net4nets: A Convolutional Neural Network to Build a Binary Classifier To Identify Smokers and Abstainers From 29-Electrode Electroencephalography Data

Adam E. Holbrook
aholbro@siue.edu
ID: 800525216

Zachary T. Heffron
zheffro@siue.edu
ID: 800563578

Lukas A. Moore
lumoore@siue.edu
ID: 800567590

Abstract

Recent exponential growth in the fields of computational neuroscience and machine learning have made possible novel means of collaboration between these fields. This paper outlines such a collaboration – a Convolutional Neural Network was trained on cleaned eyes-closed resting state electroencephalography (EEG) data collected from 200 light to moderate smokers in the Southern Illinois area to determine from which of two conditions a sample of EEG data originated – either a participant who had smoked prior to the session or abstained for 24 hours. It was tested on 50 similar participants, to a resultant 94% classification accuracy for this group.

1 Introduction

1.1 Background

Relatively recently, the use of electroencephalography in cognitive neuroscience research has become more and more commonplace due to both advances in computational technologies and the reduced financial burden of implementing the technology compared to alternatives like Functional Magnetic Resonance Imagery (fMRI), Magnetoencephalography (MEG) or Computed Tomography (CT). Along with cost, EEG has the benefit of being simple to implement and requiring less prior subject matter expertise compared to more sophisticated methodologies. EEG is also an important technique in the burgeoning field of Brain Computer Interfaces (BCI).

The growth of this field has coincided with a revolution in machine learning algorithms and methodologies, made especially poignant by the subsequent advances in distributed computing and architectures which lend to the construction and utilization of machine learning algorithms. The efficiency and accuracy of many of these algorithms, including those utilizing deep learning algorithms, make them a valid candidate for augmentation or replacement of older data analysis methods and algorithms in many fields, including computational neuroscience.

Our project specifically aims to contribute to the implementation of deep learning to the field of computational neuroscience. While there is some research into classification algorithms for a small number of specific EEG classification tasks, the field is largely unexplored. Our hope is that the results of this exercise will serve as a proof-of-concept by which future models will be compared.

1.2 Model Description

Our model was built from cleaned and processed data provided by the Integrative Neuroscience Laboratory at Southern Illinois University Carbondale. We will be using a two-layer convolutional

Preprint. Work in progress.

neural network to analyze inputted EEG data, in order to determine if a participant had smoked before the session or abstained.

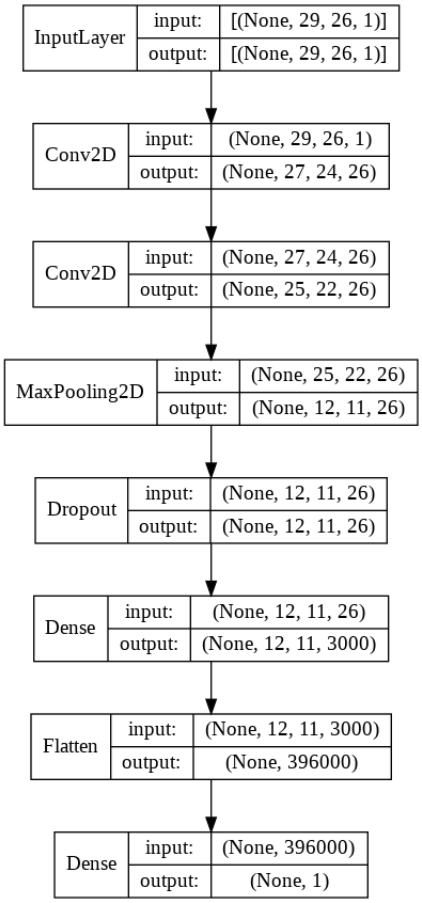

fig. 1: *A visualization of our model, which consists of the following components: a* $29x26x1$ *input layer, followed by two* 3×3 *convolution layers each outputting 26 features, followed by a* 2×2 *pooling layer, leading to a* 20% *Dropout layer, a flattening layer, and a 3000-neuron dense layer, which feeds into a sigmoidal classifier and outputs a probability between 0 and 1. It currently trains with a batch size of 1, a learning rate of 0.0001, and an Adam optimizer.*

This is the model upon which our results and figures are based, unless otherwise noted. It is, as of the writing of this article, the most recent iteration of our predictive model.

