An Evaluation of the State of the Art CNN and RNN Models for the Classification of EEG Data

Kunal Patel	Priscilla Cheng	Nathan Pilbrough
UCLA	UCLA	UCLA
704760470	404159386	904865112

{kunalpatel1793,priscillaccheng,nathanpilbrough} @ gmail . com

Abstract

The performance of several neural network architectures was evaluated for classifying EEG signals. The architectures tested were designed to take advantage of recent developments in the field and apply them to the BCI dataset. Namely, deep convolutional architectures formed a pivotal part of each model in this paper. Other structures implemented included Inception layers and deep RNN architectures to exploit the temporal nature of the EEG signals. It was found that a fully optimized RNN model outperformed other state-of-the-art CNN models.

1. Introduction

RNNs have been shown to work well on sequential data [3], while CNNs have shown remarkable results in recent years on image classification tasks [1], often achieving superhuman performance. The convolutional filter's spatial locality can mimic the temporal locality attributes of RNNs when the data is represented as a time series. Both CNNs and RNNs have been used to achieve near state-of-the-art performance [7][8] and these models were therefore chosen as starting points for further investigation in this paper. In particular the models that were implemented included two CNNs, one shallow and one deep, four RNNs of varying complexity, and lastly a spectrogram-based model, all of which are explained in more detail in the model architecture section.

2. Data Preparation

Before training, 50 samples from each subject were placed aside as the testing set. All samples were preprocessed and augmented as described above to maintain consistency. Two learning scenarios were investigated. In the first, the models were trained on the data from one subject and tested on the data from the same subject. In the second scenario, the models were trained on all the data from all subjects and subsequently tested on each subject individually.

2.1 Preprocessing

One of the main issues with EEG data is their low signal to noise ratio. To assist signal classification, the dataset was preprocessed in a similar fashion to [7]. More concretely, the data were passed through a bandpass filter with a low frequency cutoff of 4Hz and a high frequency cutoff of 38Hz. This frequency range encapsulates both the alpha and beta wave oscillation frequencies [9]. Furthermore, it is good practice to standardize the input data as this results in more optimal training and helps avoid exploding and vanishing gradients [13]. As such the standardized data was calculated for each electrode t as follows:

$$x_t' = (x_t - \mu_t)/\sqrt{\sigma^2}$$

2.2 Data augmentation

It is well known that neural nets observe performance metrics which are proportional to the amount of data they are trained over [1]. In the case of the EEG dataset, the amount of data provided is comparatively small, with only 288 samples per subject. Augmentation via windowed subsampling of the original data as per [7] significantly increased the size of the dataset, however the tradeoff was that the data was highly correlated, which made the models susceptible to overfitting. In addition, data augmentation forced the model to learn discriminative features across all sections of the data.

2.3 Spectrogram

As CNNs are known to be well suited for image classification, Fedjaev [10] showed that preprocessing of EEG signals into image-like spectrograms improve binary classification on CNNs. This was achieved in practice by using a Short-Time Fourier Transform (STFT). This resulted in 22 image-like features, corresponding to the 22 EEG electrodes. This step was only performed for one specific model and was not common to all models.

3. Model Architectures

A detailed description of the model architectures is given on page 5.

3.1 Shallow CNN

Input data samples to the Shallow CNN were augmented by subsampling into 150 windows of 512 points each which was significantly more windows than that used for the other architectures. The model performs two convolutions, the first in time and the second in space over the 22 electrodes. Overall the model has a structure which mimics the operations of a filter bank common spatial patterns algorithm, used in the past to successfully classify EEG signals [7].

3.2 Deep CNN

Similar to the Shallow CNN, the Deep CNN first performs two convolutions in time and space respectively. This is followed by 3 subsequent convolution layers whose number of filters double from the previous layer. Each convolution layer is then followed by a batch normalization, pooling, and dropout layer to create a convolution block. This model drew inspiration from successful computer vision architectures [14].

