Comparing Noise Augmentation Methods using Shallow and Deep Neural Network Architectures for Automatic Music Genre Classification

*Applied Deep Learning Report

Andrew Morgan
University of Bristol
jp19060@bristol.ac.uk

I. Introduction

Labeling audio files by genre is a difficult problem as genres are not mathematically defined, but by using deep learning we can explore a multi-dimensional feature space, treating the problem as pattern recognition. Convolutional neural networks already label music better than state-of-the-art none machine learning programs. In this report the topic of noise augmentation will be explored via two CNN architectures.

II. Related Work

The work on the topic of automatic music genre classification uses many different models to attempt to increase the accuracy of the classifications. These papers use a variety of data augmentation styles, they also use a variety of different techniques to implement noise. We observe from the following.

[1]In Exploring Data Augmentation to Improve Music Genre Classification with ConvNets we observe that the addition of noise adds the least accuracy out of all of the data augmentations. While this is true it also adds the least data n * 2 rather than n * 3. This paper also takes a unique approach to noise, adding the noise after the data is in spectrogram form. The paper sets a spectrogram's pixel to 0 with a probability of 0.1, this method will be refereed to as "Pixel Dropout".

[2]In Music Genre Classification using Neural Networks with Data Augmentation we observe the usage of Gaussian noise. We should focus on Figure 9 as we are also using a CNN not a RCNN. From this we can see that most genres benefit from Gaussian noise. The effects of this noise are varied between different genres.

[3]In Machine Learning and Noise Reduction Techniques for Music Genre Classification we observe that noise reduction improves the accuracy of classification on multiple models. The noise reduction used in this report is not what is implemented, instead a much simpler form of noise reduction is used. The method implemented smooths the data using a convolution with different strengths.

III. Results

This table shows different styles of noise addition/reduction and there effect on the accuracy. All of the variants of the data are from GTZAN.

Data	Shallow (%)	Deep (%)	Data Points
Std	63.15	63.76	11,750
Std + G	62.71	64.36	35,250
Std + PD	58.22	60.19	23,500
Std + R	61.32	64.51	35,250
Std + G + PD	60.80	57.39	70,500
Std + G + R	61.60	67.24	70,500
Std + PD + R	60.90	62.18	70,500
Std + G + PD + R	55.44	57.90	141,000

Std = GTZAN
G = Gaussian Noise
PD = Pixel Dropout
R = Reduced Noise

A. Analysis

The results obtained don't support the correlation between more data and better accuracy. If that were true then the last entry (a combination of all the data augmentation) would perform the best. Instead we see that pixel dropout performs poorly across the board with adding and reducing noise making small increases in accuracy.

This data might suggest that combining all the augmentation methods made the signal harder to detect, we could also use this data to suggest that adding and reducing noise preventing over fitting allowing the model to learn for longer.

IV. Dataset

The original training split is 11,750 data points with every point consisting of a tuple containing a filename, a spectogram, a label and an audio sample. The test/validation set is 3,750 data points containing the same.

Augmentation	Amount of Data
None	n
Noise	n * 3
Dropout	total data * 2
Reduce noise	n * 3

n = 11,750

The table shows the amount of data generated by the different types of data augmentation. The validation set is never added to, but the spectograms within it are changed. The reason for this is that the effect of adding and reducing noise is put in to the audio sample before it is converted into a spectogram. To be sure that the way we make spectograms is the same between training and validation we need to remake the validation spectograms.

A. Noise

The noise added is Gaussian noise, it makes 23,500 new data points. It does this by applying 3 levels of noise to each data point.

B. Dropout

Dropout refers to pixel dropout, this just doubles the amount of data in the system because while a version of the data is unaffected, another version is made with a 10% chance to set the a pixel value to 0.

C. Reduced

This method uses a convolution to smooth the data, it makes 23,500 new data points. It does this by applying two levels of smoothing to each track.

V. CNN Architecture

The two architectures that have been implemented are deep and shallow.

