EE603: Assignment 1 - Single Event Detection

Somya Gupta
somyagupta20@iitk.ac.in
200993

Abstract

The assignment focuses on detecting the sound(multi-class classification) given the array as an input. We use various methods such as Convolutional Neural Networks(CNN), Decision tree classification and KNN to classify the test set into sound labels. We use accuracy and f1-score as a means to determine the best model for classification. The results we have obtained show us that CNN is the best option for constructing the classifier with given training set.

1 Introduction

Classification is a supervised learning task for which the goal is to predict to which category a given sample belongs. Classification is the basis of many applications, such as detecting if an email is spam or not, identifying images, or diagnosing diseases, sound classification etc.

2 Literature Survey

The CNN is made up of three types of layers: convolutional layers, pooling layers, and fully-connected (FC) layers. The convolution layer is responsible for the extraction of the different features from the input images. The Fully Connected (FC) layer comprises the weights and biases together with the neurons and is used to connect the neurons between two separate layers. The Pooling layer is responsible for the reduction of the size(spatial) of the Convolved Feature. To avoid overfitting (when a model performs well on training data but not on new data), a dropout layer is utilised. The activation functions are used to learn and approximate any form of network variable-to-variable association that's both continuous and complex. Early stopping is a method that allows you to specify an arbitrary large number of training epochs and stop training once the model performance stops improving on a hold out validation dataset.

3 Method

3.1 Data Pre-processing

- Importing libraries
- Importing the datasets :Extracting the input data and output labels
- Finding Missing Data : First, we convert our data to a uniform size of (1,128,300) by padding the smaller arrays with 0 and uniformly slicing off the larger arrays from both left and right side.Then, we replace the missing data with the mean of that row or column
- Encoding Categorical Data : This involves converting the string labels (such as 'Bark' or 'Meow') to integer labels (1 or 2) and then converting these to one-hot vectors
- Reshape the data : We make a stack of the data and reshape it so as to easily process the CNN model
- Splitting dataset into training and test set

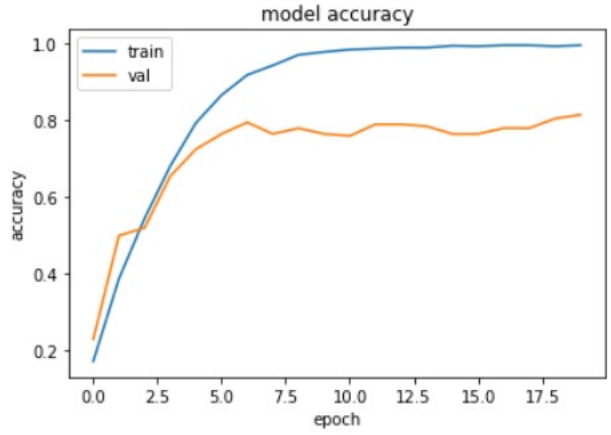
Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

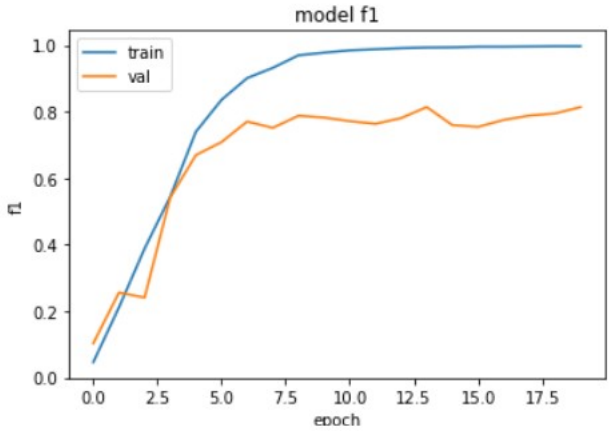
3.2 Training the model

- **Convolutional Neural Networks (CNN)** : This is the best model for our given data inputs. We have used 2 layers of Conv2D and MaxPooling2D and added some dropout layers to minimise overfitting loss. We have then flattened the array and used dense layers with activation functions relu and softmax. We have monitored categorical cross entropy loss using an adam optimiser.

  Obtained training accuracy=(0.997)
  Obtained training f1-score=(0.997)
  Obtained validation accuracy=(0.82)
  Obtained validation f1-score=(0.82)

```
Epoch 20/20
25/25 [==============================] - 105s 4s/step - loss: 0.1090 - accuracy: 0.9962 - f1_m: 0.9968
- precision_m: 1.0000 - recall_m: 0.9937 - val_loss: 0.6750 - val_accuracy: 0.8150 - val_f1_m: 0.8137
- val_precision_m: 0.9223 - val_recall_m: 0.7321
```





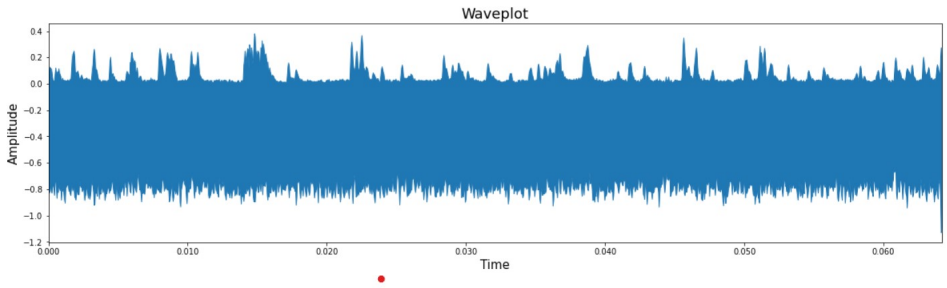
- **K-nearest neighbors (KNN)** : I have used the built in library function for implementing KNN. This algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

