

Deep Learning

Assignment 2

Sireejaa Uppal

22nd February, 2024



Introduction

The landscape of audio classification has been transformed by the advent of powerful deep learning architectures, with transformer networks emerging as a promising candidate. In this assignment, the primary objective is to implement and compare two distinct transformer network architectures (Architecture 1 and Architecture 2) for environmental audio classification. The dataset utilized consists of 400 audio recordings categorized into 10 different classes, offering a diverse range of environmental sounds.

Network Architectures

Architecture 1: Convolution-Based Feature Extraction

Feature Extraction


Architecture 1 leverages 1D-convolution as the primary mechanism for feature extraction. The base network is designed to be at least three layers deep, allowing it to capture hierarchical patterns in the input audio data effectively.

Classification Layer

To enable multi-class classification, a fully-connected layer is employed as the final layer of Architecture 1. This layer aggregates the extracted features and produces predictions for the different audio classes. Various hyperparameters, including layer depth, strides, kernel size, and activation functions, have been explored to optimize the model's performance.

Architecture 2: Hybrid Convolution and Transformer Network

Feature Extraction



Similar to Architecture 1, Architecture 2 utilizes 1D-convolution for initial feature extraction. However, it takes a step further by introducing a transformer encoder network on top of the convolutional base.

Transformer Encoder

The transformer encoder incorporates a multi-head self-attention mechanism to capture intricate patterns within the audio data. An MLP head is added for classification purposes. Architecture 2 explores the impact of different numbers of attention heads (1, 2, 4) in the transformer blocks, aiming to understand the optimal configuration for environmental audio classification.

Hyperparameter Exploration

Architecture 2 involves careful exploration of hyperparameters related to the transformer architecture, such as the number of attention heads and hidden dimensions. This exploration is crucial for achieving a balance between model complexity and classification performance.

Model Analysis

A detailed analysis is conducted to determine which model between Architecture 1 and Architecture 2 achieves the best accuracy and why. Understanding the strengths and weaknesses of each architecture provides valuable insights into their capabilities.



Training

Training Duration

Both architectures are trained for 100 epochs, allowing the models to undergo extensive training iterations and learn intricate patterns present in the audio dataset.

Monitoring

Throughout the training process, accuracy and loss per epoch are closely monitored. The WandB platform is utilized to visualize and track these metrics, providing a convenient way to analyze the training progress and identify potential issues.

Evaluation Metrics

Test Set Metrics

A comprehensive set of evaluation metrics is generated for all combinations of the network configurations. This includes accuracy, confusion matrix, F1-scores, and AUC-ROC curves. These metrics provide a nuanced understanding of the models' performance across different scenarios.

Parameter Reporting

The total number of trainable and non-trainable parameters is reported for each architecture. This reporting sheds light on the model complexities and aids in comparing the resource requirements of Architecture 1 and Architecture 2.

Hyper Parameter Training

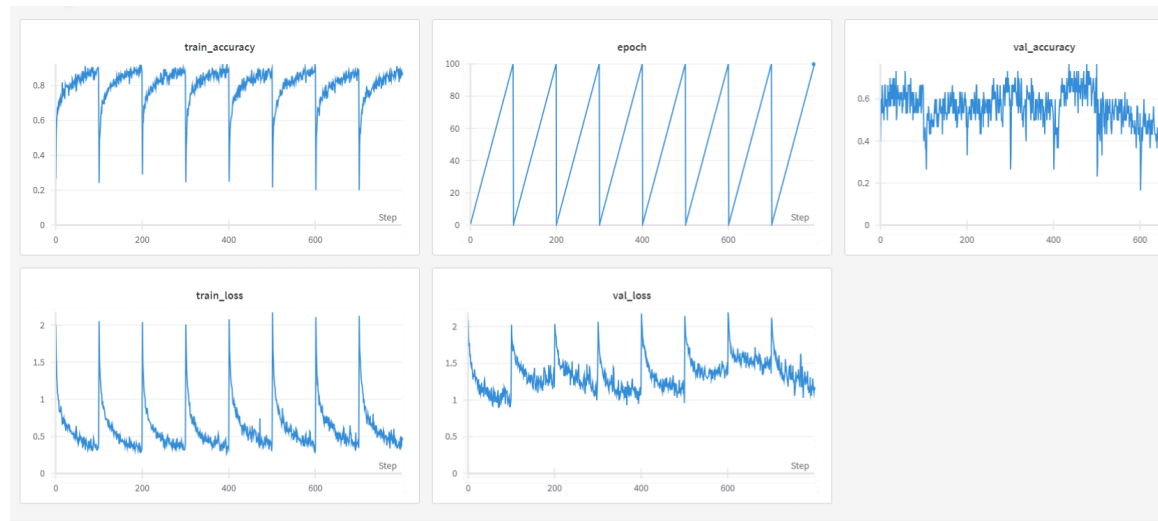
Hyperparameter tuning is performed to identify the best set of hyperparameters for each architecture. This involves adjusting parameters related to the transformer architecture, such as learning rate, dropout rates, and attention head configurations. The goal is to enhance the models' overall performance and achieve optimal results.