2 Related Works

2.1 Effects of Smoking on EEG Activity

Research exists in abundance regarding the effects of smoking on brain activity, which has indicated many means with which to differentiate nonsmokers from light or heavy smokers. J. Conrin's "The EEG effects of tobacco smoking–a review" observed changes in the alpha wave readings of people who used nicotine. Changes to the evoked potential were also observed. Conrin [1980]. Studies have similarly uncovered and quantified differences in the EEG activity of light versus heavy smokers. In David G. Gilbert's "Effects of smoking and nicotine on EEG lateralization as a function of personality", a noticeable hemispheric asymmetry in alpha waves was observed in patients who were under the influence of nicotine. Theta wave disruptions were also observed. These abnormalities are believed to be related to the personality of the user. Gilbert [1987].

Walter S. Pritchard's "Electroencephalographic effects of cigarette smoking" observed an increase in beta2 waves in smokers who inhaled deeply, an effect associated with anxiety relief. In smokers who inhaled lightly, increases in delta, theta, and alpha waves were observed. Pritchard [1991]. Walter S. Pritchard's "Flexible effects of quantified cigarette-smoke delivery on EEG dimensional complexity" demonstrated a lowered EEG dimensional complexity in subjects who smoked often, and raised EEG dimensional complexity in subjects who smoked less frequently, after smoking from a quantified smoke delivery system. Pritchard et al. [1993].

2.2 Deep Learning and EEG Analysis

The preponderance of evidence of differences in EEG patterns between heavy and light smokers has also led to further study on the neurophysiology of smoking quitting and abstinence, including a study from Gilbert et. al, which indicated patients who quit using nicotine for 31 days experienced greater EEG deactivation when performing a stressful task. It was suggested their prior nicotine use affected the results of their EEG reading. Gilbert et al. [2004].

Robin Tibor Schirrmeister's "Deep learning with convolutional neural networks for EEG decoding and visualization" gave a model for using convoluted neural networks to analyze EEG results. This article also demonstrated comparable decoding results between deep convolutional neural networks and the filter bank common spatial patterns algorithm. Schirrmeister et al. [2017].

John Thomas' "EEG Classification via Convolutional Neural Network-Based Interictal Epileptiform Event Detection" achieved a mean 4-fold classification accuracy of 83.86% for determining if a given EEG displayed signs of epilepsy. Thomas et al. [2018].

In Baoguo Xu's "Wavelet Transform Time-Frequency Image and Convolutional Network-Based Motor Imagery EEG Classification", a two-layer convolutional neural network is used to extract data from wavelet transform-based EEG input. Accuracy was evaluated as high as 90%. Xu et al. [2019].

3 Materials and Methods

3.1 Data Set and Features

3.1.1 Data Pre-Processing

The data processing pipeline begins with collecting CNT files from neuroscans of patients and converting them to `.set` and `.fdt` files for processing. The head model is inserted into the data, with the VEOG and HEOG removed. A high pass filter at 2 Hz is applied to the data, alongside a low pass filter at 55 Hz. The file is then duplicated. The first duplicate has its first and last 12 seconds of data removed, while the second duplicate loses its first 14 seconds and its last 12 seconds. The files are merged and epoched at 4.096 milliseconds, which ensures that each epoch created will have 2048 points at a sampling rate of 500 Hz. Artifacts are then removed from the data using the Automatic Artifact Removal toolbox, version 1.3. This toolbox, created by Germán Gómez-Herrero, uses Blind Source Separation, also known as BSS, to find artifacts in the data, and reconstruct the data set without its artifacts. Gómez-Herrero [2007]. The reconstructed data set is then put into CORRMAP, a set of Python tools developed by Filipa Campos Viola, to remove blinks, lateral eye movement, and heartbeats. Viola [2011]. After this process is completed, the data is ready to be used.

