Using Deep and Convolutional Neural Networks for Accurate Emotion Classification on DEAP Dataset

Samarth Tripathi  
Columbia University, New York  
samarthtripathi@gmail.com

Shrinivas Acharya  
Amazon, Hyderabad  
acharys@amazon.com

Ranti Dev Sharma  
University of California, San Diego  
ranti.iitg@gmail.com

Sudhanshi Mittal  
Oracle, Hyderabad  
sudhanshumittal1992@gmail.com

Samit Bhattacharya  
Indian Institute of Technology, Guwahati  
samit@iitg.ernet.in

Abstract

Emotion recognition is an important field of research in Brain Computer Interactions. As technology and the understanding of emotions are advancing, there are growing opportunities for automatic emotion recognition systems. Neural networks are a family of statistical learning models inspired by biological neural networks and are used to estimate functions that can depend on a large number of inputs that are generally unknown. In this paper we seek to use this effectiveness of Neural Networks to classify user emotions using EEG signals from the DEAP (Koelstra et al (2012)) dataset which represents the benchmark for Emotion classification research. We explore 2 different Neural Models, a simple Deep Neural Network and a Convolutional Neural Network for classification. Our model provides the state-of-the-art classification accuracy, obtaining 4.51 and 4.96 percentage point improvements over (Rozgic et al (2013)) classification of Valence and Arousal into 2 classes (High and Low) and 13.39 and 6.58 percentage point improvements over (Chung and Yoon(2012)) classification of Valence and Arousal into 3 classes (High, Normal and Low). Moreover our research is a testament that Neural Networks could be robust classifiers for brain signals, even outperforming traditional learning techniques.

Introduction

Emotions are very important in human decision handling, interaction and cognitive process (Sreeshakthy et al (2016)). As technology and the understanding of emotions are advancing, there are growing opportunities for automatic emotion recognition systems. There have been successful research breakthroughs on emotion recognition using text, speech, facial expressions or gestures as stimuli. However one of the new and exciting directions this research is heading is EEG-based technologies for automatic emotion recognition, as it becomes less intrusive and more affordable, leading to pervasive adoption in healthcare applications. In this paper we focus on classifying user emotions from Electroencephalogram (EEG) signals, using various neural network models and advanced techniques. For our research we particularly explore Deep Neural Networks and Convolutional Neural Networks, using advanced machine learning techniques like Dropout, for emotion classification. Neural network is a machine that is designed to model the way our brain performs a particular task, where the key concepts of brain as a complex, non-linear and parallel computer are imitated (Haykin (2004)), and possess the ability to model and estimate complex functions depending on multitude of factors. Moreover recent developments in machine learning have shown neural networks to provide prime accuracy in various varied tasks such as Text and Sentiment Analysis (Kim (2014)), Image recognition (Krizhevsky et al (2012)), and Speech analysis.

Recently, the affective EEG benchmark database DEAP (Koelstra et al (2012)) was published, which presents multimodal data set for the analysis of human affective states. The electroencephalogram (EEG) and peripheral physiological signals of 32 participants were recorded as each watched 40 one-minute long excerpts of music videos. Participants rated each video in terms of the levels of arousal, valence, like/dislike, dominance, and familiarity. A 32 EEG channels Biosemi ActiveTwo device was used to record the EEG signals when the subjects were exposed to the videos. Other than the EEG recordings, channels also recorded some physiological signals like temperatures and respiration etc. Methods and results were presented for single-trial classification of arousal, valence, and like/dislike ratings using the modalities of EEG, peripheral physiological signals, and multimedia content analysis. Automatic classification of human emotion using EEG signals has been researched upon in detail by various scholars. However in the release of DEAP data, research academia finds a standardized dataset to effectively measure and compare accuracies for various classification algorithms.