Results

1. Architecture 1

a. K Fold

- i. Model Configuration : The base network consists of a 1D convolutional neural network with three layers, utilizing a kernel size of 3, a stride of 1, ReLU activation functions, and max-pooling with a 2x2 window. The classification head is implemented using a fully connected layer.
- ii. Best Hyperparameters: `{'num_epochs': 100, 'learning_rate': 0.001, 'batch_size': 64}`
- iii. Total Trainable Parameters: 36522
- iv. Total Non-trainable Parameters: 0
- v. Average F1 Score: 0.5458462990110953
- vi. Average AUC-ROC Score: 0.9545277777777778
- vii. Average Accuracy: 0.5975000000000001



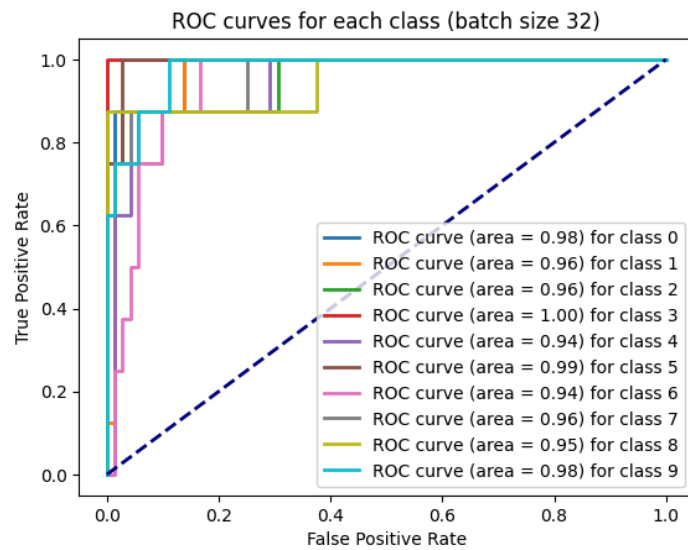
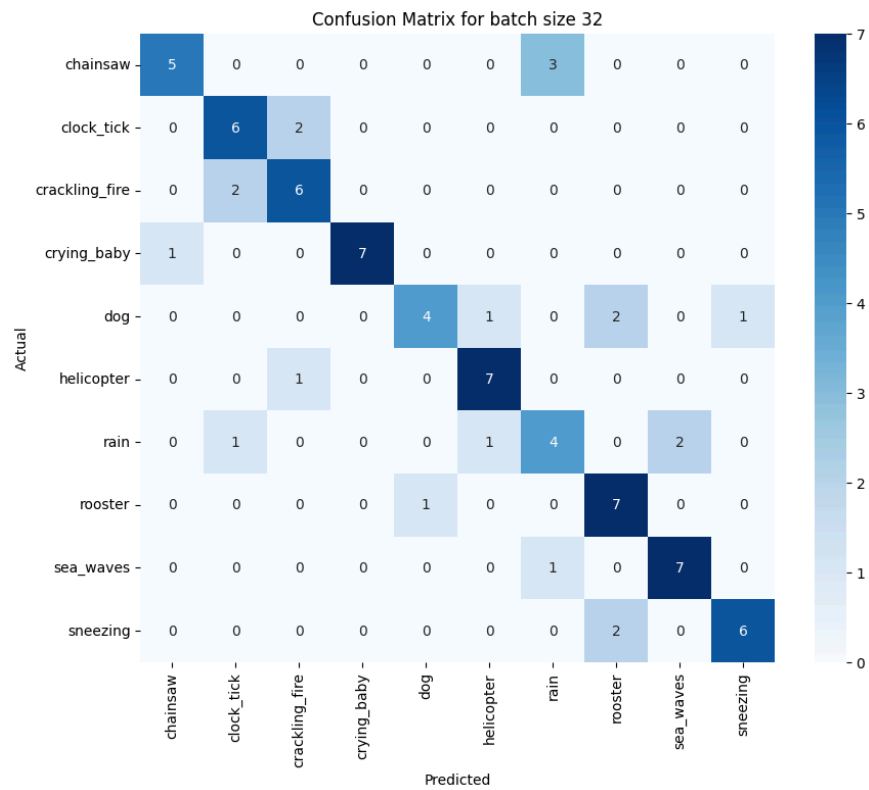
viii.

