Analyzing Music Effects on Brain Waves according to Functional Measures

Qingzheng Xu

Research School of Computer Science,
Australian National University
Canberra, Australia
u6174243@anu.edu.au

Abstract. Researchers in the field of affective neuroscience have found that music can affect brain activity. Different types of music can be identified based on analyzing their effects on brain waves. Researchers can build neural networks to find the relationship between brain waves and music; however, the data extracted from the electroencephalograms often contain redundant or irrelevant information, which may decrease the modelling efficiency and classification accuracy. In this paper, a Neural Network is built, which aims to use data from the electroencephalograms of the frontal lobe to predicate the music type among 3 different music genres. The validation accuracy of the initial network is low (47%) compared with the model of Rahman et. al. (97.5%) [1] because the dataset used in this paper is a subset of Rahman et. al.'s dataset. To deal with it, four kinds of models are used to prune redundant data points, while these models can calculate the importance of input fields based on Functional Measures. According to the ranking of models' outputs, by pruning the 5 most significant inputs, the validation accuracy can drop from around 47% to around 42%; by pruning the 5 least significant inputs, the model performance will not be influenced a lot and the validation accuracy stays at around 47%. The results show that the techniques used in this paper are effective in pruning irrelevant information and will be helpful for the analysis of music effects on brain waves.

Keywords: Brain Activity, Affective Neuroscience, Electroencephalogram, Neural Network, Classification, Data Mining, Functional Measures.

1 Introduction

The relationship between music and brain activity is attractive to researchers in the field of affective neuroscience. Music is supposed to be helpful in the treatment of tension and negative emotions [2]. Moreover, music seems can improve children's reading abilities and mathematical task performances [3]. Rahman et. al. [1] built a classification model using Neural Network which can classify the music type according to brain wave patterns and the accuracy can reach up to 97.5%. By analyzing the effects of different kinds of music on brain activity, researchers can identify which music type will have positive effects on human brains.

In this paper, the main task is to use Neural Networks for classifying the type of music from 3 music genres based on features extracted from 24 participants' frontal lobe electroencephalograms provided by Rahman et. al. [1]. By choosing this dataset, the outputs can be compared with Rahman's, which makes it convenient for further analysis. Those Neural Networks were trained using error-backpropagation [4] and the initial network topology is 26-40-3. Simple weighted links are used to connect each neuron in a layer to each neuron in the next layer, without backwards connections and multi-layer connections. The activation function is the sigmoid function, and the optimizer is Adam. After building up the Neural Network, the initial dataset is spitted into the train, validation, and test set in the ratio of 8:1:1. With a learning rate of 0.01, the net with the highest validation accuracy will be selected from the first 1000 epochs and the test accuracy for the chosen net will be calculated as an evaluation method for the model.

The additional task for this paper is to prune redundant data points and try to improve validation/testing accuracy. To do so, four different models introduced by Gedeon are applied to prune inputs based on Functional measures [5].

This paper facilitates the following:

- Automatic analysis of electroencephalograms and identify the behaviors of brain waves under different types of music. This would assist the research on finding the suitable music genre for psychological treatment and the development of children's learning skills.
- The ranking of inputs' importance and suitable inputs selecting. This paper shows and verifies an effective and efficient method for dataset mining. Its limitations will be discussed, too.

2 Method

Pre-processing steps including standardization and transcoding are applied to make the dataset more suitable for training. Four models (model W, U, I and C) are applied for sorting the inputs based on their Functional measures. Modifications have been made on these models to fit the dataset.

