Neurofusion: Fusing MEG and EEG Data

Paul Warren
Stanford CS 231n, Winter 2015-2016
pwarren@stanford.edu

1. Abstract

Understanding the human brain requires advancing non-invasive human neuroimaging monitoring and analysis capabilities. Currently, two dominant non-invasive brain activity collection methods are electro- and magneto-encephalography (EEG and MEG). They focus, respectively, on recording the electrical and magnetic fields generated by the flow of electrical current through neurons and head tissue [1]. This project is part of a year-long neuroimaging side project that explores different ways of fusing MEG and EEG data together and examines whether that fusion increases our ability to predict a user's intention from raw MEG/EEG signals. The CS 231n portion of this project was spent attempting to develop a neural network architecture that could use EEG data alone (64 data streams x 198 time steps per stream) to classify the examples into three categories based on whether the subject was shown a familiar face, an unfamiliar face, or a control. We implement an SVM baseline and describe how we can use convolutional neural networks, recurrent neural networks, and optimally convolutional recurrent neural networks for the task. We experiment with those architectures and with transforming the EEG data into a spectrogram using a Fast Fourier Transform. Although we see disappointing accuracy, we believe this is the result of poor hyperparameter choice, and that future work will demonstrate that a recurrent convolutional neural network is the best way to perform classification over EEG data. Future work will then extend this into a multimodal model to examine whether an EEG+MEG fusion increases our ability to predict a user's intention compared to either modality alone.

2. Introduction

Understanding the human brain requires advancing non-invasive human neuroimaging monitoring and analysis capabilities. Currently, two dominant non-invasive brain activity collection methods are electro- and magneto-encephalography (EEG and MEG). They focus, respectively, on recording the electrical and magnetic fields generated by the flow of electrical current through neurons and head tissue [1]. Both EEG and MEG provide good temporal resolution (on the order of 1ms [2]) but poor spatial resolution (on the order of 1cm [3]). Together, they provide a more accurate understanding of brain activity than either modality does alone [4].

Current neuroscience efforts to fuse MEG and EEG data focus on improving the estimated location of the neural activity that generated the observed signals ("source localization"). Source localization using concurrently collected MEG and EEG signals is more accurate than source localization using MEG or EEG data alone [5]. Source localization is useful for furthering our understanding of how the tasks the subjects are performing relate to underlying neural mechanisms. We may not, however, always have prior knowledge about what task the subject is performing when we look at the MEG and EEG data, as we do during neuroimaging studies. In recent years, there has been an increasing interest in using neuroimaging signals to predict the task the subject is thinking about. These predictions can then be interpreted as commands and relayed to external devices like prosthetics, vehicles, user interfaces, and other control systems. The systems that acquire brain signals, decode them into intentions, and relay commands to external devices are called Brain-Computer Interfaces (BCIs) [6].

Currently, most BCIs focus on collecting and analyzing EEG signals. If EEG signals perfectly captured the electric field generated by neuronal activity, or if MEG signals perfectly captured the magnetic field generated by neuronal activity, using a single sensor type would be enough to decode the signals. However, both EEG and MEG data have relatively high signal-to-noise ratios, and combining them has been shown to increase accuracy during source localization. This suggests that combining concurrently collected MEG and EEG data decreases the overall signal-to-noise ratio. Neurofusion is a year-long project that explores different ways of fusing MEG and EEG data together and examines whether that fusion increases our ability to predict a user's intention from raw MEG/EEG signals. If it does, we may be able to develop more accurate BCIs by moving from collecting MEG or EEG signals alone to collecting both concurrently.

This project was started as a final project in CS 221 in the Fall, continued as a CS 231N final project and CS 199 independent study this Winter, and will continue this Spring. CS 221 and CS 199 were spent defining the problem, exploring relevant literature, and choosing a dataset. CS 231n was spent properly preprocessing the dataset and implementing some unimodal (EEG only) models.

Our specific task for cs231n is (after preprocessing) to classify each example (64 channels x 198 time steps per channel) into three categories based on which of the three stimuli the subject was shown: a familiar face, an unfamiliar face, or a control.

3. Problem

The data was obtained from openfMRI, a website dedicated to the free and open sharing of neuroimaging datasets [7]. The data collection methods and equipment are described in [8]. The information is summarized here for convenience.

3.1. Task

The task began with the appearance of focus screen depicting a white cross centered on a black background for a random duration between 400 and 600 milliseconds (ms). One of three stimuli (a familiar face, an unfamiliar face, or a scrambled face) was then superimposed onto the white cross for a random duration between 800 and 1,000 ms. The subjects were then asked to press one of two buttons based on whether they thought the image was more or less symmetric than the average symmetry of a practice set the subjects had seen earlier. The experimental design groups each trial into one of three conditions based on the stimulus shown: familiar face, unfamiliar face, and scrambled face. This task was chosen because previous neuroimaging studies have shown that the brain activity differs when processing an unfamiliar faces versus a familiar face.

Figure 1. Stimuli examples. Familiar face (left), unfamiliar face (middle), scrambled face (right).