3.3 RNN Models

RNN models differ in structure from traditional feedforward architectures, in that each node points to all nodes ahead of it in the sequence. The training dataset for the RNN model was standardized and augmented with the default settings. The convolutional layers provide spatial locality as well as downsampling of the signal for more manageable computation. A single convolutional layer does not allow the network to learn over multiple time scales and so the second RNN model (IC-RNN) implemented Inception modules comprised of three convolutional layers. The third RNN model (C-DRNN) drew inspiration from ResNet [11] and DenseNet [12] architectures whereby skip connections were employed such that the model could be fit to less complicated data. Concretely, the output of each GRU layer was connected to the input of every other GRU layer in the forward direction.

Finally, the IC-RNN and C-DRNN models were combined to form the final model, ChronoNet. This architecture was designed to combine the best features from IC-RNN and C-DRNN and to be flexible with input data. The Inception layers of differing filter sizes allowed for feature extraction from different time scales and windows while the residual GRU layers prevent the problem of vanishing and exploding gradients that would negatively impact the training accuracy [8].

4. Results

The results from the first learning scenario for all models are shown on page 6 and those of the Shallow CNN and ChronoNet are repeated in Table 1. These two models performed the best in comparison to the other listed models.

Table 1. Performance summary of Shallow CNN and ChronoNet models

Model Name	Training Score	Validation Score	Testing Score
Shallow CNN	60.4%	71.4%	67.6%
ChronoNet	93.0%	72.0%	74.0%

Figure 1. Training and validation accuracies of the ChronoNet model

The Shallow CNN performed slightly below par in comparison to that implemented by Schirrmeister et al [7] and this is discussed in the next section. ChronoNet, however, achieved close to state of the art performance for the one-to-one learning case. Whilst this is less than the reported accuracies from the ChronoNet paper, it is noted that those accuracies were achieved for a binary classification task. A more thorough description of ChronoNet's results is given since it achieved the best performance. The training and validation accuracy for ChronoNet over the number of epochs is shown in Figure 1. The validation accuracy of the model steadily climbs at the beginning of the training session but begins to flatten out as the training accuracy climbs above 90%.

The confusion matrix for the ChronoNet model is shown in Figure 2 in the Model Performance section on page 6. As expected, the majority of the predicted labels lie on the diagonal which corresponds to correct predictions.

The results from the second learning scenario for all models are shown on page 6. Once again, the ChronoNet achieved the highest accuracies out of all the models. The scores achieved for individual subjects at test time is shown in Table 2. The mean overall test score for this case was 51.7%. The model's hyperparameters were adjusted to optimize the testing score overall. This testing score was less than the scores recorded for the individual case and this is discussed in more detail in the next section.

Table 2. ChronoNet Test Accuracy Per Subject

Subject	1	2	3	4	5	6	7	8	9
Test Acc. (%)	75.4	45.6	68.4	45.0	44.6	48.2	59.2	49.2	30.0

Table 3. ChronoNet Window size vs Testing Accuracy

Window Size	250	512	750
Testing Accuracy (%)	61.0	74.0	62.2

5. Discussion

While neural nets are often trained as end-to-end models and thus lessen the need for feature extraction, it was found that preprocessing was one of the major performance boosters for all models. This is likely because preprocessing increases the signal to noise ratio, making discriminative features easier to learn.

In general, the CNN models did not perform quite as well as they did in the original paper [7]. One reason for this is that the data in the original paper was subsampled into 625 windows and the mean of these predictions was used as the final prediction. Resource constraints, however, limited the number of windows used in this project to significantly less than this. All recurrent models were trained over 10 windows of the data with window-width 512, as this proved to be the optimal input. Table 3 suggests that window sampling is an additional hyperparameter that needs to be optimized and that there exists a high-level correlation between the time window and the model performance.

The Shallow CNN performed best on a single subject with vastly more data window subsamples than the other models (150 vs 10), an interesting result because the intensive windowing indicated that the data were highly correlated and this in turn lead to very short training times with respect to the number of epochs. Thus, the model was able to train over multiple copies of the data all within the same epoch, which is analogous to training less intensively subsampled data over multiple epochs.