We use two different Neural Models for classification, the first being a Deep Neural Network comprising of 4 Neural layers. The model contains an initial neural layer of 5000 nodes, followed by layers of 500 and 1000 neurons respectively, before the output neural layer of 2 or 3 nodes depending on the classification classes. All the layers are fully connected with Softmax (Dunne and Campbell (1997)) acting as the Activator, and use Dropout (Srivastava et al (2014))
Emotion is a psycho-physiological process triggered by conscious and/or unconscious perception of an object or situation and is often associated with mood, temperament, personality and disposition, and motivation. Emotions play an important role in human communication and can be expressed either verbally through emotional vocabulary or by expressing nonverbal cues such as intonation of voice, facial expressions, and gestures (Liu and Sourina (2014)). The recent public release of DEAP dataset, provides a much needed impetus to the growing community of HCI researchers in emotion recognition. Before DEAP, most of the studies on emotion assessment had focused on the analysis of facial expressions and speech to determine a person's emotional state. However, physiological signals are also known to include emotional information that can be used for emotion assessment, but they have received less attention. The database explores the possibility of classifying emotion dimensions induced by showing music videos to different users, using the signals originating from the central nervous system (CNS) and the peripheral nervous system (PNS). DEAP uses Russell's valence-arousal scale, widely used in research on affect, to quantitatively describe emotions. In this scale, each emotional state can be placed on a 2D plane with arousal and valence as the horizontal and vertical axes. Arousal can range from inactive (e.g., uninterested, bored) to active (e.g., alert, excited), whereas valence ranges from unpleasant (e.g., sad, stressed) to pleasant (e.g., happy, elated). It contains EEG and peripheral physiological signals recorded using a Biosemi ActiveTwo system at a sampling rate of 512 Hz using 32 active AgCl electrodes (placed according to the international 10-20 system). DEAP has the highest number of participants in publicly available databases for analysis of spontaneous emotions from physiological signals. In addition, it is the only database that uses music videos as emotional stimuli (Liu and Sourina (2014)). Since the release of DEAP dataset, multiple researchers have been using it for emotion recognition. (Liu and Sourina (2014)) research, explores real-time Electroencephalogram (EEG)-based emotion recognition algorithm using Higuchi Fractal Dimension (FD) Spectrum. They recognize EEG as a nonlinear and multi-fractal signal, hence its FD spectrum can give a better understanding of the nonlinear property of EEG using Support Vector Machines as a classifier. They test their approach on both DEAP and their own dataset. On DEAP database, they report a classification accuracy of 8 emotions 53.7% in subject-dependent classification. (Srivastava et al (2014)) research, depicts models for Classification of DEAP's EEG data to different energy bands using wavelet transform and neural networks. They divide EEG signal into different bands using discrete wavelet transformation with db8 wavelet function for processing. Statistical and energy based features are extracted from the bands, based on the features emotions are classified with feed forward neural network with weight optimized algorithm like PSO.

Our research builds upon the works of (Chung and Yoon(2012)) which focuses mainly on classification of DEAP data into classes of Valence and Arousal using statistical and shallow learning methods like Bayesian Classification. Their simple classification methods provide a starting baseline for our study to compare results. They classify the user data into 2(high and low) and 3(high, normal, low) classes for both Valence and Arousal. They achieve 66.6% and 66.4% accuracy for 2 classes, and 53.4% and 51.0% for three classes, on Valence and Arousal. Similarly the work of (Candra et al (2015)) investigates the how the window size effects the classification of DEAP's EEG data using wavelet entropy and SVMs. They conclude that an overly wide window can lead to information overload which causes the feature to be mixed up with other information. Similarly the information about emotion might not be adequately extracted if the time window is too short. They then use the popular discrete wavelet transform (DWT) coefficient for extracting time-frequency domain features in EEG signals. Their investigation revealed that arousal can be classified up to 65.33% accuracy using the window length of 310 seconds; while valence can be classified up to 65.13% accuracy using the window length of 312 seconds. (Sohaib et al (2013)) provide a concise evaluation for various classifiers for Emotion Recognition. However instead of the DEAP data, they test using their own EEG dataset of 20 subjects when subjected to images from International Affective Picture System (IAPS). They evaluate the classification for K-Nearest Neighbor (KNN), BayesianNetwork (BN), Artificial Neural Network (ANN) and Support Vector Machine (SVM). Their results showed that it is difficult to train...
a classifier to be accurate over large datasets (15 subjects) but KNN and SVM with the proposed features were reasonably accurate over smaller datasets (5 subjects) identifying emotional states with an accuracy up to 77.78%. Our research is also immensely inspired by the works of (Rozgic et al (2013)). Their research provides the previous best classification accuracy on the DEAP data for both Valence and Arousal, which we successfully improve upon. Their innovative technique is based on three steps: Firstly, in contrast to the typical feature extraction on the response-level, they represent the EEG signal as a sequence of overlapping segments and extract feature vectors on the segment level; Secondly they transform segment level features to the response level features using projections based on a novel non-parametric nearest neighbour model; and Thirdly they perform classification on the obtained response-level features. They used KPCA dimensionality reduction as a preprocessing step for each classifier, and deployed their data on classification algorithms such as Naive Bayes Nearest Neighbour, Nearest Neighbour Voting and RBF SVMs. They used leave-one-response-out cross validation scheme to obtain single subject accuracy and report accuracies averaged over all subjects.