3.1.2 Processing the Data For the Network

The data was provided to our team in `.set` and `.fdt` formats - a binary data format and file reading protocol plus metadata format, respectively. These files were generated by and meant to be interpreted in a `MATLAB` library called `EEGLab`. Delorme and Makeig [2004]. For the purposes of this pipeline, these files were uploaded to a Google Colab notebook and processed locally using the `MNE` Python library. Gramfort et al. [2014].

Once in MNE, the data is "clumped" into individual sets of 3 epochs, which are then averaged and transformed using a Morlet wavelet transformation into a 4-dimensional `numpy` array. Each entry on axis 0 of that array represents the mean power spectral density in decibels for 26 distinct frequency bands between 4Hz and 30Hz for a period of 12 seconds, measured at each of 29 total electrodes. In the process of data transformation, labels for each experimental condition are produced as a second `numpy` array from file labels provided by the lab.

Once the data has been imported and transformed, the arithmetic mean of the time dimension is calculated for each spectral window to represent the average value of that frequency window at that electrode for the given sample. Then, the data is shuffled so that the model receives random samples of data for training.

3.1.3 Power Spectral Density Data

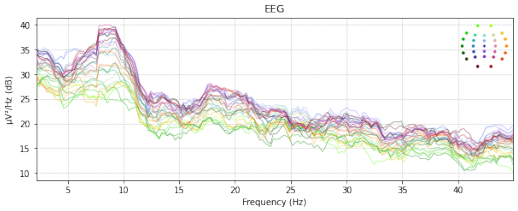

fig. 2a: Power Spectral Density plot with amplitude on the y axis and frequency on the x axis, from a single epoch sampled from a participant who had smoked a few minutes prior to their EEG session.

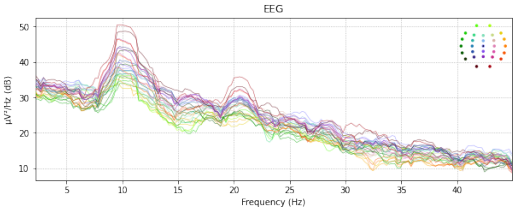

fig. 2b: Power Spectral Density plot with amplitude on the y axis and frequency on the x axis, from a single epoch sampled from a participant who had abstained from smoking 24h prior to their EEG session.

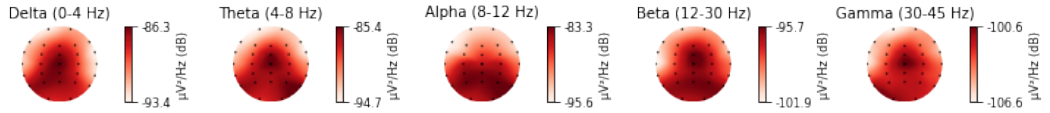

fig 3a: *Single epoch sample from a participant in the smoker group plotted to each sensor on a flattened topographic representation of the human scalp. The top represents the front of the head. Each model represents a different frequency traditionally considered of interest to neuroscience.*

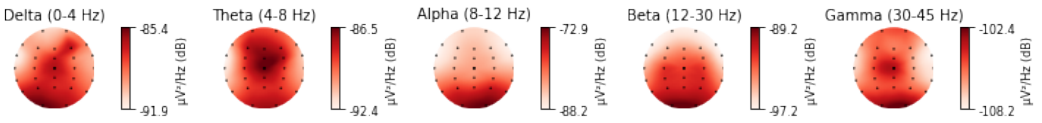

fig 3b: *Single epoch sample from a participant in the abstainer group plotted to each sensor on a flattened topographic representation of the human scalp. The top represents the front of the head. Each model represents a different frequency traditionally considered of interest to neuroscience.*

3.1.4 Test and Train Sets

After processing and shuffling, the data and labels are divided into a train and test set, with the training set comprising 80% of the data in the data set and the testing set making up the other 20%. The train set is then further divided again before fitting, into a validation set comprising 16% of the total data and the final training set then comprising the remaining 64%.

