Investigating the Suitability of Low-Cost EEG Hardware for Consumer Devices

Mitchell Bagot

Research School of Computer Science. Australian National University
u6940504@anu.edu.au

Abstract. Virtual reality devices have taken enormous strides in recent years, with many new interface features becoming commonplace in consumer-level devices. This paper investigates the potential of four neural architectures for classifying EEG data, using a single EEG tracking point, to determine the plausibility of introducing low-cost EEG hardware interfaces into new VR HMDs. Consideration is given to the quality of classification from both raw data, and statistically computed features. Throughout this investigation, dynamic auto-constructive architectures, such as the Cascade Correlation architecture, are found to be highly effective at classification when statistical features are used as input. Neural features, as generated by an LSTM, are found to be more accurately classified for all architectures, but achieve best classification with a simple feedforward network. This work demonstrates the incredible potential of low-overhead neural models at working with low-cost EEG hardware, and that this technology has considerable potential in consumer-level devices.

Keywords: Virtual Reality, Classification, Electroencephalogram, Music Classification

1 Introduction

Modern interface devices are consistently evolving with many new features integrated into each successive generation. Along with these improvements often comes a reduction in size and weight, which occasionally yields a simplification of hardware. In the case of Virtual Reality hardware, simple, but effective electroencephalography (EEG) hardware integrated into consumer level devices could yield incredible potential for the fields of both gaming [7, 10] and medical rehabilitation [6, 8, 9].

However, despite this potential, most EEG-based systems rely on many EEG points [1, 6, 7, 9, 10], which dramatically increases the price and computational complexity of implementing a brain-computer interface (BCI), beyond that acceptable for consumer-level hardware. These costs could be brought down drastically if new models could be developed to function with a minimal number of EEG data points.

To this end, this paper aims to investigate various simple neural models for the task of music genre classification, in a similar vein to previous works [1, 8, 9], but making use of only a single EEG point. Although this task may not see significant use in most consumer-level VR BCIs, it functions effectively as a proof-of-concept for the use of neural models to process reduced EEG signals.

This paper explores the use of four different models on an EEG timeseries dataset [1]. The dataset provides band power values computed at periodic steps from 128Hz EEG data, collected from 14 points on the head. The use of raw timeseries power values allows for both statistical features, and neural features to be be experimented with throughout training. Furthermore, due to the separation of the data by point, features specific to certain areas of the brain can be used easily to demonstrate the models' efficacy with only a single EEG point.

The four types of models explored in this work include a feedforward shallow neural network, a Cascade Correlation network [3], a Local Feature Constructive Cascade network [2] and a convolutional neural network. Each network has two variants, once which uses fixed statistical features, as in the original paper [1], and a second which makes use of a simple bidirectional LSTM model to process the values into neural feature vectors. For fixed statistical inputs, auto-constructive cascade models were found to far outperform the fixed structure models, yet feedforward networks performed vastly better when neural features were used, as shown in Section 3.

2 Method

2.1 Data Set

The dataset used is a modified subset of that originally developed by Rahman et al. [1]. The full dataset consists of a timeseries of power values, collected into 8 bands, from a total of 14 points on each of the 24 subjects' heads. Each subject listened to 8 songs total, 4 from each of 2 categories. This yielded a dataset with a total of 2,688 timeseries vectors. Although the original dataset contained raw EEG values, the paper also makes use of 26 statistical feature values for classification [1], of which a subset is taken as input to the classifiers.

For this task, a subset of the original dataset was taken, consisting only of data from the F7 EEG point, located on the far left side of the subject's frontal lobe. According to the results by Rahman et al. [1], data from this point, alongside F3, was found to have the greatest correlation with accurate classification results. Data from this point alone is used for both the neural feature networks, and for the statistical feature networks.

Some minor transformations were applied to the dataset, including both the statistical features, and the raw timeseries values. Two features, the Absolute Summation, and the Summation, were found to have the same value for all entries in the dataset. To reduce redundancy, the Summation feature was removed. Additionally, the Simple Squared Integral and Absolute Summation features were found to often have values that far exceeded those of the other features. To improve convergence performance, these features were converted to the square root of their original values. Lastly, all feature vectors were normalised to ensure that no large outlying features would flood the inputs.