**Dataset**

The DEAP dataset consists of two parts, firstly the ratings from an online self-assessment where 120 one-minute extracts of music videos were each rated by 14-16 volunteers based on arousal, valence and dominance. Secondly, the participant ratings, physiological recordings and face video of an experiment where 32 volunteers watched a subset of 40 of the above music videos. EEG and physiological signals were recorded and each participant also rated the videos as above. For 22 participants frontal face video was also recorded. The official dataset contains all individual ratings from the online self-assessment, YouTube links of the music videos used, all ratings participants gave to the videos, the answers participants gave to the questionnaire before the experiment, the frontal face video recordings from the experiment for participants 1-22 and the original unprocessed physiological data recordings from the experiment in BioSemi bdf format. However for our experiment we use the preprocessed (data downsampled to 128Hz, EOG removal, filtering, segmenting etc.) physiological data recordings from the DEAP experiment in Matlab and Python (numpy) format. This format is especially useful for testing classification or regression techniques without hassle of explicitly processing all the data first. For each of the 32 participants we have 2 arrays illustrated in Table 1.

<table>
<thead>
<tr>
<th>Table 1: DEAP dataset representation for each subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array Name</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>data</td>
</tr>
<tr>
<td>labels</td>
</tr>
</tbody>
</table>

Our data. We divide the 8064 readings per channel, into 10 batches of approximately 807 readings each. For each batch we extract the mean, median, maximum, minimum, standard deviation, variance, range, skewness and kurtosis values for the 807 readings. Hence for each of the 10 batches of a single channel we extract 9 values mentioned above, we get 90 values as our processed dataset. We further add the net mean, median, maximum, minimum, standard deviation, variance, range, skewness and kurtosis values for the entire 8064 readings along with the experiment and participant number to our dataset, bringing it up to 101 values per channel.

As mentioned in the work of (Candra et al (2015)), the optimal sliding window size of 310 and 3-12 seconds was ideal for classification of Valence and Arousal respectively. The 8064 readings represent the EEG values recorded over the duration of 1 minute of the participant viewing the video; classifying them into 10 batches gives us a comprehensive outlook of their emotion for a 6 second range. Moreover we chose statistical methods to reduce the dimensionality of our EEG dataset mapping its feature probability density function to a Gaussian distribution and then effectively catching it using statistical features like mean, variance, range etc (Gupta and Gupta (2009)). (Jahankhani et al (2007)) effectively demonstrate this method using maximum, minimum, mean and standard deviation of wavelet coefficients for signal classification. To reduce the volume of their EEG data, they partitioned their samples into 16 windows of 256 time points each. We effectively follow a similar statistical dimensionality reduction for our dataset.