The Deep CNN achieved a lower accuracy for the single subject test than the Shallow CNN, which matches the results obtained from the original paper [7]. This can most likely be attributed to an overly complicated model structure without skip connections such that features in the layers can be missed altogether. A high testing accuracy was achieved when training over 45 epochs, which is significantly more than the Shallow CNN model due to less intensive subsampling (10 vs 150).

Using spectrograms did not boost the model's accuracy. Although it may seem intuitive that analyzing the power density of the electrode waveforms over time and frequency would allow the Deep CNN to hone in on features specific to each class, this was not the case. One possible reason for this is that in the original paper [10], the authors only used 3 electrodes and as such, the input was more analogous to 3 channel RGB images for which CNNs have been optimized. Furthermore, the classification task in that paper was simply binary classification of left or right hand movement.

The simplicity of the C-RNN architecture for the given dataset is what may have allowed it to be so successful. As described earlier, the convolution layers serve to reduce dimensionality while interpreting time adjacent features, prior to being passed in to RNN type layers for further analysis. It is for this reason that the architecture may have been successful for the given dataset.

The IC-RNN structure did not perform as well for this dataset as other models did. The ChronoNet paper utilized a different dataset than this study, where their task was binary with 15000 time measurements over a 1 minute sample. For such a large time dimension, a series of Inception layers makes sense as there is more information to sweep. One can also make the same argument for the C-DRNN architecture, as the two models achieve similar performance.

In general, the models did not perform as well on the second learning scenario as they did in the first learning scenario. Certain trends did present themselves however. Looking more closely at the data in Table 4, it can be seen that some datasets result in consistently better accuracies than others. In particular datasets 1 and 3 yielded high accuracies whilst 2 and 9 yielded low accuracies consistently. These observations are examples of higher level interpretations that result from this study. Subjects 1 and 3 may have similar signal data as well as subjects 2 and 9 if they indeed behave consistently across all models tested. For ensemble methods in future work, independent subjects can be grouped based on how well they adhere to the types of architectures studied here, allowing for a better understanding of neural activity in the brain.

6. Conclusion and Future Work

In conclusion, the ChronoNet architecture has been shown to outperform traditional CNN architectures in multiclass classification. This is state-of-the-art research because the ChronoNet paper was only applied to binary classification on a different dataset than the BCI dataset. Furthermore, this proves that combining structures from other state-of-the-art architectures such as ResNet and GoogleNet can create new architectures suitable for a different application. Future research would involve studying the relationship between the time sampling and the ChronoNet architecture. The research conducted in the ChronoNet paper involved 15000 time samples for 1 minute, which is significantly more than the 1000 samples that are used in this paper and could help improve the performance of the ChronoNet. Manipulating the data with increased time sampling was not conducted here due to computational constraints.

References

[1] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), pp.211-252.

[2] Subasi, A. and Ercelebi, E., 2005. Classification of EEG signals using neural network and logistic regression. *Computer methods and programs in biomedicine*, *78*(2), pp.87-99.

[3] Lipton, Z.C., Berkowitz, J. and Elkan, C., 2015. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

[4] Oord, A.V.D., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G.V.D., Lockhart, E., Cobo, L.C., Stimberg, F. and Casagrande, N., 2017. Parallel WaveNet: Fast High-Fidelity Speech Synthesis. *arXiv preprint arXiv:1711.10433*.

[5] Pete Warden. TensorFlow Speech Recognition Challenge | Kaggle. Link: https://www.kaggle.com/c/tensorflow-speech-recognition-challenge, 2017.

[6] Bai, S., Kolter, J.Z. and Koltun, V., 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv preprint arXiv:1803.01271*.

[7] Schirrmeister, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W. and Ball, T., 2017. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human brain mapping*, *38*(11), pp.5391-5420.

[8] Roy, S., Kiral-Kornek, I. and Harrer, S., 2018. ChronoNet: A Deep Recurrent Neural Network for Abnormal EEG Identification. *arXiv preprint arXiv:1802.00308*.

[9] da Silva, F.L., 1991. Neural mechanisms underlying brain waves: from neural membranes to networks. *Clinical Neurophysiology*, *79*(2), pp.81-93.