In addition to these data sets, we also reserve 25% of available data as a "second training set" composed specifically of epochs processed identically to the data listed above, except from participants not used to train the model. This is used in addition to the test set to determine both the model's overall generalizability and the ratio between its ability to classify data from previously-encountered participants and those to whom the model has not been exposed.

3.2 Methods

3.2.1 Decoding the Input

This section more formally defines how brain-signal decoding can be viewed as a supervised classification problem and includes the notation used to describe the methods.

It is assumed that we are given one EEG data set per subject i. Each data set is separated into labeled trials, which are segments of the original recording that each belong to one of several classes. Concretely, we are given data sets $Di = (X_1, y_1), \ldots, (X_{Ni}, y_{Ni})$, where N_i denotes the total number of recorded trials for subject i. The input matrix $X_j \in \mathbb{R}^{E \times F}$ of trial $j, 1 \leq j \leq N_i$ contains the pre-processed signals of E-recorded electrodes and F-discretized and averaged frequency windows discerned from the Morlet transformation process.

The corresponding class label of trial j is denoted by $y^{\,j}$. Values from a set of K class labels L that, in our case, correspond to the experimental condition, for our binary classification example:

$$\forall y_j : y_j \in L = \{l_0 = Smoking, l_1 = Abstaining\}$$

The decoder f is trained on these existing trials to ensure that it is able to assign the correct label to any unprocessed data sets. Concretely, we aim to train the decoder to assign the label y_j to trial X_j using the output of a parametric classifier $f(X_j; \theta) : \mathbb{R}^{E \times F} \longrightarrow L$ with parameters θ.

For the rest of this article, we assume that the classifier $f(X_j; \theta)$ is represented by a standard machine learning pipeline converted into two parts: a first part that extracts a feature representation $\varphi(X_j; \theta\varphi)$ with parameters $\theta\varphi$s learned from the data, and a second part consisting of a classifier g with parameters θg that is trained using these features, that is, $f(X_j; \theta) = g(\varphi(X_j; \theta\varphi), \theta g)$.

3.2.2 Convolution Layers

Two convolution layers are used to thoroughly obscure the data. To describe with precision: Where E is the number of electrodes, F is the number of frequency windows, and φ is the number of features produced by the convolution layer, for the i^{th} item in the data set:

$$X_i : \mathbb{R}^{E_{prev} \times F_{prev}} \longrightarrow X_i : \mathbb{R}^{(E_{prev}-1) \times (F_{prev}-1) \times \varphi}$$

Both layers are convoluted using a Rectified Linear Unit function to ensure linear transformation and that no negative data is used. In both layers, our model produced output features $\varphi = 26$, and the data was convoluted using a 3×3 grid.

3.2.3 Pooling

The pooling layer reduces the dimensionality of the data in a predictable way. Our two dimensional pooling filter contains dimensions α and β. The result of this function is as follows:

$$X_i : \mathbb{R}^{E_{prev} \times F_{prev} \times \varphi} \longrightarrow X_i : \mathbb{R}^{\frac{E_{prev}}{\alpha} \times \frac{F_{prev}}{\beta} \times \varphi}$$

3.2.4 Dropout

Dropout is a method of checks and balances for neural networks. By dropping random nodes of data, we help ensure that we are not overfitting with regards to our model. In our network, the dropout simply removes 20% of our data at random, utilizing TensorFlow's Dropout layer template. This layer, then, also outputs an item of shape

$$X_i \in \mathbb{R}^{E_{prev} \times F_{prev} \times \varphi}$$

relative to the previous layer.

3.2.5 Dense Layer

The dense layer is used to connect nodes together. Groups of n_{dense} size are connected on a dense layer using a Rectified Linear Unit function. The dimensionality of this layer is as follows:

$$X_i : \mathbb{R}^{E \times F \times \varphi} \longrightarrow X_i : \mathbb{R}^{E_{prev} \times F_{prev} \times n_{dense}}$$

In our model, while other values of n were attempted, the model seemed to best optimize with a value of $n = 3000$ neurons in this layer. After these weights are applied to each data item X_i, that item is compressed using a Flatten layer, which adjusted the dimensionality from three dimensions into a single dimension of the same size listed above.