From 322560 readings per experiment, we finally arrive at 4040 values (101 reduced readings * 40 channels). These 4040 values per experiment form our initial processed dataset, which we use to train our neural model. As mentioned before we use the leave-one-response-out cross validation. This implies that for participant 1, we train our model using readings for participant 2 to 32 and record our classification accuracy for participant 1. For participant 2, we train a new model with the same architecture but this time we train using readings for participant 1 and 3 to 32 and record our classification accuracy for participant 2 and so on. This allows us to train our model for 1240 experiments (31 participants * 40 experiments) and predict for the 40 experiments for each of the subjects one after another. The labels data are iterated and for each of Valence and Arousal we extract one hot encoding based outputs for classification in both 2 classes (ratings divided as more than 5 and less than 5), and 3 classes (ratings divided as more than 6, between 4 and 6, less than 4).
huge datasets, in the order of millions. However, as dataset size grows, the computational complexity of classical machine learning techniques increases significantly. Deep learning usually involves neural layers learning on high-dimensional data, allowing for the modeling of complex, non-linear relationships. For our problem, we deploy Rectified Linear Units (ReLU) as our non-linear activation function. As mentioned before, we use ReLU as the activation function, as it has been argued to be more biologically plausible and practical (Glorot et al. (2011) (LeCun et al. (1998)). The values for the learning rate and gradient descent for RMSProp are standard over academia implementation (Dauphin et al. (2015)). The number of nodes in various Neural Layers is crucial to the performance of such a network. We test our valence model for multiple such permutations to attain the aforementioned values and detail them in the results section.

**Convolutional Neural Network (CNN)**

CNNs are very effective models for Image classification tasks. For our model, we try and convert our DEAP data into 2D image format so our CNN model can learn to classify them effectively. So for each experiment we have 40 channels with processed 101 readings each. So we represent the data for each experiment as a 2D array image of 40 * 101 size. Our first Convolution layer takes this 2D array as input and the Convolution operation uses 100 initial convolution filters and a convolutional kernel of 3 rows and 3 columns. The first Convolution layer uses ‘TanHyperbolic’ as the Activation function for Valence Classification model, and ‘Relu’ or Rectilinear units as Activation for Arousal model. In our experiment we realized the choice of Activation functions for this first layer are of cardinal importance, as some functions (like sigmoid, softmax) might not be able to activate neurons of later layers consistently, making the model defective. The next layer is another Convolutional Neural Layer which again with 100 filters and 3*3 size kernel. This layer uses ‘TanHyperbolic’ as the Activation function for both Valence and Arousal classification.

The next layer is a MaxPooling layer, and our pooling is traditional 2 dimensional max pooling over 2x2 blocks. We use Dropout on the outputs of MaxPooling layer, with a Dropout probability of 0.25, to form a Flat 1 dimensional layer. The layer feeds to a Fully connected Dense neural layer with an output dimensionality of 128. We use ‘TanHyperbolic’ as our activation function again use Dropout with 0.5 probability on the Outputs of Dense layer. Finally our output layer with an output dimensionality of 128, with a gradient direction of 0.9 for our RMSProp learning. Neural networks generally use Activation functions which are used to transform the activation level of a unit (neuron) into an output signal. A non-linear activation function allows Neural networks to compute nontrivial problems using only a small number of nodes. As mentioned before we deploy Rectified Linear Units (ReLU) and Softmax as our non-linear activation function.

The neural networks works by minimizing the cross entropy or noise between the actual and predicted outcomes. Deep learning usually involves neural layers learning on huge datasets, in the order of millions. However as dataset size is constrained, we use Dropout technique to obtain good results. While using the Dropout technique it is important to use a high epoch. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. For batch training all of the training samples pass through the learning algorithm simultaneously in one epoch before weights are updated. We train our model in batches of 310, with an epoch of 250 to provide for good detailed learning process. For our input layer we use a Dropout probability of 0.25, while for the successive hidden layers we use a Dropout factor of 0.5. These probabilities have been empirically shown to yield best results which we also confirm in our own tests and our standard in research academia. Our decision to use ReLu as the activation function is justified because it has been argued to be more biologically plausible and practical (Glorot et al. (2011) (LeCun et al. (1998)).

