

# **Music Genre Classifier Using Feed Forward Neural Network**

**Aashwin Vaidya**

Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya

Machine Learning

08/06/2026

## **Abstract**

This project builds a multi-layer neural network to classify musical genres using 3-second clips from the GTZAN dataset. The model uses a csv file of 3-second audio clips which contain 57 features for every audio clip. This data is normalised using Z-score normalisation and fed into the neural network as matrices. The final architecture has a two-hidden-layer structure which uses Rectified Linear Unit (ReLU) activations, which kept learning signals sharp and eliminated gradient saturation. To prevent the model from getting stuck in local traps, standard gradient descent was replaced with a native mini-batch partitioner and an adaptive Adam optimization loop. This approach successfully stabilized training, dropping cross-entropy loss to near-zero and increasing testing accuracy to 88.09%. The model's success was visually verified using an Epoch vs Categorical Cross Entropy Loss graph and confusion matrix heatmap. The project is inspired by the music genre classifier using convolutional neural networks published by Qiuqiang Kong, Xiaohui feng and Yanxiong Li.

## **1. Introduction**

The massive volume of digital music on modern streaming platforms makes manual organization completely impractical, creating an urgent need for automated media analytics. While automatic music classification is a staple of modern industrial data science, it is almost universally handled by heavy, external deep learning frameworks like PyTorch or TensorFlow.

## 2. Problem Statement

Building a high-performing audio classifier natively in Scilab presents severe data and structural challenges. First, deep learning models require a massive volume of data to generalize effectively, but standard music collections like the GTZAN dataset offer only 1,000 baseline audio tracks. Second, raw audio exists as a complex 1D time-domain signal, a long sequence of amplitude waves that standard neural networks cannot easily process. To build an optimized solution, these 30-second tracks must be preprocessed and systematically sliced into 3-second segments to expand the data pool into 10,000 distinct samples. Furthermore, the raw signal must be transformed into static multi-feature acoustic vectors so that mathematical gradients can be calculated stably. To achieve this, the project implements a multi-layer neural network through matrix linear algebra. The data matrix is first normalized using Z-score calculation to eliminate feature scale disparities. The system then routes a 57-feature input vector through a two-hidden-layer network using Rectified Linear Unit (ReLU) activations to prevent gradient saturation. To improve convergence speed and prevent the network from getting trapped in local minima, standard gradient descent is replaced with a mini-batch partitioner with an adaptive Adam optimization loop.

## 3. Basic concepts related to the topic

### 3.1 Z-score Normalization

Z-score normalization rescales the dataset so every feature has a mean of 0 and a standard deviation of 1. The expression is given as follows:

$$Z = X - \text{Mean} / \text{Standard Deviation}$$

### 3.2 Rectified Linear Activation Function

The hidden layers implement the Rectified Linear Unit (ReLU) activation function. ReLU outputs the input directly if it is positive, and clips it to zero if it is negative. The formula for the ReLU function is given as follows:

$$A = \max(0, Z)$$

Where,

$Z$  = Input Value

$A$  = Activation Function.

### 3.3 Softmax Activation Function

It turns raw network outputs (logits) into a probability distribution over the ten distinct music genres. The mathematical expression for the  $i$ -th output class is:

$$P(y = i | x) = \frac{e^{(Z_i)}}{(\sum_{j=1}^K e^{(Z_j)})}$$

Where,

$Z_i$  = logit for class  $i$

$K = 10$  for ten genres.

### 3.4 Categorical Cross Entropy Loss

To measure the mathematical error between the network's predicted genre probabilities and the actual one-hot encoded targets, the model calculates the categorical cross-entropy loss function ( $L$ ) which is given as:

$$L = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K Y_{n,k} \cdot \log(A_{n,k} + \epsilon)$$

Where,

$N$  = Batch size

$Y$  = True Binary Targets

$A$  = Softmax Output Probabilities.