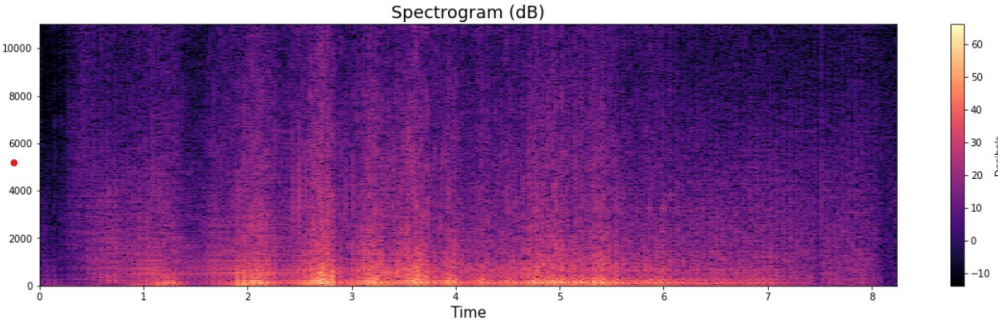
- **Decision Tree Classifier** : I have used the built in library function for implementing Decision Tree Algorithm. For predicting a class label for a sound we start from the root of the tree. We compare the values of the root attribute with the given sound's attribute. On the basis of the comparison, we follow the branch corresponding to that value and jump to the next node.

3.3 Data visualisation

I have plotted waveplots for some of the given files to better visualise the sound data given. Along with this, I have also plotted spectrograms containing the data given to us, along with the values of nfft, nmels, hann window specifications.

Eg. The following depict one of the 'Meow' sounds:





3.4 Classification report

I have judged the model on the basis of its accuracy, f1 score, recall and precision and then returned the results. I have used the classification report to analyse the F1 score for each class separately, along with the weighted and macro averages of F1 scores.

4 Measures of Result Relevancy

4.1 Confusion matrix

Table 1: Decision Tree for label Walkandfootsteps

		Predicted label	
		Negative	Positive
True label	Negative	156	26
	Positive	16	2

Table 2: KNN for label Walkandfootsteps

		Predicted label	
		Negative	Positive
True	Negative	174	8
label	Positive	14	4

Table 3: For CNN

```
[[13  2  0  1  0  1  0  1  1  1  3]
 [ 0 16  0  0  0  2  0  0  0  0  0]
 [ 0  0 14  1  1  0  0  1  0  0  1]
 [ 1  0  0 14  0  1  0  0  0  0  2]
 [ 0  1  1  3  8  1  0  0  0  3  1]
 [ 1  0  0  2  0 13  0  1  0  0  1]
 [ 0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  1 15  0  1  1]
 [ 0  1  2  0  1  2  0  1 17  2  1]
 [ 0  0  0  3  3  0  0  1  3 14  0]
 [ 1  0  1  6  0  0  0  1  0  1  8]]
```

4.2 Precision

$P = \frac{TP}{TP+FP}$

- For CNN: 0.69
- For KNN: 0.33
- For decision tree algorithm:0.12

4.3 Recall

$R = \frac{TP}{TP+FN}$

- For CNN: 0.66
- For KNN:0.22
- For decision tree algorithm:0.122

4.4 F1-score

$F1 = 2 * \frac{P*R}{P+R}$

- For CNN: 0.67
- For KNN: 0.27
- For decision tree algorithm: 0.12

4.5 Accuracy

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

- For CNN: 0.70
- For KNN: 0.17
- For decision tree algorithm: 0.10

5 Observation and Conclusion

- Removing NaN values vs replacing them with mean: I observed that using SimpleImputer to replace the values with mean gives better results in classifying the data
- Construction of CNN: If we use MaxPooling with kernel size 2, we will get lower accuracy and f1-score (0.5-0.6 with 2 convolution layers) Using a larger kernel gives better results (0.73 with 2 convolution layers)
- Overfitting: Using dropout layers reduces overfitting of data. Along with that, another possible way to reduce the overfitting of data is l1 or l2 regularisation. L1 regularisation gives lower accuracy (0.6 with 3 convolution layers) and f1-score as compared to using l2 regularisation (0.7 with 3 convolution layers)
- F1-score is a better measure of correctness for our classifier since it takes into consideration both precision and recall
- Using the classification report, we obtain the weighted, micro, macro and sample averages for precision, recall, f1-score. We conclude that weighted average will be the best means of comparison since it takes into account the number of samples of each class before calculating the average for the whole data.
- If we use too many CNN convolution layers, then the accuracy and f1 score decrease, and time taken is of course more. Similarly, lesser number of layers give bad results even though they take lesser time.
- KNN vs Decision Tree: Since we have sound labels, a similar sound produces similar spectrograms, gives closer array values, thus KNN can better classify the data than a decision tree.
- Spectrograms: Using waveplots and spectrograms, a pattern for different sounds can be observed and hence it is a good measure of visualising sound data
- Lastly, we can conclude that **Convolutional Neural Networks(CNN) is the best model** to classify the given data. It gives the best accuracy and f1-score even on test data.