**Model**

We use two different neural models for our research, a simple Deep Neural Network and an advanced Convolutional Neural Network. Both models are complemented using contemporary machine learning techniques like Dropout technique and Rectilinear Units to introduce non-linearity in our model. The architectures of our models are elaborated. Both models are implemented using Theano (Bastien et al (2012)) and Keras (Chollet (2015)) libraries in Python.

**Deep Neural Network (DNN)**

Our DNN model uses 4 fully connected, dense neural layers where the output of one layer serves as the input for the next layer. The first input layer takes 4040 readings for the experiment and the neural layers output dimensions are 5000 with a feature count of 400. The layer uses ReLu as an activation function and has a dropout probability of 0.25. The next layer takes 5000 values as input and has an output dimension of 500. The next consecutive layer increases the dimension to 1000. Both these layers use ReLu as the Activation function and a Dropout probability of 0.5. The final neural layer takes these 1000 inputs and reduces them to the final output of 2 or 3 classes as one hot encodings. The final layer uses Softmax (Dunne and Campbell (1997)) as the Activation function with a Dropout input probability of 0.5. Figure 1 represents the basic architecture of our DNN.

To speed up the learning rates we use RMSProp (Tieleman and Hinton (2012)) for gradient descent, which divides the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight. We have used a learning rate of 0.00001 and a gradient direction of 0.9 for our RMSProp learning. Neural networks generally use Activation functions which are used to transform the activation level of a unit (neuron) into an output signal. A non-linear activation function allows Neural networks to compute nontrivial problems using only a small number of nodes. As mentioned before we deploy Rectified Linear Units (ReLU) and Softmax as our non-linear activation function.

The neural networks works by minimizing the cross entropy or noise between the actual and predicted outcomes. Deep learning usually involves neural layers learning on huge datasets, in the order of millions. However as dataset size is constrained, we use Dropout technique to obtain good results. While using the Dropout technique it is important to use a high epoch. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. For batch training all of the training samples pass through the learning algorithm simultaneously in one epoch before weights are updated. We train our model in batches of 310, with an epoch of 250 to provide for good detailed learning process. For our input layer we use a Dropout probability of 0.25, while for the successive hidden layers we use a Dropout factor of 0.5. These probabilities have been empirically shown to yield best results which we also confirm in our own tests and our standard in research academia. Our decision to use ReLu as the activation function is justified because it has been argued to be more biologically plausible and practical (Glorot et al. (2011) (LeCun et al. (1998)). The values for the learning rate and gradient descent for RMSProp are standard over academia implementation (Dauphin et al. (2015)). The number of nodes in various Neural Layers is crucial to the performance of such a network. We test our valence model for multiple such permutations to attain the aforementioned values and detail them in the results section.

**Convolutional Neural Network (CNN)**

CNNs are very effective models for Image classification tasks. For our model, we try and convert our DEAP data into 2D image format so our CNN model can learn to classify them effectively. So for each experiment we have 40 channels with processed 101 readings each. So we represent the data for each experiment as a 2D array image of 40 * 101 size. Our first Convolution layer takes this 2D array as input and the Convolution operation uses 100 initial convolution filters and a convolutional kernel of 3 rows and 3 columns. The first Convolution layer uses ‘TanHyperbolic’ as the Activation function for Valence Classification model, and ‘Relu’ or Rectilinear units as Activation for Arousal model. In our experiment we realized the choice of Activation functions for this first layer are of cardinal importance, as some functions (like sigmoid, softmax) might not be able to activate neurons of later layers consistently, making the model defective. The next layer is another Convolutional Neural Layer which again with 100 filters and 3*3 size kernel. This layer uses ‘TanHyperbolic’ as the Activation function for both Valence and Arousal classification.