With these changes, the final list of features for the new subset is listed in Table 1.

Table 1. List of features taken as a subset from the original dataset.

Type	Names
Linear	Mean, Maximum, Minimum, Standard Deviation, Interquartile Range, Root of the Absolute Sum, Variance, Skewness, Kurtosis, Root Mean Square, Average of the power of signals, Peaks in Periodic Signals, Integrated Signals, Root of the Simple Square Integral, Means of the absolute values of the first and second differences, Log Detector, Average Amplitude Change, Difference Absolute Standard Deviation Value
Non-Linear	Detrended Fluctuation Analysis, Approximate Entropy, Fuzzy Entropy, Shannon's Entropy, Permutation Entropy, Hjorth Parameters, Hurst Exponent

The data input for the networks differs depending on whether neural features or statistical features are to be used for that particular network. For the neural feature variants, all four networks have the same input format, consisting of a timeseries vector of vectors, each of length 8. This yields a matrix, seq_lenx8 in size, where seq_len is the length of the timeseries sequence. The LSTM encoder uses this matrix to calculate features which then match the shape of the expected statistical feature inputs for that network.

For the statistical feature networks, the data input for each varies slightly, but still results in the same data being input overall. For the CasCor and feedforward networks, which make use of 1D input vectors, the three bands of features are reshaped into a single vector of 72 values, and are concatenated to the one-hot subject vector, which provides a single input vector of 96 values total.

Since both the convolutional network and the LoCC process the data as a 2D grid, the data is passed in as a 48x3 vector, with the one-hot subject vector concatenated to each of the three 24 feature vectors. Within each of those networks, the repeated subject vectors are split from the tensor and processed separately, so as not to cause confusion in the convolution filter patterns, which aim to process the features only.

2.2 Neural Model Structures

As previously mentioned, this paper aims to explore the efficacy of four classes of neural models as methods of classifying music genres. Each of these models is implemented simplistically, since the goal is to generate lightweight models if possible. For all of the following models and structures, the tanh activation function is used.

Feedforward Neural Network. The first of the four models is the traditional feedforward neural network. This model consists of a number of inputs, which utilise a chain of linear matrix multiplications to transform the inputs into a number of outputs. For complex tasks, activation functions can be used to introduce non-linearity between the inputs and outputs, and allow for more complex functions to be represented.

For this classification task, two slightly different structures were used. The first of these utilises a bidirectional LSTM to process the raw timeseries values, and passes the calculated features into a single-layer feedforward network. The feedforward section consists of 48 inputs, a single hidden layer of 10 nodes, and 3 outputs nodes, loosely based on the structure used by Haggblade et al. [4]. Activation was applied to the outputs of only the middle hidden layer. As mentioned, the LSTM consists of a bidirectional LSTM, with an input size of 8 (to match the timeseries element vector length), and a hidden state size of 24 (which yields an output containing 48 features).

The second structure only takes in the statistical features of the EEG data, and so forsakes the LSTM encoder in favour of a slightly larger input layer. In particular, the second network variant consists of 96 inputs, a single hidden layer of 10 nodes, and 3 output nodes. As with the first structure, activation was applied to the outputs of the middle hidden layer only.

Convolutional Neural Network. Convolutional neural networks are widely regarded for their ability to recognise patterns within multidimensional data. The dataset makes use of three bands of data, each containing multiple features. It is thus presumed that a convolutional neural network is expected to provide equal or greater performance than the feedforward networks described above, whilst requiring fewer parameters and less computational power due to the sparsity of filters.

To test this hypothesis, two structures were proposed for the convolutional models, with each structure having two variants depending on the input data used. The first of these is a network based on 1D convolutional filters, operating over the three bands as input channels to the network. Two layers are used, the first with 3 input channels, 12 output channels, a filter length of 6, and stride of 2. The second layer uses the same filter sizes and stride, with 12 input channels, and 36 output channels. Each of these convolution layers is proceeded by an activation layer. To classify the output, average pooling, followed by a two layer feedforward network is used to map the features of the convolution filters into 3 output classes. This network structure is used directly to process the statistical feature matrices as described in Section 2.1, but is also used as a backbone for a second network variant which uses an LSTM encoder to encode timeseries data into a feature matrix, which is then passed through the same convolutional filters. This LSTM encoder is identical to that used by the feedforward network above.