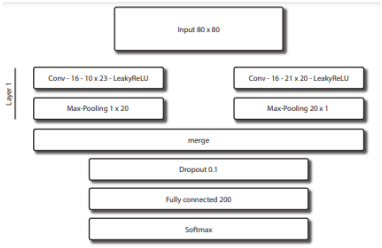

Fig. 1. Shallow Architecture

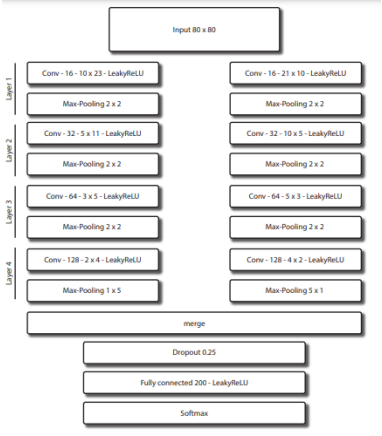

Fig. 2. Deep Architecture

VI. Implementation Details

A. Models

All models and data augmentation options are run from model.py, here are some of the key flags that control it's execution.

Flag	Function
–deep	Runs the deep model
–noise	Adds noise
–reduced	Adds smoothing
–pixel	Adds pixel dropout

All these flags take a Boolean.

B. Data Augmentation

This program does not take in any flags but instead variables must be changed within the code. These variables split the work to be done by into chunks. The work being done in the data-augmentation.py script involves adding or removing noise. The val-augmentation.py reproduces the spectograms so that they match the method used to generate them in the new augmented data.

The data augmentation code splits the task in to 4 and combines the data on blue crystal with the script combine.py. The reason the task has been split is so that it can be run on a machine with access to lesser resources (in this case memory).

The "pixel dropout" is added before the model runs inside model.py. This is done because pixel dropout doesn't need to remake the spectogram. There is no new train.pkl made for this method.

VII. Replicating Quantitative Results

These are the results that attempt to mirror the results of "Comparing Shallow versus Deep Neural Network Architectures for Automatic Music Genre Classification".

Model	Raw Accuracy	Max Accuracy	Epochs
Shallow	62.04	63.09	100
Deep	63.50	65.01	100
Shallow	63.15	65.72	200
Deep	63.76	66.10	200
Shallow Aug	61.60	63.96	200
Deep Aug	67.24	68.02	200

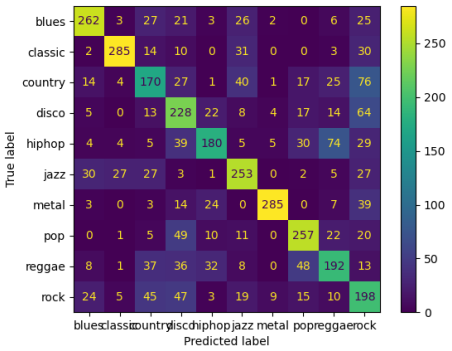

Fig. 3. Deep Architecture's Matrix

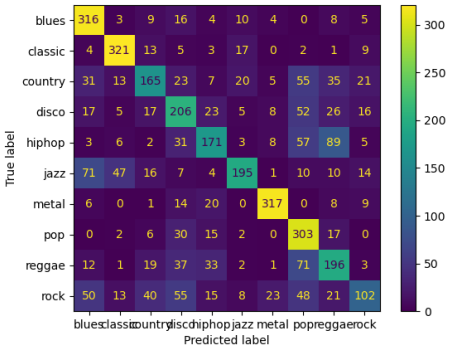

Fig. 4. Shallow Architecture's Matrix

VIII. Training Curves

A. Training Curves

The data augmentation reduces over fitting slightly but the both models still tend to over fit. The graph show that reducing noise does a better job at stopping over fitting than adding it.

Unfortunately this graph also shows my data augmentations to be ineffective, a better form of augmentation might of kept the model from reaching 95-100% batch accuracy so quickly. These results are supported by the lack of test accuracy increase from 100 to 200 epochs.

Number	Genre
1	Deep
2	Shallow
3	Deep Noise
4	Shallow Noise
5	Deep Reduced
6	Shallow Reduced
7	Deep Pixel Dropout
8	Shallow Pixel Dropout

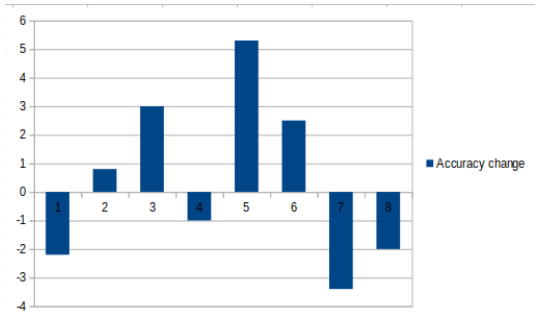

Fig. 5. Changes in accuracy from 100 epochs to 200 epochs

As you can see the model doesn't keep learning for the whole experiment.