The next layer is a MaxPooling layer, and our pooling is traditional 2 dimensional max pooling over 2x2 blocks. We use Dropout on the outputs of MaxPooling layer, with a Dropout probability of 0.25, to form a Flat 1 dimensional layer. The layer feeds to a Fully connected Dense neural layer with an output dimensionality of 128. We use ‘TanHyperbolic’ as our activation function again use Dropout with 0.5 probability on the Outputs of Dense layer. Finally our output layer has an output dimensionality of 128, with a gradient direction of 0.9 for our RMSProp learning. Neural networks generally use Activation functions which are used to transform the activation level of a unit (neuron) into an output signal. A non-linear activation function allows Neural networks to compute nontrivial problems using only a small number of nodes. As mentioned before we deploy Rectified Linear Units (ReLU) and Softmax as our non-linear activation function.

The neural networks works by minimizing the cross entropy or noise between the actual and predicted outcomes. Deep learning usually involves neural layers learning on huge datasets, in the order of millions. However as dataset size is constrained, we use Dropout technique to obtain good results. While using the Dropout technique it is important to use a high epoch. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. For batch training all of the training samples pass through the learning algorithm simultaneously in one epoch before weights are updated. We train our model in batches of 310, with an epoch of 250 to provide for good detailed learning process. For our input layer we use a Dropout probability of 0.25, while for the successive hidden layers we use a Dropout factor of 0.5. These probabilities have been empirically shown to yield best results which we also confirm in our own tests and our standard in research academia. Our decision to use ReLu as the activation function is justified because it has been argued to be more biologically plausible and practical (Glorot et al. (2011) (LeCun et al. (1998)). The values for the learning rate and gradient descent for RMSProp are standard over academia implementation (Dauphin et al. (2015)). The number of nodes in various Neural Layers is crucial to the performance of such a network. We test our valence model for multiple such permutations to attain the aforementioned values and detail them in the results section.
our experiments we use 250 epochs and train our model using batches of 50 experiments each. The model is detailed in Algorithm 1 and Figure 2.

Algorithm 1 Pseudocode for Convolutional Neural Model:

Require: Training EEG Dataset \( nntrX \), Training Valence/Arousal Values \( nntrY \), Testing subject’s EEG Dataset \( nnteX \), Testing Valence/Arousal Values \( nnteY \)

Ensure: Accuracy Array for Subjects \( \text{accAll} \)

1: procedure NEURALMODEL ( \( nntrX, nnteY \) )
2: \( \text{batchSize} = 50; \text{nbClasses} = 2; \text{nbEpoch} = 5; \text{imgRows, imgCols} = 40, 101 \)
3: \( \text{nbFilters} = 100; \text{nbPool} = 2; \text{nbConv} = 3; \text{NEpoch} = 50 \)
4: \( \text{model} = \text{Sequential()} \)
5: \( \text{model.add(Convolution2D}(\text{nbFilters, nbConv, nbConv, border_mode = 'valid', input_shape = (1, imgRows, imgCols)))} \)
6: \( \text{model.add(Activation('tanh'))} \)
7: \( \text{model.add(Convolution2D}(\text{nbFilters, nbConv, nbConv})) \)
8: \( \text{model.add(Activation('tanh'))} \)
9: \( \text{model.add(MaxPooling2D(pool_size = (nbPool, nbPool))} \)
10: \( \text{model.add(Dropout}(0.50)) \)
11: \( \text{model.add(Flatten)} \)
12: \( \text{model.add(Dense}(128)) \)
13: \( \text{model.add(Activation('tanh'))} \)
14: \( \text{model.add(Dropout}(0.25)) \)
15: \( \text{model.add(Dense}(\text{nbClasses})) \)
16: \( \text{model.add(Activation('softmax'))} \)
17: \( \text{sgd} = \text{SGD}(lr = 0.00001, \text{decay} = 1e-6, \text{momentum} = 0.9, \text{nesterov} = \text{True}) \)
18: \( \text{model.compile(loss = 'categorical_crossentropy', optimizer = sgd)} \)
19: \( \text{accAll} = \emptyset \)
20: for all epoch in (1 : NEpoch) do
21: \( \text{batchSize, nbEpoch, validation_data = (nnteX, nnteY)} \)
22: \( \text{valLoss, valAccuracy} = \text{model.evaluate}(nnteX, nnteY, batch_size = 1) \)
23: \( \text{accAll.append(valAccuracy)} \)
24: end for
25: return accAll
26: end procedure