### 3.5 Adam Optimization Algorithm

The Adaptive Moment Estimation (Adam) optimizer computes adaptive learning rates for each parameter by tracking both the first moment ( $m$ ) and the second moment ( $v$ ). The equations updated at each time step  $t$  are:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Because  $m_t$  and  $v_t$  are typically initialized as vectors of all zeros, they are biased toward zero. To counteract this, the algorithm applies bias-correction steps:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

:

Using these corrected moment vectors, the network weights ( $W$ ) are updated:

$$W_{t+1} = W_t - \frac{\alpha}{\sqrt{\widehat{v}_t} + \eta} \widehat{m}_t$$

Where,

$\alpha$  = Base learning rate

$g_t$  = Current gradient

$\beta_1 = 0.9$

$\beta_2 = 0.999$ .

## 4. Flowchart

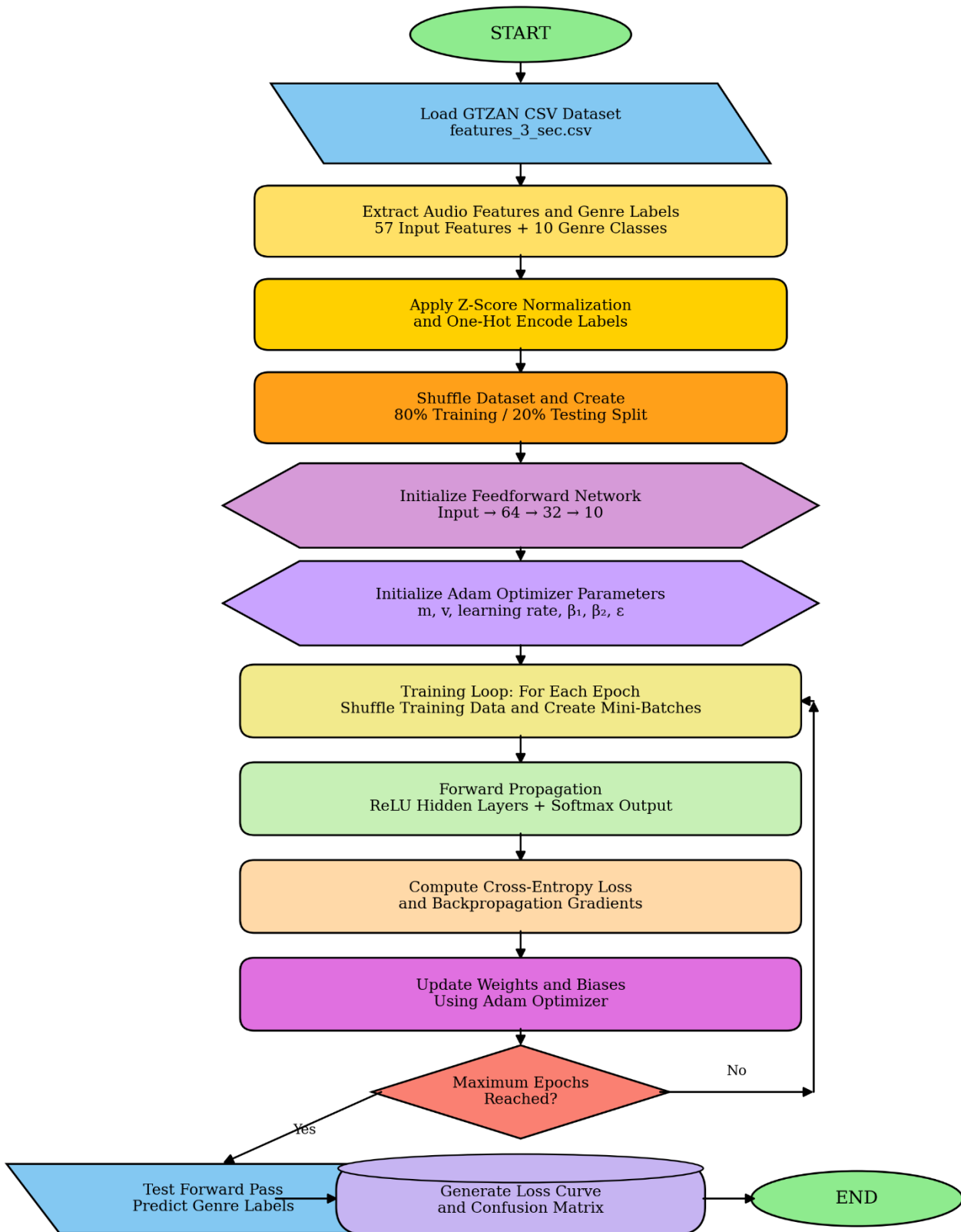


Figure 1: Flowchart of the Project

## 5. Software/Hardware used

- Operating System: Windows 11
- Toolbox: None
- Hardware: None
- Software: Scilab 2026.0.1

## 6. Procedure of execution

1. Open the file `music_classifier.sce`.
2. Go to execute → save and execute.