B. Testing Curves

This curve further supports the method of reducing noise, it can be seen that the model using the reduced noise learns for longer and ends the exsperiment with a highier accuracy as a result.

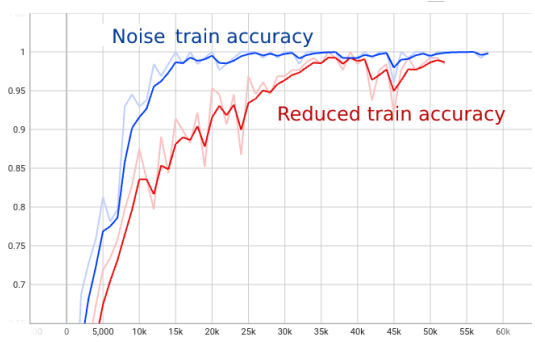

Fig. 6. Train Accuracy Curves

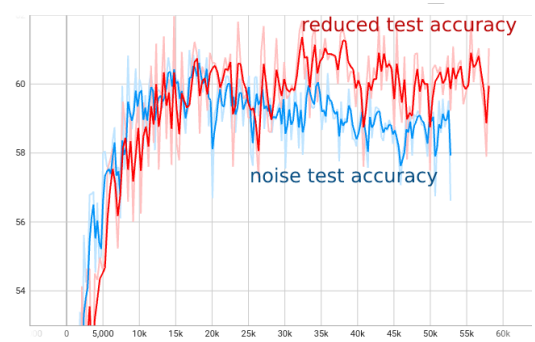

Fig. 7. Test Accuracy Curves

C. Loss Curves

This curve shows that over fitting happens very quickly with my model, it also shows that again reduced noise over fits slightly less later into the experiment.

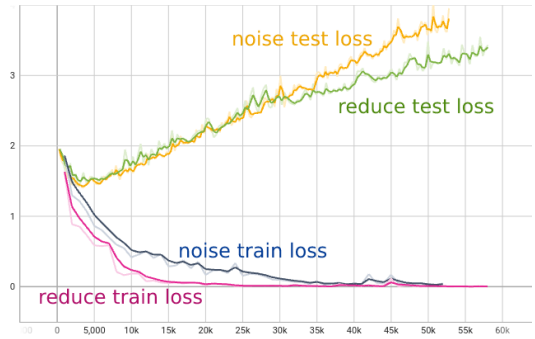

Fig. 8. Loss Curves

IX. Qualitative Results

In the graphs the numbers represent the genre labelled.

The graphs shows both the deep and shallows results, along with all the data augmentation methods tried in this report. They also show that none of the augmentation methods improve the classification for every class.

The graph also shows that data augmentation reduces the accuracy for disco on both models.

Number	Genre
1	Blues
2	Classic
3	Country
4	Disco
5	Hiphop
6	Jazz
7	Metal
8	Pop
9	Reggae
10	Rock