The Convolution layer is the core building block of a Convolutional Network. It computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and the region they are connected to the input volume. Our CNN works as follows, during the forward pass, we slide (more precisely, convolve) each filter across the width and height of the input volume, producing a 2-dimensional activation map of that filter. As we slide the filter, across the input, we are computing the dot product between the entries of the filter and the input. Once a filter has been glazed over the complete input, we find the single most important feature using max-over-time pooling operation (POOL Layer). This allows us to correctly identify one feature for each filter. The model repeats this for each filter in the image, to obtain best features for an experiment, in each convolution. Stacking these features for all filters along the depth dimension forms the full output volume. Thus, every entry in the output volume can also be interpreted as an output of a neuron that looks at only a small region in the input and shares parameters with neurons in the same activation map.

The Max Pooling layer’s function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. Neurons in a Fully Connected layer have full connections to all activations in the previous layer, as seen in regular neural network models. Their activations can hence be computed with a matrix multiplication followed by a bias offset. The model uses the SGD method which first divides the dataset into small batches of examples, compute the gradient using a single batch and make an update, then move to other batches of examples.

Results

In this final section we compare our classification results for both DNN and CNN. We also compare our results with that of (Chung and Yoon(2012)) for classification in 3 classes and (Rozgic et al (2013)) for classification in 2 classes, which represent the previous state of the art classification accuracy for both Valence and Arousal. We first evaluate our simple DNN model and some of the various configurations of neural layer sizes and parameters we experimented with, as detailed in Table 2. The table illustrates various DNN models used for Valence classification using our initial processed dataset. As is evident from the Table the size of minibatches have minimal effect on the classification accuracy. The learning rate is more important to the classification accuracy but only marginally, while the dropout probability that provides the best classification uses 0.25 drop probability for input layer and 0.5 for subsequent layers. For classification in 2 classes, as High(more than 5) and Low(less than 5) for both Valence and Arousal, our DNN performs commendably with maximum classification accuracies of 75.78 and 73.281 respectively. Our best DNN model had layers of...
To reach this final configuration we did multiple experiments varying layer sizes and other parameters, and were constrained by hardware limitations to test larger Neural Models. Some implications from our DNN results include the size of initial hidden layer leads to a general increase in classification accuracy. This difference is even more pronounced for Arousal classification where both the DNN and CNN models provide comparable accuracies but the CNN model being more unpredictable ranging accuracies as low as 45% and as high as 92.5%. Compared to the DNN model which provides a consistent classification between 65%-80% per subject, the CNN models range does seem intriguing, though better.

For classification in 3 classes High (more than 6), Normal (between 4 and 6), and Low (below 4), both our models achieve State of the Art classification accuracies, convincingly outperforming the previous research. The CNN models again outperforms the DNN model, but the DNN models accuracies are more uniform for Arousal. Our DNN model provides 58.44% and 55.70% for Valence and Arousal respectively for 3 classes. The CNN model provides 66.79% and 57.58% for Valence and Arousal.

**Conclusion**

In this paper we build upon prior research in the field of Emotion recognition and explore new techniques of using the effectiveness of Neural Networks to classify user emotions using EEG signals from the DEAP (Koelstra et al. (2012)) dataset. Our study provides the state-of-the-art classification accuracy, obtaining substantial improvements over prior researches and more importantly prove that Neural Networks could be robust classifiers for brain signals, outperforming traditional learning techniques. Our prime classification model uses a 2 dimensional Convolutional Neural Networks to effectively classify preprocessed EEG data presented in the form of 2D array, while incorporating contemporary techniques like Dropout and Rectilinear Units.

**References**


Francois Chollet.: "Keras”. In: Github 'https://github.com/fchollet/keras' (2015)