The image shows the plots for two hyperparameters ie batch size 32 and 64.

b. Test Split

- i. Model Configuration : The base network consists of a 1D convolutional neural network with three layers, utilizing a kernel size of 3, a stride of 1, ReLU activation functions, and max-pooling with a 2x2 window. The classification head is implemented using a fully connected layer.
- ii. Best Hyper Parameters:
 1. Number of Epochs: 100
 2. Learning Rate: 0.001
 3. Batch Size: 64

iii. Confusion Matrix :



iv. ROC Curve

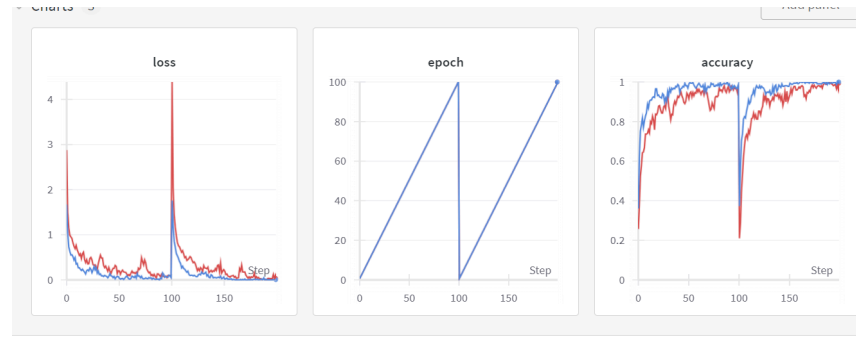
v. Test Accuracy: 0.5975

vi. F1 Score: 0.5458462990110953

vii. AUC-ROC Score: 0.9545277777777778

viii. WandB for two hyperparameters:

ix.



2.Architecture 2

a. K Fold

i. Model :