3. You will see the categorical entropy loss being outputted for every 50 epochs. At the end, observe the final validation accuracy.
4. Observe the Epoch vs Categorical entropy loss graph and the heat map of the confusion matrix.

## 7. Result

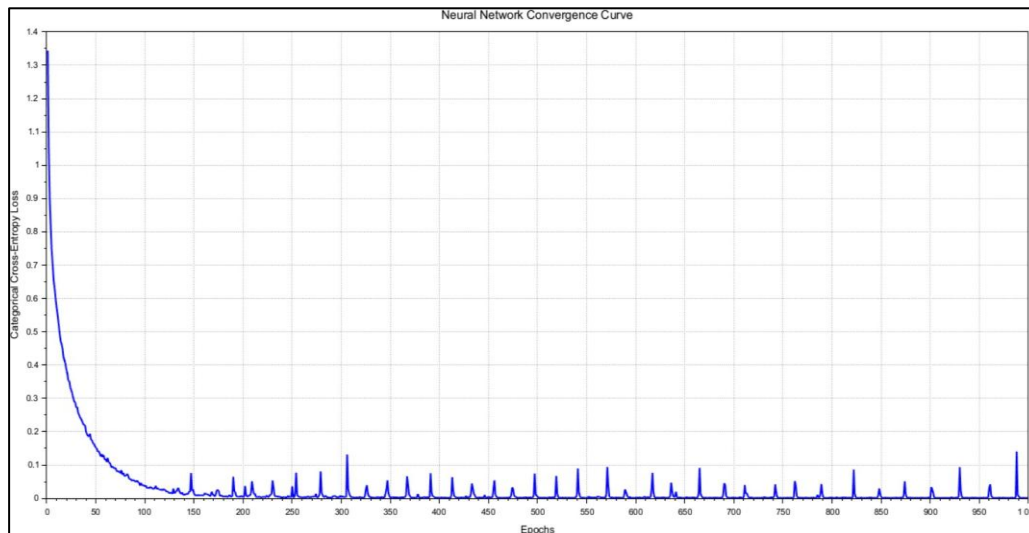


Figure 2: Neural Network Convergence Curve

## 7.1 Epoch vs Categorical Cross Entropy Loss

- At the beginning of the training phase, the loss drops steeply from an initial value of approximately 1.35 down to 0.05. This highlights the efficacy of the Adam optimizer. Because Adam calculates adaptive learning rates for each individual weight by tracking the first  $m_t$  and second  $v_t$  moments of the gradients, it rapidly corrects large initial classification errors.
- After epoch 150, the curve flattens out, asymptotically hugging the baseline near 0.01. This plateau proves that the network has reached mathematical convergence. During this extended phase, the model fine-tunes its hidden layer decision boundaries to separate closely related acoustic textures, resulting in the final accuracy of 88.09%.

