Single Event Detection

Lakshya Gujar
Department of Electrical Engineering
IIT Kanpur
lakshyag@iitk.ac.in

1 Introduction

We have been given a dataset with 1000 spectrograms and their labels for this task. We must determine which event the provided audio's spectrogram corresponds to. This report will go over our strategy, process, experiments, and outcomes for the assigned work.

2 Literature Survey

For sound classification, a variety of machine learning and signal processing techniques, including matrix factorization, dictionary learning, etc., have been used. However, CNN in particular produces the best results, according to a study titled "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification." They can record energy modulation patterns over time and frequency when applied to spectrogram-like inputs, which is a crucial ability for differentiating between various, frequently noise-like noises like engines and jackhammers. Second, the network should be able to correctly train and later recognise spectro-temporal patterns that are indicative of various sound classes by employing convolutional kernels (filters) with a narrow receptive field.

3 Method

3.1 Data Preparation

A dataset of 1000 tagged spectrograms from 10 sample classes has been handed to us. We require one dimension for the CNN model's input in order to implement it. We must preprocess our data in accordance with the different dimensions of the spectrograms in our given dataset in order to execute CNN. After reviewing a few references, we made the decision to use below approaches for this.

To assess our model's performance after training, we must divide our dataset into training data and validation data. We randomly selected 15% of the data for each class to create the validation dataset.

3.2 Convolutional Neural Network

A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign learnable weights and biases to various aspects in the image and be able to differentiate one from the other.

3.2.1 Model Details

We utilised 3 CNN layers because they enable learning time-frequency filters to locate pertinent representations and learn high-level features. Max pooling returns the maximum value from the area of the picture covered by the kernel, while pooling minimises the spatial size of the convolved feature, reducing the computational resources needed to analyse data. Hence, every CNN layer is followed by

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Max Pooling and certain small dropouts to prevent overfitting. The activation function for each CNN layer used is **ReLU**.

We have one final fully linked layers after going through three layers of CNN, with max-pooling and dropout occurring after each layer. The degrees of freedom of the model are determined by hidden layers in the dense units. The fully connected layer of 100 units has an activation function **softmax**.

The model was trained using 200 epoch with the batch size of 50. We have used **"categorical crossentropy"** for loss calculation and **"rmsprop"** Optimizer with a decay and learning rate of 0.01.

The entire architecture of the model can be seen below.

Layer (type) Output Shape Param

dense (Dense) (None, 100) 12900

activation (Activation) (None, 100) 0

dropout (Dropout) (None, 100) 0

dense_1 (Dense) (None, 200) 20200

activation_1 (Activation) (None, 200) 0

dropout_1 (Dropout) (None, 200) 0

dense_2 (Dense) (None, 100) 20100

activation_2 (Activation) (None, 100) 0

dropout_2 (Dropout) (None, 100) 0

dense_3 (Dense) (None, 10) 1010

activation_3 (Activation) (None, 10) 0

Total params: 54,210
Trainable params: 54,210
Non-trainable params: 0

When we took the fixed dimensions as **(128,1000)** and concatenating, we have been able to achieve training accuracy of **76%** and validation accuracy of **88%** with 200 epochs.

4 Results

We calculated metrics for each label, and find their average.

4.1 Precision & Recall

0.58 & 0.79
0.75 & 0.50
0.76 & 0.89
0.35 & 0.72
0.62 & 0.56
0.80 & 0.67
0.80 & 0.89
0.82 & 0.52
0.77 & 0.71
0.62 & 0.28

4.2 F1 Score

0.67
0.60
0.82
0.47

0.59
0.73
0.84
0.67
0.74
0.38

4.3 Confusion Matrix

[[0.57575758 0.79166667 0.66666667]
[0.75 0.5 0.6]
[0.76190476 0.88888889 0.82051282]
[0.35135135 0.72222222 0.47272727]
[0.625 0.55555556 0.58823529]
[0.8 0.66666667 0.72727273]
[0.8 0.88888889 0.84210526]
[0.82352941 0.51851852 0.63636364]
[0.77272727 0.70833333 0.73913043]
[0.625 0.27777778 0.38461538]]