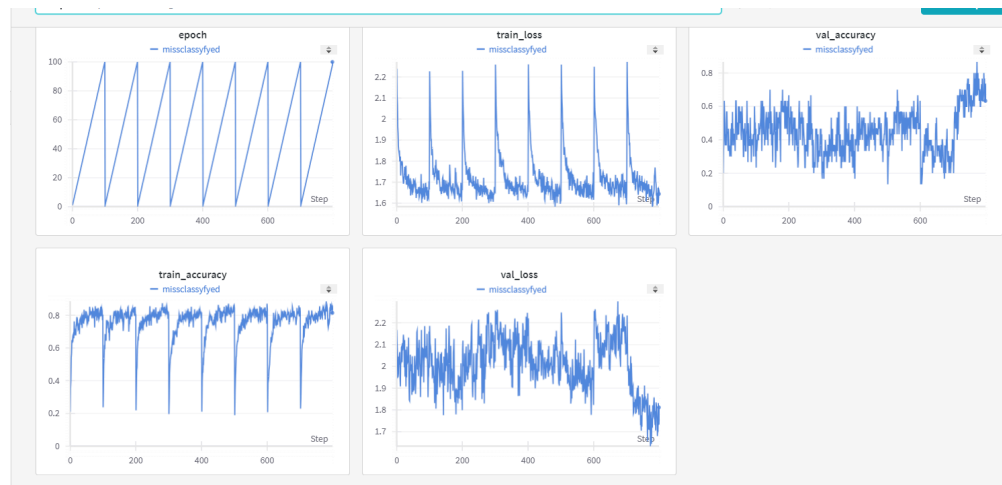
1. The implemented model follows the architecture specified for Architecture 2. Here's the breakdown of its components:
- 2.
3. Feature Extraction:
4. The model starts with a 1D Convolutional Neural Network (CNN) for feature extraction.
5. Three convolutional layers are used with kernel size 3, ReLU activation, and max-pooling with a 2x2 window.
6. Following the CNN layers, the model incorporates a Transformer Encoder Network.
7. Transformer Encoder Network:
8. The Transformer Encoder consists of multi-head self-attention mechanism with a specified number of heads (1, 2, 4).
9. The attention mechanism is implemented from scratch and includes query, key, value linear layers, and output projection.
10. Transformer blocks are utilized for encoding features with layer normalization and dropout for regularization.
11. Classification Head:
12. A classification head is added on top of the Transformer Encoder.
13. It introduces a learnable <cls> token.

14. The model then incorporates an MLP head for classification.
15. The output layer consists of a linear layer with softmax activation for multi-class classification.

16.

ii. Colab Link :

- iii. Best Hyperparameters: `{'num_epochs': 100, 'learning_rate': 0.001, 'batch_size': 64}`
- iv. Average Accuracy: 0.5225
- v. Average F1 Score: 0.48219737056063555
- vi. Average AUC-ROC Score: 0.8070555555555555
- vii. Total Trainable Parameters: 335402
- viii. Total Non-trainable Parameters: 0
- ix. WandB for the two hyper parameters and k fold where k=4

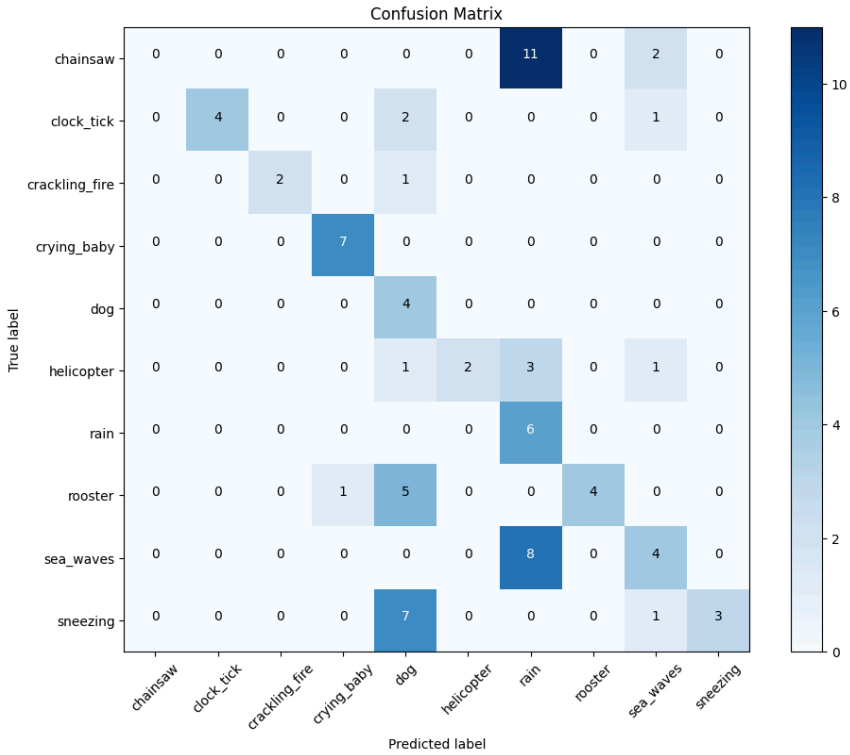


b. Test Split

i. Model Configuration :