## 7.2 Heat Map of the Confusion Matrix

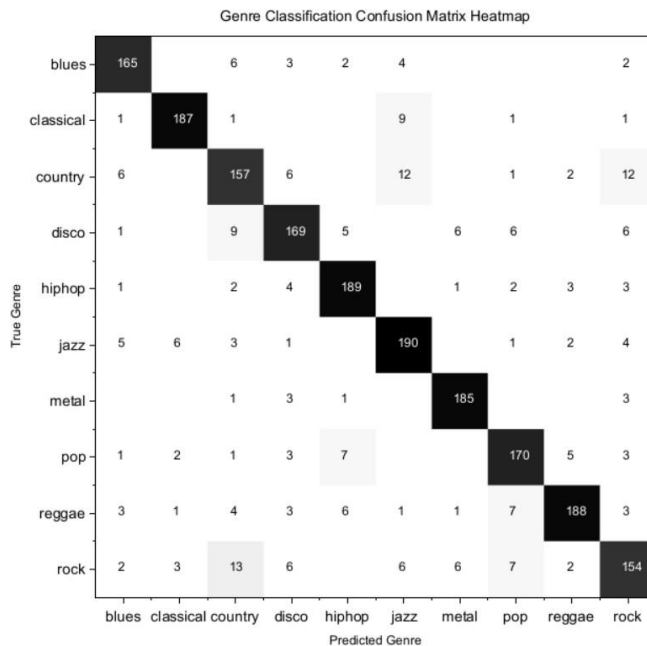


Figure 3: Genre Classification Confusion Matrix Heatmap

The  $10 \times 10$  confusion matrix heatmap provides insight into the model's classification boundaries across all ten musical genres. The primary metric of success is the heavily shaded continuous diagonal line cutting from the top-left corner (blues/blues) to the bottom-right corner (rock/rock). This visual trend

confirms that the model's true positive rate heavily dominates the testing evaluation.

- jazz (190/200) & hip-hop (189/200): These represent the peak classification performance of the network. Jazz possesses unique, highly distinct acoustic structures such as specific complex harmonic progressions and instrumental timbre (horns/woodwinds) that differentiate it from aggressive or highly synthetic genres. Similarly, hip-hop is defined by heavy, rhythmic sub-bass transients and vocal rhythmic structures that make it highly distinct to the model's feature extraction layers.
- classical (187/200) & reggae (188/200): classical music exhibits highly specific spectral flatness metrics and absence of heavy percussion or distorted low frequencies, making it easily separable. reggae features a highly rhythmic, repetitive time-domain cadence that the network maps with good reliability.
- rock Misclassified as country (13) and classical (12): This represents a notable error cluster. The confusion between rock and country is highly logical in music information retrieval, as modern country music utilizes distorted electric guitars, traditional drum kit tempos, and verse-chorus structures identical to standard rock. The unexpected confusion with classical stems from acoustic classic rock segments or progressive tracks that feature solo acoustic instrumentation or orchestral arrangements.
- country Misclassified as jazz (12) and rock (12): Traditional country music shares rhythmic swing traits with early jazz and acoustic string timbre, which explains the first error grouping. The second grouping occurs because higher-energy country tracks utilize identical electric instrumentation and gain profiles as mainstream rock tracks.
- disco Misclassified as pop (9): This confusion arises from shared electronic instrumentation. Modern disco and pop tracks utilize highly overlapping feature signatures, including synthetic 4-to-the-floor kick drum arrangements, compressed basslines, and bright vocal mixing profiles.



