	49
1	550	0	5	0	1
2	690	0	754	0	4
3	161	0	1	0	0
4	345	0	7	0	1256

What Did I Learn?

- Even simple CNNs can achieve strong performance on time-series data in Arrhythmia Classification
- Implementing backpropagation, training logic, and loss function deepened understanding of ML fundamentals
- Future improvements:
    - Add more layers
    - Adjust Parameters
        - conv_filters=6,
kernel_size=5,
pool_size=2,
hidden_units=32,
learning_rate=0.01,
epochs=10,
batch_size=32
    - Try Scaling



Thank you



Any Questions?