2.1 Functional measures

Functional measures can determine the similarity between two hidden neurons over a training set, which calculates the angles between activation results vectors of those neurons [6]. Here is its formula.

$$angle\ (i,j) = tan^{-1}\left(\sqrt{\frac{\sum_p^{pats} sact\ (p,i)^2 * \sum_p^{pats} sact\ (p,j)^2}{\sum_p^{pats}(sact\ (p,i) * sact\ (p,j))^2} - 1}\right) \tag{1}$$

$$\text{where } sact\ (p,h) = activation\ (p,h) - 0.5 \tag{2}$$

Gedeon extended this technique, and it can now determine the similarity between two hidden neurons based on the weight matrix [7]. To evaluate the inputs, the technique should be modified to calculating vectors of the weight matrix belongs to different input fields [5]. Here is the new formula for equation (2).

$$\text{where } sact\ (p,h) = norm\ (weight\ (h)) - 0.5 \tag{3}$$

In equation (3), the weight matrix is normalized. By subtracting 0.5, about half of the values in the matrix will be positive and the others will be negative, which will lead to better output angles. In this paper, according to the dataset characteristic and model performance, this equation is modified to use standardization instead. Here is the new formula for equation (3), which is used in model W.

$$\text{where } sact\ (p,h) = stand\ (weight\ (h)) \tag{4}$$

Besides, this technique can be adjusted to analyse the input data itself as well. In this case, each feature column will be considered as a vector for calculating angels [5]. Here is the new formula for equation (2).

$$\text{where } sact\ (p,h) = pattern\ (h) - 0.5 \tag{5}$$

As mentioned before, instead of using normalization, standardization is used for the dataset in this paper. Thus, equation (5) should be modified as well. Here is the new formula for equation (5), which is used in model I.

$$\text{where } sact\ (p,h) = pattern\ (h) \tag{6}$$

For those angels, if one angle between two input fields is close to 90 degrees, it indicates that these two inputs are less similar to each other. Small angles (< 15 degrees) indicates big similarity and too large angles (> 165 degrees) indicates they are effectively complementary, and all input pairs with those two kinds of angles should be removed [6]. For the dataset in this paper, input pairs are sorted based on their angle distances to 90 degrees in ascending order. In that sorted list, extracting the first 5 unique inputs as the least significant inputs and the last 5 unique inputs as the most significant inputs. Those will be the outputs of model W and I; this paper will compare the performances of initial networks with the networks removed the most/least significant inputs.

For model C (aggregate of I) and U (aggregate of W), instead of sorting input pairs, they are created by sorting the average angle of each input to all the other inputs.

This technique, functional measures, is well suited for the dataset in this paper. During this model training, the training accuracy will get close to 100% but the validation/testing accuracy is never higher than 50%, which conforms to Gedeon's description, that the dataset may contain much irrelevant or redundant information [5]. Thus, by applying functional measures, many redundant data points will be removed effectively and efficiently.

2.2 Data Inspection

The size of this dataset is 576*27, which means the dataset contains 576 data points and each data point has 27 attributes. The first attribute is the participant number P1, P2 … P24, as mentioned in part 1, there are 24 participants. The last attribute is the labels of music type, 1 for classical, 2 for instrumental and 3 for pop. All the other attributes are features extracted from the frontal lobe (represented by F7) electroencephalograms, which are the Mean, Maximum, Minimum, Standard Deviation, Interquartile Range, Variance, Sum, Skewness, Kurtosis, Means of the first differences, Means of the second differences, Root Mean Square, Sum of Absolute Values, Simple Square Integral, Variance of Absolute Values, Means of the absolute values of the first and second differences, Log Detector, Average Amplitude Change, Difference Absolute Standard Deviation Value, Detrended Fluctuation Analysis, Fuzzy Entropy, Shannon's Entropy, Permutation Entropy, Hjorth Parameters and Hurst Exponent of electroencephalograms data. Thus, the first and last attribute are both nominal data represented by integers, while the other attributes are floats.