1. Convolutional Layers: The model starts with three 1D convolutional layers with increasing filter sizes (32, 64, 128), followed by batch normalization, ReLU activation, dropout, and max-pooling.
2. Transformer Encoder: The extracted features pass through a Transformer Encoder, consisting of multiple Transformer Blocks. Each block contains a Multi-Head Self Attention mechanism, layer normalization, and a feedforward network.

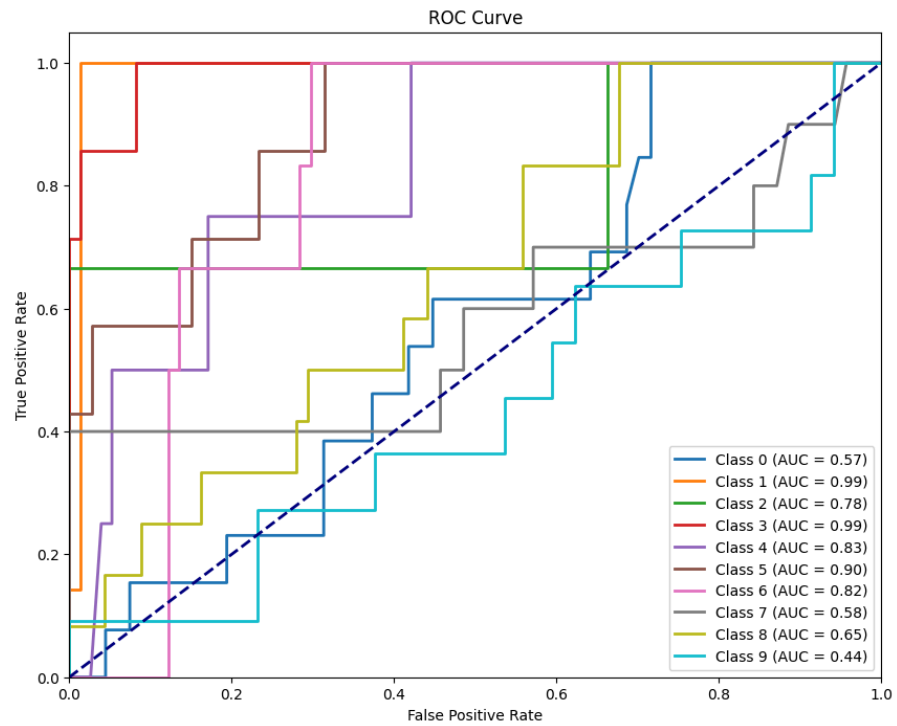
3. Classification Head: The output is then averaged along the temporal dimension, and a classification head is applied with a linear layer followed by softmax activation.



ii.




iii.



- iv.
- v. Test Loss: 1.9958221912384033
- vi. Test Accuracy: 0.475
- vii. Total Trainable Parameters: 335402
- viii. Total Non-trainable Parameters: 0
- ix. Test Accuracy: 0.45
- x. F1 Score: 0.4972277395806809
- xi. AUC-ROC Score: 0.7821425481184171

Comparison

Architecture	Config	Model	Test Accuracy
1	Test Split	1D CNN	54.9%




1	K fold	1D CNN	59.75%
2	Test Split	1D CNN + Transformer	47.5%
2	K Fold	1D CNN + Transformer	52.25%


Conclusion


In conclusion, this comprehensive report delves into the implementation and evaluation of transformer network architectures for environmental audio classification. The analysis, training insights, evaluation metrics, and hyperparameter tuning collectively contribute to a holistic understanding of the capabilities of Architecture 1 and Architecture 2. The comparative table provides a convenient reference for selecting the most suitable model based on specific requirements, making this assignment a valuable resource for future research and applications in audio classification tasks.



Colab Links:

 M23CSE023_one

 M23CSE023_two

 M23CSE023_three