The second structure considers the input data as a single channel set of 2D data (rather than multi-channel 1D data), and to this end uses a similar structure, but with 2D filters replacing the 1D filters of the previous structure. The first of the two layers uses a 6x2 filter, with a stride of 2x1, padding of 0x1, with 1 input channel, and 4 output channels. The second uses the same size filter and stride, removes all padding, and has 4 input channels, to 12 output channels. As with the first structure, activations follow each convolutional layer, as well as an average pooling layer, although only on the x-dimension of the data. This all feeds into a two-layer feedforward network of the same layout as with the first model structure. Just as with the first CNN structure, this structure also is used for two variants, one using an LSTM encoder, and one which takes the statistical features directly.

Cascade Correlation Neural Network. The Cascade Correlation learning architecture, originally described by Fahlman [3], offers an effective means of generating neural network structures at training time to efficiently satisfy the problem function. As a neural model, a simple CasCor model will differ with each run, but uses fixed blocks from which to generate the complete structure of the final model. For this task, much of the network architecture is implemented as in the original paper.

Following the process laid out by Fahlman [3], the CasCor model begins with 72 feature inputs, and 24 one-hot subject inputs. These inputs are connected to the 3 outputs through a fully connected weight layer, with no initial hidden layers. As the network structure is built up, hidden cascade units are added, each of which connects to the 96 inputs, as well as all prior hidden units. Additionally, each hidden cascade unit has an activation function applied to it's output value before it is connected to the 3 outputs or to subsequent hidden units. This allows for each hidden unit to form a simple feature detector, and to recognise both linear and non-linear features in the data, or in the prior detected features, ideally yielding an effective network layout for classification of the EEG data.

Just as with the CNN and feedforward networks, a second variant which makes use of an LSTM encoder is also implemented to test the efficacy of neural features vs statistical features. Since the outputs of the LSTM do not include an explicit subject encoding, this second CasCor model is only instantiated with 72 feature inputs, connected to 3 outputs. Aside from the inclusion of the LSTM and the number of inputs, all other details remain the same as the statistical feature CasCor model.

Local Feature Constructive Cascade. Extending the Cascade Correlation model outlined above, a Local Feature Constructive Cascade network [2] is also investigated as a potential classification model. As with the Cascade Correlation model, this is a self-constructing neural network that is built at training time, rather than using a pre-specified layout. As such, the exact structure differs slightly with each run. However, the cascade and hidden layers used remain fixed. Similarly to the CasCor, this network architecture has two variants, though the underlying the LoCC structure is identical, with the only difference being the inclusion of the LSTM encoder for one of the two variants.

It's important to note that unlike the original paper's task, the 2D input to this network forms a very short but wide shape, and cannot be easily padded to a square. Thus, to account for this, the hidden layer has been reshaped to an 8x3 layer of neurons, rather than an 8x8 as used in the paper [2], and the input layer is formed as a 24x3 grid of inputs, rather than the 32x32 grid of the paper. The cascade layer dimensions remain the same.

To account for these changes to the hidden and input layers, different sizes of convolution filters (or "receptive fields") can be used to feed the neurons of the cascade and hidden layers. In particular, this classification model uses 4x1 rectangles to feed the hidden layer neurons from the input layer, with a stride of 3x1, meaning no rows overlap, and 1 column overlaps with each step of the filter.

For the cascade layers, a 8x2 filter with a stride of 5x1 takes values from the input layer, whilst a 3x2 filter with a stride of 2x1 takes values from the hidden layer. Both cascade layer filters use a single row of zero padding, with the hidden layer also padding an additional column with zeros. As with the original paper, each cascade layer has a one-to-one weighted connection with each previous cascade layer [2].

In order to aid with the non-linearity of the model, activation is applied to the output of the hidden layer values, and to each cascade layer's output.