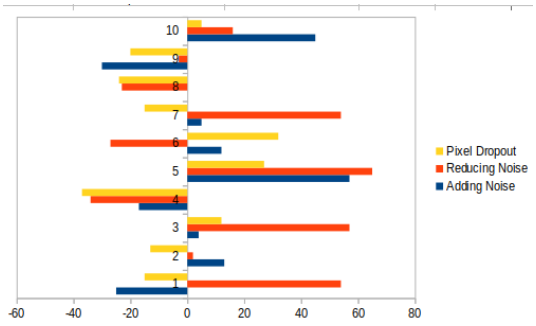

Fig. 9. Deep Architecture's Number of Correct Predictions per genre

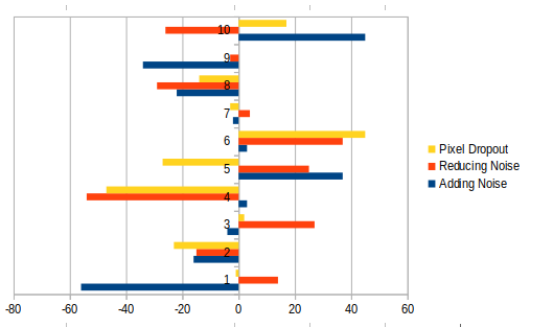

Fig. 10. Shallow Architecture's Number of Correct Predictions per genre

A. *Shallow Network and Rock*

The shallow network really struggled with identifying rock, before any augmentation the network had a 27% accuracy. Reducing noise further decreased this accuracy to 21%, and in the best case noise was added to the data and the accuracy was 39%.

B. *Deep Network and Hiphop*

The deep network with no augmentation has a 48% accuracy when identifying hiphop, after reducing noise the accuracy goes to 64%. In fact all forms of noise increases the accuracy for the deep network identifying hiphop.

C. *Disco*

Both networks have about a 55% accuracy for identifying disco, augmenting the data has a 9% decrease in accuracy on average across both models.

X. Improvements

The improvements that have been explored focus on data augmentation and deeper neural networks. Theoretically the more data provided the better a neural network should do, this should be especially true for the deeper neural network.

Applying different types of data augmentations gives a more complete view of the effects.

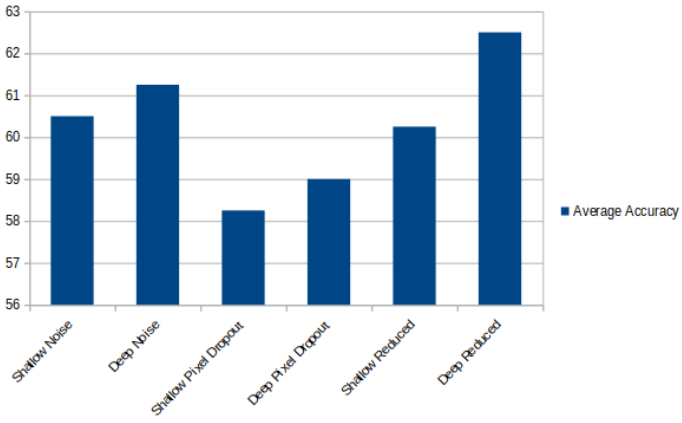

Fig. 11. Average Accuracy for each Augmentation

The graph shows that deep reduced has the best overall performance, and it shows that pixel dropout is the worst data augmentation displayed here. Pixel dropout uses are arbitrary 0.1 probability whereas the noise added or reduced can still be checked to see if the values used make intuitive sense. This can be done by listening to the edited audio track and deciding weather you can determine the genre. The same cannot be done with pixel dropout.

XI. Conclusion

In summery this report shows that data augmentation has varying results, it also shows that deeper neural networks benefit most off of augmentation. This report also compares three types of augmentation, the comparison showed that data augmentation should avoid big chances. Pixel dropout has a chance of changing a value on the spectogram to 0 from a potentially high value. The other augmentation methods only change values proportionally, therefore the data still follows the same "trend".

To improve the results of this report more testing would be needed, firstly changing the amount of noise reduced and added would give a clearer image of the effects. Secondly varying the percentage dropout for pixel dropout would maybe highlight potential for this method with a lower probability than 0.1.

A. *Further Work*

The work done in this report show that data augmentation can make small changes to accuracy depending on it's implementation and method. A further exploration of this could try to identify where this line is, how much augmentation is too much and how does the relationship between augmentation and accuracy look.

REFERENCES

[1] Rafael L. Aguiar, Yandre M. G. Costa, andCarlos N. Silla Jr, "Exploring Data Augmentation to Improve Music Genre Classification with ConvNets" . IEEE, 2008.

[2] Macharla Vaibhavi, P. Radha Krishna, "Music Genre Classification using Neural Networks with Data Augmentation" .J.Innovation Sciences and Sustainable Technologies, 1(1)(2021), 21 - 37.

[3] Omkar Chavan, Nikhil Kharade, Amol Chaudhari , Nikhil Bhalke, Prof. Pravin Nimbalkar "Machine Learning and Noise Reduction Techniques for Music Genre Classification" 2019, IRJET.