	subject no.	mean_F7	max_F7	min_F7	std_F7	iqr_F7	var_F7	sum_F7	skw_F7
count	576.000000	576.000000	576.000000	576.000000	576.000000	576.000000	576.000000	576.000000	576.000000
mean	12.500000	1.819086	6.354124	0.243775	1.900527	1.902208	12.188467	24.834008	0.857294
std	6.928203	2.570039	9.783731	0.763617	2.931105	2.934445	33.388805	37.050789	0.772770
min	1.000000	0.177153	0.331130	0.000036	0.058098	0.022917	0.003375	1.311274	-1.368202
25%	6.750000	0.489684	1.227658	0.005698	0.363696	0.514053	0.132275	6.050474	0.320250
50%	12.500000	0.814839	2.014539	0.015316	0.580049	0.807612	0.336458	9.727701	0.773900
75%	18.250000	1.632414	6.132998	0.230895	1.885264	1.514906	3.554224	23.708503	1.421059
max	24.000000	21.310083	49.653301	14.591949	16.243698	22.097017	263.857719	230.562023	2.390834

Fig. 1. A part of summary statistics of the data, generated by function "describe()"

As shown in Fig. 1, by comparing the maximum value of mean_F7 and var_F7, the range of different inputs varies a lot, which indicates that some pre-processes, such as normalization might be applied to make it easier for model training.

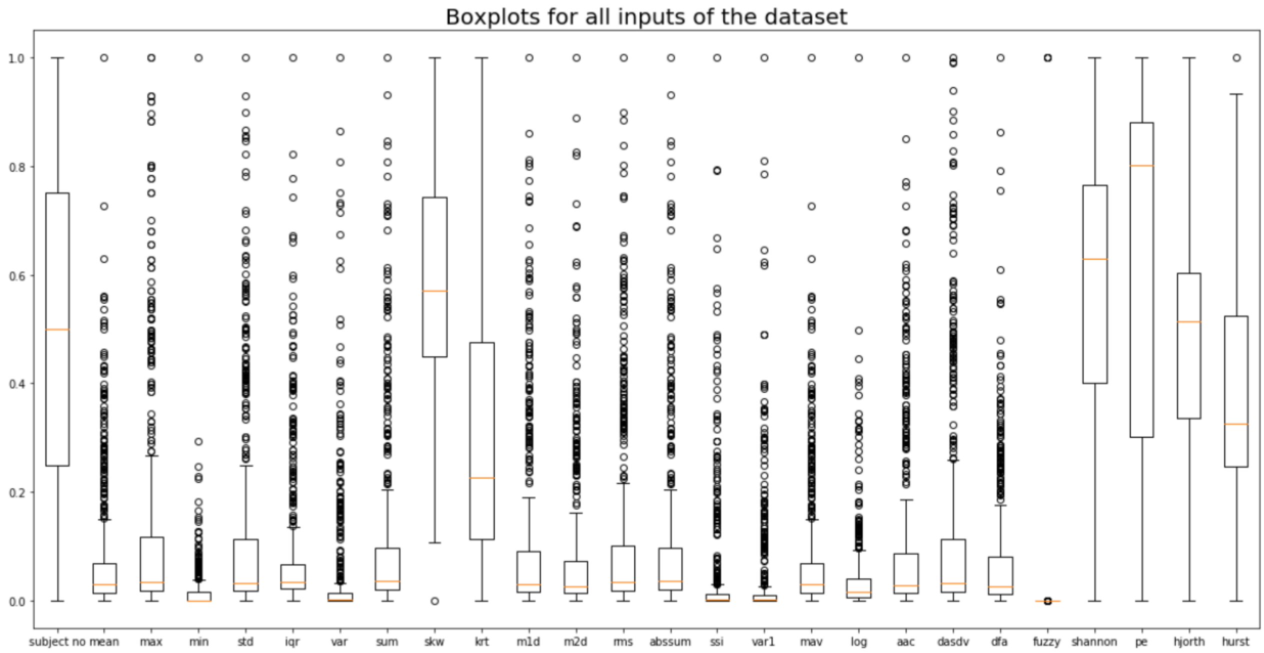


Fig. 2. Boxplots for showing the distributions of each input (first 26 attributes) of this dataset after basic pre-processing steps (normalizing numeric variables)

As shown in Fig. 2, most of the inputs have many outliers, which indicates that it is not suitable to squash data to range 0 – 1. Moreover, for the model I and C, when applying Functional measures on inputs themselves, normalized inputs with many outliers are more likely to have different vectors. In this case, angels between inputs will all be similar, and it will be hard to determine the significant inputs. Thus, to apply Functional measures, specifically tailored pre-processing steps to the dataset should be applied to get a model with better performances.