2.3 Training Methodology

The training methods used for all four networks are largely identical. K-fold cross validation is used on the dataset, with a K value of 6 chosen to offer a good balance between training and test set sizes. All weights for all networks are initialised in the respective default manner as implemented by the PyTorch library, and are backpropagated using the autograd library included with PyTorch.

All networks, including the candidate cascade nodes and layers, are trained using RPROP, as in the original paper [2], as it was found from informal experimentation to yield the best classification results. The RPROP optimisation algorithm begins with the default learning rate as specified in the PyTorch library.

For each K-fold model, the feedforward and convolutional networks are trained for 100 epochs, with a learning rate scheduler halving the learning rate every 30 epochs. Similarly, the CasCor and LoCC models are trained for 100 epochs, before adding a new node or layer. The candidate layer/node is trained for 200 epochs before being added, after which the entire model is again trained for 100 epochs. This repeats until the accuracy no longer improves, or until the network consists of 20 cascade nodes/layers.

For all training, cross-entropy loss is used, to ensure that the loss function accurately matches the classification task being trained for. All networks finish with 3 outputs, one for each genre of music that can be classified for.

Lastly, for the feedforward and convolutional networks, dropout layers [5] are used after each activation to reduce the overfitting which is common for such small datasets as is used here.

3 Results and Discussion

As outlined in Section 2.3, all models are trained using 6-fold cross validation, with four metrics of performance used:

- Accuracy (Percentage of correctly classified inputs)
- Precision (Fraction of the predicted labels matched)
- Recall (True Positive Rate)
- F-measure (Harmonic mean of Precision and Recall)

These four metrics of performance are derived from those used in the original dataset paper [1], and are used to allow for these results to be directly compared to those achieved with the full dataset.

The feedforward and convolutional network results are reported based on the best of five runs, whilst the CasCor and LoCC networks are each run 10 times, with the best performing metrics recorded for each. The results of these experiments are recorded in Table 2, and included alongside the results of the original dataset [1].

Table 2. Performance metrics of the four models, and the original dataset paper's NN model. Higher is better, best scores in bold.

Model	Variant	Accuracy	Precision	Recall	F-measure
Feedforward	96-10-a-3	0.4531	0.4688	0.4403	0.4541
	LSTM 48-10-a-3	0.5936	0.6087	**0.6424**	**0.6251**
CNN	1D Filters	0.2865	0.2981	0.2813	0.2895
	1D Filters w/ LSTM	0.5885	0.5816	0.5752	0.5784
	2D Filters	0.3490	0.3639	0.3549	0.3593
	2D Filters w/ LSTM	**0.5990**	**0.6142**	0.5950	0.6045
CasCor	CasCor	0.5417	0.5398	0.5695	0.5543
	CasCor w/ LSTM	0.5625	0.5696	0.5736	0.5716
LoCC	LoCC	0.4844	0.4958	0.4614	0.4780
	LoCC w/ LSTM	0.5208	0.5546	0.5749	0.5646
RSFS NN [1]		0.9755	0.9615	0.9633	0.9624

Based on the above results, it is immediately apparent that the neural features generated by the LSTM encoder are far superior at producing meaningful classification than the statistical features manually generated from the same data. For all four architectures, the LSTM variants outperform the base version, some by more than others. In particular, the feedforward and CNN networks experience a significant improvement, of up to 30% gain in accuracy. It is likely that the improvement for the CNNs stems from the fact that the LSTM is able to be trained to generate features with meaningful positional information for the CNN, rather than simply stacks or channels of statistical features. This allows the CNN to make better use of the extracted features, dramatically improving performance.

The feedforward and 2D CNN match very closely in all metrics, with the CNN outperforming in accuracy and precision, and the feedforward excelling with recall and F-measure. Given the slight edge in computational performance from the feedforward, it is likely more suitable to use in virtual reality systems than the 2D CNN, although either network would be well suited.

Despite the excellent perfomance from networks using neural feature encoders, the networks making use of only statistical features still struggle, aside from the auto-constructive networks. The convolutional networks in particular have incredibly poor performance, performing worse than random chance in the case of the 1D filter variant. The use of 2D filters slightly improves the performance above randomness, but is still vastly outmatched by all other models. The feedforward statistical networks perform better, but still do not match the CasCor and LoCC models.