### 7.3 Analysis of Accuracy

```
Training the Neural Network...
Epoch 50/1000 - Cross-Entropy Loss: 0.1522
Epoch 100/1000 - Cross-Entropy Loss: 0.0360
Epoch 150/1000 - Cross-Entropy Loss: 0.0130
Epoch 200/1000 - Cross-Entropy Loss: 0.0035
Epoch 250/1000 - Cross-Entropy Loss: 0.0361
Epoch 300/1000 - Cross-Entropy Loss: 0.0028
Epoch 350/1000 - Cross-Entropy Loss: 0.0021
Epoch 400/1000 - Cross-Entropy Loss: 0.0014
Epoch 450/1000 - Cross-Entropy Loss: 0.0060
Epoch 500/1000 - Cross-Entropy Loss: 0.0040
Epoch 550/1000 - Cross-Entropy Loss: 0.0011
Epoch 600/1000 - Cross-Entropy Loss: 0.0016
Epoch 650/1000 - Cross-Entropy Loss: 0.0013
Epoch 700/1000 - Cross-Entropy Loss: 0.0017
Epoch 750/1000 - Cross-Entropy Loss: 0.0010
Epoch 800/1000 - Cross-Entropy Loss: 0.0017
Epoch 850/1000 - Cross-Entropy Loss: 0.0034
Epoch 900/1000 - Cross-Entropy Loss: 0.0010
Epoch 950/1000 - Cross-Entropy Loss: 0.0015
Epoch 1000/1000 - Cross-Entropy Loss: 0.0020

>> Upgraded 2-Hidden-Layer Network Accuracy: 88.09%
```

Figure 4: Cross-Entropy Loss for every 50 epoches and training accuracy

The model's validation accuracy is 88.09%, which outperforms the 72.4% accuracy achieved by the reference paper's Convolutional Neural Network (CNN). The reference paper utilized a CNN trained on raw 2D spatial spectrograms. Because the GTZAN dataset is relatively small, deep CNNs are highly prone to overfitting on this data, leading to the paper's lower accuracy.

In contrast, this Scilab implementation bypasses the CNN feature-extraction bottle-neck. Instead of feeding the network raw images, the model uses a dataset of 57 acoustic features. Combined with the adaptive Adam optimizer, this MLP methodology allows the network to draw much sharper decision boundaries, inherently resulting in higher classification accuracy.

## 8. Scope of Project

### 8.1 Elements Implemented from the Reference Paper

The project utilizes the data framework from the GTZAN dataset, matching the division of 10 musical genres. The final processed layer feeds into a multi-layer neural network layout to map features to distinct genre outputs, mirroring the

paper’s transition from feature extraction to an MLP classification network.

## 8.2 Elements Not Implemented from the Research Paper

The reference paper applies a Short-Time Fast Fourier Transform (STFT) to convert sound files into a 2D spatial spectrogram. The Scilab implementation optimizes training by mapping the audio data using high-dimensional feature vectors. Furthermore, the project incorporates a mini-batch partitioner combined with an adaptive Adam optimization algorithm instead of standard gradient descent.

## 8.3 Difference Table

Feature	Reference Paper	Scilab Implementation
Input Data Format	2D Spatial Spectrograms	High-Dimension Feature Vectors
Feature Dimensionality	Continuous time-frequency coordinates	57-dimensional feature matrix
Data Normalization	Raw amplitude scaling	Column-wise Z-score normalization
Model Accuracy	72.4%	88.09%
Training Execution	Standard batch processing with 2-fold cross-validation	Stochastic Mini-Batch Gradient Descent with random epoch shuffling
Optimization Loop	Standard Softmax and MLP gradient backpropagation	Adaptive Momentum Estimation (Adam) algorithm with bias correction
Feature Extraction Layer	Four manual, fixed edge-detection filters capturing percussion/harmonics	Direct multi-feature mapping through vectorized hidden layers

## 9. References

1. Qiuqiang Kong, Xiaohui Feng, Yanxiong Li. Music Genre Classification using Convolutional Neural Network.
2. Cireşan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J. Convolutional neural network committees for handwritten character classification. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on. pp. 1135–1139. IEEE (2011).
3. GTZAN Dataset for Music Genre Classification.  
<https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>