2.3 Data Preparation

First, for the target attribute "label", its value is replaced from "1, 2, 3" to "0, 1, 2" to enable using the Cross-Entropy Loss function in the network for classification. Each music genre has the same number of data points, which is 192.

Then, all feature attributes (all attributes expect "label") are standardized by subtracting the mean value and divided by the standard deviation of that attribute. As shown in Fig.3, this pre-process ensures that around half of the data value will be positive and around half of the data value will be negative, which can directly apply functional measures now (for the model I and C).

Moreover, the attribute "subject no." (participant number) is nominal data, but in this paper, no further pre-processes are applied to modify this input. The reason is that to deal with nominal data, a common way is to convert it into several columns using one-hot representation. In that case, there will be more input fields and those new input vectors will be different from each other absolutely, which is not good for applying Functional measures.

Also, this dataset is a subset, and the results will be compared with the whole dataset in further work, so the attributes' names are not changed and all "F7" are kept, which will make it easier for further analysis.

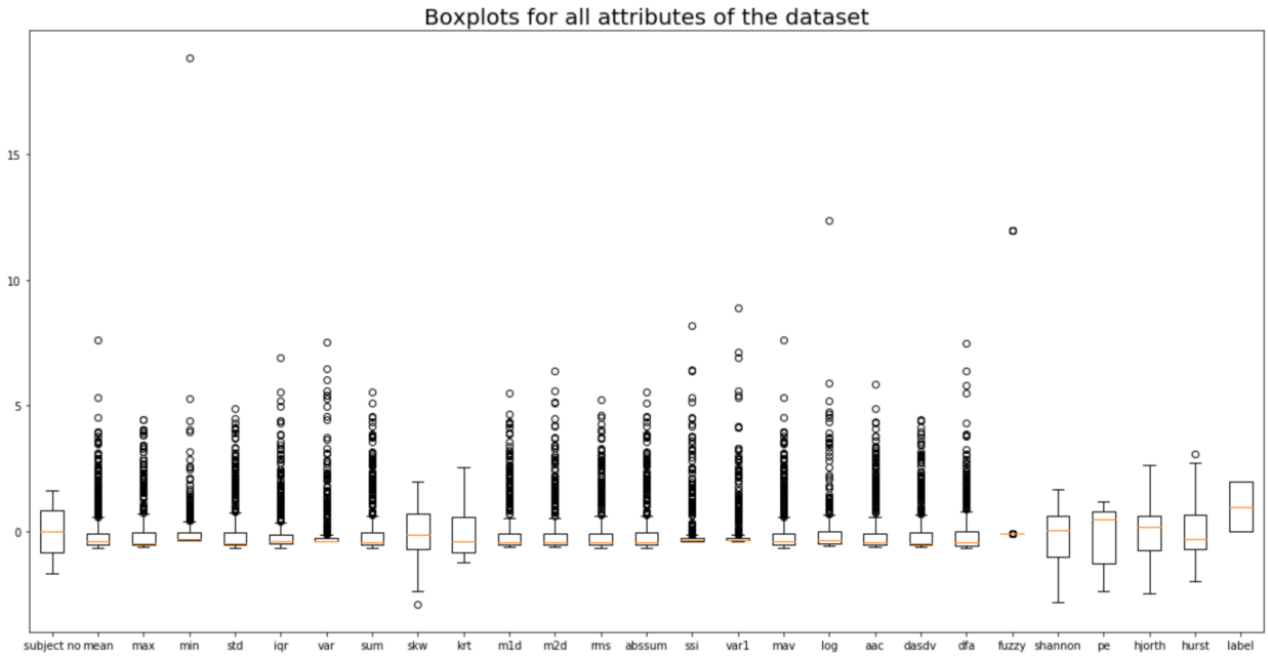
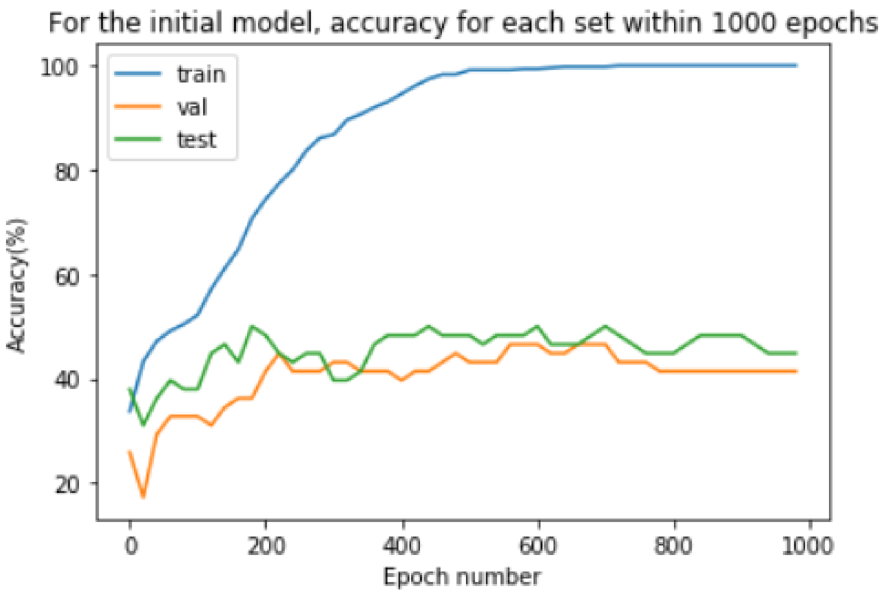