Of the two auto-constructive cascade networks, the CasCor network performs the best in all metrics. This matches the results of the pattern from the feedforward and convolutional statistical networks, indicating that the data inputs used are not well suited to analysis over dimensional space, and thus perform poorly with convolutional networks. Since LoCC makes heavy use of convolution filters [2], this yields poor performance compared to the individual cascade units of the CasCor network.

4 Conclusion and Future Work

This paper has investigated four different neural architectures as potential options for the classification of music genres based on EEG features. Each model was trained to classify music into one of three genres based on only a single EEG point, with a variant operating on the raw data, as well as statistical features. This work was used as an initial experiment to determine the potential suitability for small-scale EEG hardware in consumer-level products to work as a BCI interface, in such fields as gaming and medicine. Through this testing, it was found that small-scale feedforward networks, and in particular auto-constructive cascade networks, such as in Fahlman’s work [3], were best able to classify the EEG features, and thus have considerable potential for future development alongside low-cost EEG hardware. Furthermore, the use of LSTM encoders on raw EEG data was found to offer significant improvement to classification accuracy, at a cost of computational throughput.

This work demonstrates that through neural models, great improvements can be made in the processing of signals from small and cheap hardware, and thus introduces multiple new prospects for future work, including:

- Emotional classification with minimal points, to allow for procedural modification to video games according to a player’s emotion, or potentially for automatic gameplay feedback telemetry.
- General movement prediction, potentially allowing for optimisations in video game rendering, or as part of lower cost EEG-based artificial limbs.
- Mental sentiment detection, which could grant video games the ability to determine intent behind actions, and integrate this into assistance or training sub-systems.

It’s important to note that although this work introduces the potential for neural processing of reduced EEG signals, many more developments are necessary to ensure that this technology would be satisfactorily capable, especially in medical rehabilitation. In particular, this work relies on either expensive sequential LSTM neural processing of high resolution signals, or generalised statistical analysis. Further work could be done to investigate the suitability of different models to generate neural features, which could yield more suitable features, or produce similar results, with far less overhead.

Nevertheless, this study demonstrates that with the right computational models, low-cost EEG hardware has significant potential for future growth and use in consumer-grade devices and use-cases.

References

1. Rahman, J., 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), 2020, pp. 1-8. (2020)
2. Khoo, S., Gedeon. T.: Generalisation Performance vs. Architecture Variations in Constructive Cascade Networks. In: Köppen M., Kasabov N., Coghill G. (eds) Advances in Neuro-Information Processing. ICONIP (2009)
3. Fahlman, S., Liebiere, C.: The Cascade-Correlation Learning Architecture. Technical report, School of Computer Science, Carnegie Mellon University (1990)
4. Haggblade, M., Hong, Y., Kao, K.: Music Genre Classification. Technical report, Stanford University.
5. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. Technical report, Department of Computer Science, University of Toronto. (2012)

6. W.-E. Wang et al.: A Novel Method to Understand Neural Oscillations During Full-Body Reaching: A Combined EEG and 3D Virtual Reality Study. In: IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 28, no. 12, pp. 3074-3082.(2020)
7. Kerous, B., Skola, F. & Liarokapis, F.: EEG-based BCI and video games: a progress report. In: Virtual Reality vol. 22, pp. 119–135. (2018)
8. J. Marín-Morales et al.: Affective computing in virtual reality: emotion recognition from brain and heartbeat dynamics using wearable sensors. In: Scientific Reports (Nature Publisher Group), vol. 8, pp. 1-15. (2018)
9. Suhaimi, N.S., Mountstephens, J., Teo, J.: EEG-Based Emotion Recognition: A State-of-the-Art Review of Current Trends and Opportunities. In: Computational Intelligence and Neuroscience, vol. 2020, Article ID 8875426, 19 pages. (2020)
10. Brouwer, A., d. W. Jasper van, Stokking, H.: BCI to Potentially Enhance Streaming Images to a VR Headset by Predicting Head Rotation. In: Frontiers in Human Neuroscience. (2018)