3.2.6 Sigmoidal Classifier

The sigmoidal classifier works similarly to a logistic regression, inputting a set of real numbers with the above dimensionality and outputting a single item X_i with the following characteristics:

$$X_i \in \mathbb{R}[0, 1]$$

3.2.7 Training the Model

The model is trained one epoch at a time. A callback is used to save the weights of the model, and another callback is used to check for a premature stop. An Adam optimizer is used, with a learning rate of 0.0001, as it achieved the fastest optimization that did not result in overfitting.

4 Experimental Approach

4.1 Random Forest

An initial Random Forest Classifier was trained on cleaned, transformed samples of the data, both in its initial epoch-aggregate mean form and later in its individual-epoch form, in order to determine the feasibility of classifiers on the data. This random forest used bootstraps, gini estimators, and 1000 estimators. It was trained on a sample consisting of 1/10 of the total data, with a testing set consisting of 1/5 of the total data. This random forest was trained to 76% test set accuracy, a result which provided context for our later results on our CNN.

Our reasoning for a Random Forest comparison model is its use of multiple models trained only on specific subsets of the data. We believed that this trait would make it resistant to overfitting in the context of our dataset, and therefore a reliable comparison point to our convolutional net.

4.2 Baseline Model

A Baseline model was constructed based upon the same network and algorithms used in our final model. Recall that in the very first forward propagation during training, the model is initialized with weights comprising the dimensionality of the input data. These are all initialized with random weights. Specifically, if $P(\mathbf{x})$ is a random distribution of real numbers, and W represents our layer composed of weights W_i, then:

$$\forall \mathbf{W_i} \in \mathbf{W} : P(\mathbf{W_i})$$

4.3 Epoch-Aggregate Data

Our initial approach to training and evaluating the model consisted of aggregating the epoched data $X_i \in \mathbb{R}^{E \times F \times T}$ from each epoch i into an "aggregate epoch" consisting of the averaged data from all recorded epochs for that subject:

$$X_{avg} = \frac{1}{i} \sum_{n=1}^{i} X_i : X_i \in \mathbb{R}^{E \times F \times T}$$

However, this method was found to have less-than-ideal generalizability and tended to fit itself on individual noise and artifact characteristics of participant sessions. Additionally, it resulted in a far reduced sample size and therefore training set, since the fifty to eighty epochs recorded for each subject were reduced to a single data point.

4.4 Single-Epoch Data

As such, and in an effort to improve the ability of the model to work with artifact-laden data, we opted to approach the data set on an epoch-by-epoch basis, treating each four-second epoch as an individual data point. In this instance, the data prior to power-spectral-density transformation was defined simply by:

$$X_i \in \mathbb{R}^{E \times F \times T_{epoch}}$$

However, while this model showed promise in improved accuracy on our test and validation sets, it also did not appear to generalize. This was evidenced by the model's inability to accurately classify data from subjects to which it had not been exposed (our "second test set"), performing worse on this group than the baseline model. We made the determination that a possible cause of this issue was a relative lack of abundance in the spectral features which were necessary to form a model predicting unseen data. This led us to develop the current method of merging and averaging multiple EEG sections into a single representative Power Spectral Density data point:

$$X_i \in \mathbb{R}^{E \times F}$$

4.5 Evaluation Criteria

A successful model should be able to classify data from sources based on whether or not they have recently consumed nicotine. Our baseline model is able to sort these groups with an accuracy of 41%. Our model predicts this with an accuracy of 94%, which is an improvement of 53%. The secondary point of evaluation we are using is how well it sorts epochs of patients it has seen before, versus patients it hasn't seen before. Upon testing, patients it had seen before were sorted with an accuracy of 93%, while subjects it hadn't seen before were sorted with an accuracy of 91%, which is a decrease of 2%. The final means by which we are evaluating our model pertains to the quality and usability of the features it extracts. Determining why our model is accurate can shed light on the relationship between smoking and concurrent or subsequent brain activity, which can shed light on, among many other things, the relationship between addiction and cognition.