[10] Fedjaev, J., Decoding EEG Brain Signals using Recurrent Neural Networks.

[11] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A., 2017, February. Inception-v4, Inception-resnet and the impact of residual connections on learning. In *AAAI* (Vol. 4, p. 12).

[12] Huang, G., Liu, Z., Weinberger, K.Q. and van der Maaten, L., 2017, July. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (Vol. 1, No. 2, p. 3).

[13] Pascanu, R., Mikolov, T. and Bengio, Y., 2012. Understanding the exploding gradient problem. CoRR, abs/1211.5063.

[14] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[15] Brunner, C., Leeb, R., Müller-Putz, G., Schlögl, A. and Pfurtscheller, G., 2008. BCI Competition 2008–Graz data set A. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, *16*.

Model Architectures

Shallow CNN - The Shallow CNN architecture is described as follows. All weights and biases are initialized using Xavier initialization and Adam used as the optimizer. Inputs were standardized and augmented by dividing each sample into 150 windows of 512 steps each.

[Conv2D(filters= 40, filter_size(1,25)) - Relu - Conv2D(filters= 40, filter_size(1,22)) - Relu - BatchNorm - Maxpool2D(pool_size= (60,1), stride = (15,1)) - Dropout(0.8)] – [Dense(units=4, softmax)]

Deep CNN - The DeepCNN architecture is described as follows. All weights and biases are initialized using Xavier initialization, and RMSProp is used as the optimizer. Inputs were standardized and augmented by dividing each sample into 10 windows of 512 steps each.

[[Conv2D(filters=20, filter_size=(20,1))]x2 - Relu - BatchNorm - MaxPool2D(3,1) - Dropout(0.5)]
[Conv2D(filters=40, filter_size=(10,20) - Relu - BatchNorm - MaxPool2D(3,1) - Dropout(0.5)]
[Conv2D(filters=80, filter_size=(10,40) - Relu - BatchNorm - MaxPool2D(3,1) - Dropout(0.5)]
[Conv2D(filters=160, filter_size=(10,80) - Relu - BatchNorm - MaxPool2D(3,1)] - [Dense(units=4, softmax)]

VGGnet - This is standard algorithm found throughout literature in image classification. This architecture was modified to operate with 1D convolution, and the complete architecture is described below

[[Conv1D(filters=32, filter_size=(4), stride=1) - Relu - BarchNorm]x2 - MaxPool1D(2) - Dropout(0.25)]x2 - Flatten - Dense(units=256) - Relu - Dropout(0.5) - Dense(4) – Softmax

CRNN - The issue with GRU layers is that they are very computationally expensive. As a result, this model was comprised of three ReLu activated, 1-D convolutional layers with 32 filters each, a length of 4 and stride of 2. After which the output was passed through 4 tanh activated GRU layers and finally a softmax activated dense layer with 4 nodes. The convolutional layers allow for downsampling while extracting features. This allows the GRU layers to operate efficiently within the constraints of this data.

[Conv1D(filters=32,filter_size=4, stride=2) - BatchNorm]x3 - [GRU(Units=32,return_sequences=True)-Tanh]x3 - GRU(units=32) - Tanh - Dense(4) – Softmax

IC-RNN - This architecture builds on the CRNN architecture, but utilizes the Inception layer architecture to extract information from convolutional windows of different sizes. The ICRNN uses windows of 2, 4, and 8 units with stride 2, effectively reducing the dimension of the output by approximately half. These Inception layers are batchnormalized and stacked prior to being fed into the connected GRU layers.

I = Concatenate(Conv1D(filters=32,filter_size=X,stride=2)) - BatchNorm - Relu) {X = 2,4,8}
[I - Dropout(0.5)] - [GRU(Units=32,return_sequences=True)-Tanh]x3 - GRU(units=32) - Tanh - Dense(4) – Softmax

C-DRNN - This architecture is inspired by the feedforward structure of networks such as resnet where GRU layers are connected to all GRU layers ahead of it in the sequence. The architecture is also known as a Densely Connected architecture, and is described below (DC).