Fig. 3. Boxplots for showing the distributions of each attribute of this dataset after specifically tailored pre-processing, the order of attributes is the same as the original dataset.

As shown in Fig. 3, after specifically tailored pre-processes, all attributes are prepared and ready for building networks.

3 Results and Discussion

As mentioned in part 1 (introduction), the initial dataset is spitted into the training, validation, and testing set in the ratio of 8:1:1 using two train_test_split functions. Neural Networks will be trained on the training set only. As shown in Fig. 4, by scanning through Networks in the first 1000 epochs, the Network with the highest validation accuracy is chosen and the Network can also be judged based on the corresponding testing accuracy. All these steps ensure that the evaluation is fair, valid and all of the available data has been used for training and evaluation.




```
Epoch 561 reaches the best validation score, which is 46.55172413793103 %.
The corresponding training score is 99.1304347826087 %.
The corresponding testing score is 48.275862068965516 %.
```

Fig. 4. Accuracy for each set within 1000 epochs. During the first 1000 epochs when training a neural network, find the network with the highest validation accuracy, together with its training and testing accuracy.

To compare the created models, the program will run 10 times and the result of each run will be recorded to generate an average result. Here is a table showing the average accuracy of each model mentioned in this paper compared with the results provided by Rahman et. al. [1].

Table 1. Average accuracy (%, rounded to 2 decimals) of models for 10 runs compared with the results provided by Rahman et. al. [1]. For the Network Type, (least) means that it is a model after pruning the 5 least significant inputs, while (most) means that it is a model after pruning the 5 most significant inputs.

Network Type	Training Accuracy	Validation Accuracy	Testing Accuracy
Initial Network	76.59	46.55	38.45
Model I (least)	80.15	47.93	37.76
Model I (most)	76.54	48.10	38.45
Model C (least)	77.13	50.69	36.38
Model C (most)	51.67	41.55	37.76
Model W (least)	74.24	48.10	38.45
Model W (most)	63.52	46.90	35.69
Model U (least)	73.30	48.62	39.48
Model U (most)	90.93	46.90	40.34
The result from Rahman et. al. [1]	97.50		

From Table 1, the accuracy from Rahman et. al. [1] is high (97.50%) compared with models in this paper. The reason is that the dataset used in this paper is a subset of the model from Rahman et. al. and it may not contain enough information to build a model with high accuracy.