5 Next Steps

There is still plenty of work to be done to this network. Our classification results are very promising, and serve as a reminder that this approach can work very well, but there is room to improve the

performance. As a team, we are always looking into new methods, bug fixes, and test results to ensure that our data is being processed properly. We are currently not testing our full data set, as most of it is required to train the network. One of our next goals is to process more patients at a time, which requires more processing power. We are also looking to further optimize our input data and pipeline to allow more patients to be compared at a time.

References

J. Conrin. The EEG effects of tobacco smoking–a review. *Clinical EEG (electroencephalography)*, 11(4):180–187, October 1980. ISSN 0009-9155. doi: 10.1177/155005948001100407.

Arnaud Delorme and Scott Makeig. Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1): 9–21, Mar 2004. ISSN 01650270. doi: 10.1016/j.jneumeth.2003.10.009.

David Gilbert, Francis McClernon, Norka Rabinovich, Chihiro Sugai, Louisette Plath, Gregory Asgaard, Yantao Zuo, Jodi Huggenvik, and Nazeih Botros. Effects of quitting smoking on eeg activation and attention last for more than 31 days and are more severe with stress, dependence, drd2 a1 allele, and depressive traits. *Nicotine & tobacco research : official journal of the Society for Research on Nicotine and Tobacco*, 6:249–67, 05 2004. doi: 10.1080/14622200410001676305.

David G. Gilbert. Effects of smoking and nicotine on eeg lateralization as a function of personality. *Personality and Individual Differences*, 8(6):933–941, 1987. ISSN 0191-8869. doi: https://doi.org/10.1016/0191-8869(87)90144-9. URL `https://www.sciencedirect.com/science/article/pii/0191886987901449`.

Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S. Hämäläinen. Mne software for processing meg and eeg data. *NeuroImage*, 86:446–460, Feb 2014. ISSN 10538119. doi: 10.1016/j.neuroimage.2013.10.027.

Germán Gómez-Herrero. Automatic Artifact Removal (AAR) toolbox v1.3 (Release 09.12.2007) for MATLAB, 2007. URL `https://germangh.github.io/aar/aardoc/aar.html`.

Walter S. Pritchard. Electroencephalographic effects of cigarette smoking. *Psychopharmacology*, 104(4):485–490, August 1991. ISSN 1432-2072. doi: 10.1007/BF02245654. URL `https://doi.org/10.1007/BF02245654`.

Walter S. Pritchard, David G. Gilbert, and Dennis W. Duke. Flexible effects of quantified cigarette-smoke delivery on EEG dimensional complexity. *Psychopharmacology*, 113(1):95–102, November 1993. ISSN 1432-2072. doi: 10.1007/BF02244340. URL `https://doi.org/10.1007/BF02244340`.

Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420, August 2017. ISSN 1065-9471. doi: 10.1002/hbm.23730. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5655781/`.

John Thomas, Luca Comoretto, Jing Jin, Justin Dauwels, Sydney S. Cash, and M. Brandon Westover. EEG Classification via Convolutional Neural Network-Based Interictal Epileptiform Event Detection. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, 2018:3148, July 2018. doi: 10.1109/EMBC.2018.8512930. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6775768/`.

Filipa Campos Viola. CORRMAP plug-in for EEGLAB, 2011. URL `http://www.debener.de/corrmap/corrmapplugin1.html`.

B. Xu, L. Zhang, A. Song, C. Wu, W. Li, D. Zhang, G. Xu, H. Li, and H. Zeng. Wavelet transform time-frequency image and convolutional network-based motor imagery eeg classification. *IEEE Access*, 7:6084–6093, 2019. doi: 10.1109/ACCESS.2018.2889093.