X1 = Input
X2 = GRU(units=32,activation=tanh,return_sequences=True)(X1)
X3 = concatenate(X1,X2)
X4 = GRU(units=32,activation=tanh,return_sequences=True)(X3)
DC = GRU(units=32,activation=tanh,return_sequences=True)(concatenate(X1,X2,X3)
[Conv1D(filters=32,filter_size=4, stride=2) - BatchNorm]x3 - DC - Dense(4) - Softmax

ChronoNet - This is current state-of-the-art for binary classification of EEG data. The model essentially merges innovations in the current leading models in the deep learning, GoogleNet and ResNet. The Inception layers inspired by GoogleNet are fed into the densely connected structure from Resnet and scored. The simplified architecture is described below

[I - Dropout(0.45)]x3 - DC - Dense(4) - Softmax

Model performance

The tables below show the model performance for two evaluation sets. Table 4 was trained and tested on subject 1. Table 5 was trained using all the data and tested on all the data

Table 4. Scores of all models trained on all subject data

Model	Training Score (%)	Validation Score (%)	Testing Score by Subject (%)								
			1	2	3	4	5	6	7	8	9
1D VGGNet	50.6	35.3	**42.0**	26.2	**47.8**	29.6	25.8	27.6	28.8	**34.4**	28.0
Shallow CNN	41.8	42.9	**58.6**	29.2	**53.4**	38.6	23.0	31.6	39.2	**44.6**	33.6
Deep CNN	61.7	47.5	**61.6**	37.0	**63.2**	37.6	28.8	37.2	**50.6**	42.2	28.6
C-RNN	70.6	45.7	**70.0**	33.2	**63.4**	40.2	36.3	46.2	**48.2**	44.4	26.0
IC-RNN	55.9	41.5	**57.2**	34.4	**46.6**	37.6	27.4	29.8	38.4	**39.0**	30.0
C-DRNN	71.6	44.2	**50.8**	24.4	**47.4**	36.6	29.4	39.0	41.4	**43.0**	29.4
ChronoNet	78.1	56.1	**75.4**	45.6	**68.4**	45.0	44.6	48.2	**59.2**	49.2	30.0
Spectrogram Deep CNN	58.9	31.7	**41.4**	32.6	**35.0**	26.6	31.0	32.2	**36.8**	33.4	26.8

Table 5. Scores of all models trained on subject 1

Model Name	Training score (%)	Validation score (%)	Testing score (%)
1D VGGnet	92.1	49.1	47.4
Shallow CNN	60.4	71.4	67.6
Deep CNN	85.5	60.4	60.4
C-RNN	94.3	64.5	71.6
IC-RNN	78.1	56.7	58.4
C-DRNN	83.8	59.7	61.0
ChronoNet	93.0	72.0	74.0
Spectrogram DeepCNN	85.0	41.7	38.6

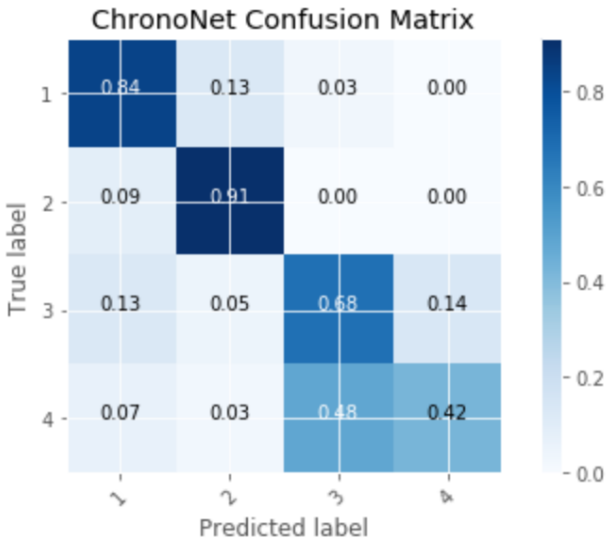

Figure 2. Confusion Matrix of the ChronoNet Model