For these three accuracies, the training accuracy indicates the probability of model overfitting and the testing accuracy is an additional evaluation method, while validation accuracy should be paid with more attention. By comparing the validation accuracy of each model, for models removed the 5 least significant inputs, the model performance does not drop a lot (around 47%), which indicates that Functional measures [5] is a suitable technique for removing redundant information, and it conforms to this paper's stated goals. While for models removed the 5 most significant inputs, only model C shows a clear bad effect on the model performance (from 47% to 42%), which indicates that Functional measures [5] still has some limitations.

4 Conclusion and Future Work

In this paper, Neural Networks are built to classify the music types based on features extracted from the frontal lobe electroencephalograms and the Functional measures technique has been used to remove irrelevant inputs. From the results of part 3, the training score is high (can reach up to 100%), but the validation/testing score is low (the average validation accuracy is 47%), which is much lower than Rahman et. al.'s result (97.50%) [1]. Compared with Rahman et. al.'s model, networks in the table only use data from the frontal lobe part instead of the whole brain and the information in the dataset is less and redundant, so this result is accepted.

To remove redundant information, the Functional measures technique is a good method [5]. Modifications on both dataset and Functional measures technique have been applied, which are described in part 2.1 and 2.3 to ensure that the technique will run appropriately on the dataset. By analyzing the results of those 4 models using Functional measures, this paper finds that Functional measures can be used to remove the least significant inputs and work effectively, as model performance will not be influenced. But this technique also has limitations when it is used to find the most significant input. Among those 4 models, only the results of model C shows that the model performance has a relatively significant reduction (5%). This evidence indicates that Functional measures cannot find the most significant inputs stably on the dataset with less information.

For future work, Functional measures should be applied to the whole dataset instead of the subset to further check its performance on finding the most significant inputs. Moreover, for model W and U, they will sort inputs based on the existing weight matrix, so if the reliability of the current weight matrix is low and cannot provide models with a high validation/testing score, the effect of Functional measures will be small as well. More efforts need to be taken to minimize the negative effect from models providing unsuitable weight matrixs when using the Functional measures technique.

References

1. Rahman, J. R., Gedeon, T., Caldwell, S., Jones, R.: Brain Melody Informatics: Analysing Effects of Music on Brainwave Patterns. In 2020 International Joint Conference on Neural Networks (IJCNN). 1--8 (2020)
2. McCraty, R., Barrios-Choplin, B., Atkinson, M., Tomasino, D.: The effects of different types of music on mood, tension, and mental clarity. Institute of HeartMath, Boulder Creek, Calif., USA (1998)
3. Das, P., Gupta, S., Neogi, B.: Measurement of effect of music on human brain and consequent impact on attentiveness and concentration during reading. Procedia computer science. 172, 1033--1038 (2020)
4. Rumelhart, D. E., Hinton, G. E., Williams, R. J.: Learning internal representations by error propagation. Parallel distributed processing: explorations in the microstructure of cognition. 1, 318--362 (1986)
5. Gedeon, T. D.: DATA MINING OF INPUTS: ANALYSING MAGNITUDE AND FUNCTIONAL MEASURES. International Journal of Neural Systems. 8(2), 209--218 (1997)
6. Gedeon, T. D., Harris, D.: Network Reduction Techniques. Proceedings International Conference on Neural Networks Methodologies and Applications, AMSE. 1, 119--126 (1991)
7. Gedeon, T. D.: Indicators of Hidden Neuron Functionality: Static versus Dynamic Assessment. Australasian Journal of Intelligent Information Systems, invited paper. 1--10 (1996)