3.2. Data

There were 19 subjects. Each subject had 6 runs. Each run was 7.5 minutes long and had an average of 148 tasks.

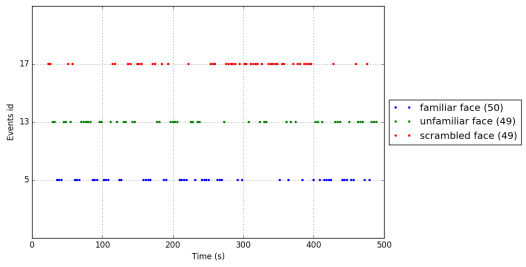


Figure 2. Distribution of 148 conditions of Subject 1 Run 1.

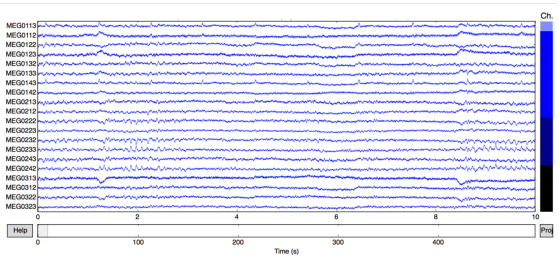


Figure 3. Raw MEG and EEG data from Subject 1 Run 1.

Subject 1 Run 3, Subject 4 Run 2, and Subject 6 Run 3 were corrupted, leaving us with 111 runs for a total of 16,354 events: 5,448 for the familiar face condition, 5,462 for the unfamiliar face condition, and 5,444 for the scrambled face condition.

There were 404 channels recording data at a sampling rate of 1,100 Hz during each run with a lowpass filter of 350 Hz. 71 channels measured EEG data. 306 channels recorded MEG data. The rest recorded eye movements, heart rate, environmental noise, head position, and stimuli presentation.

3.3. Preprocessing

All MEG and EEG (together: MEEG) data was run through Signal Space Separation to remove environmental noise. The MEEG data was then cropped into epochs that include the 500 ms before and 1,200 ms after stimulus onset. The epochs were run through a Savitzky-Golay 32 Hz low-pass filter to remove environmental noise. The first and last 400 ms were cropped to remove filter artifacts, leaving us with data from 100 ms before and 800 ms after event onset. At a sampling rate of 1,100 Hz, this is 991 time steps per epoch. This approach matches the preprocessing steps done during the technical validation of [8].

The epochs of each condition were averaged to create grand average Evoked Response Potentials (ERP) describing the neuronal activity that resulted from each of the stimuli. The ERP graph roughly matches the ERP graph created during the technical validation of the dataset in [8]. The difference can be explained by slight differences in preprocess-

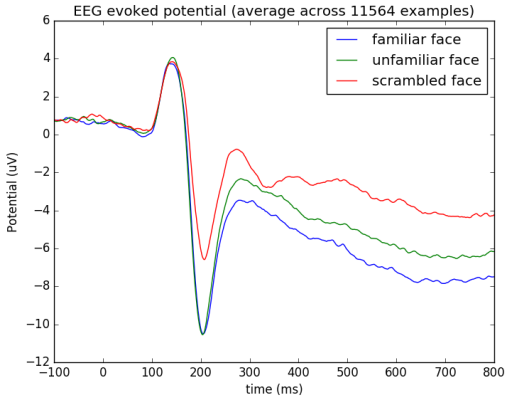


Figure 4. This author's grand average ERP graph (channel: EEG65).

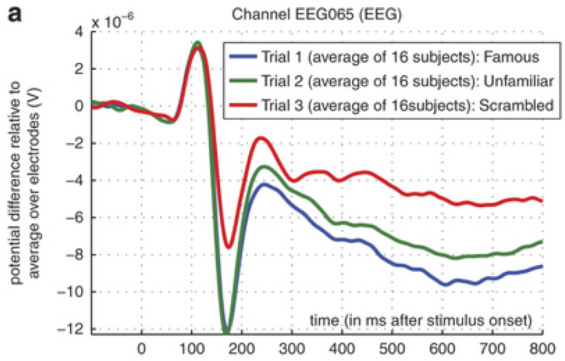


Figure 5. The data's original author's grand average ERP graph (channel: EEG65).

ing. The original authors mean-center the channels relative to the average across electrodes, which we perform later in the preprocessing step. Additionally, the original authors manually remove bad EEG channels and reference the remaining channels, which we're not doing to ensure all examples have the same dimensionality. Finally, the original authors reject all trials in which the amount of eye movement pass a threshold, which we're not doing because the eye and muscular movements don't change depending on the condition.

The ERP graphs for most EEG channels and most MEG channels showed a (varying) discernible difference between each of the three conditions. This demonstrates that the data is being segmented correctly and that the signals for each of the three conditions can be differentiated.

The data was split into 80% training and 20% testing sets. The training set was then split into a smaller 80% training set and a 20% validation set. This left 10,466 training examples, 2,617 validation examples, and 3,271 testing examples. The training data was centered around the mean and scaled to unit variance with respect to the average values for each channel over all of the training examples. These means and standard deviations were then used to center and scale the validation and testing sets.

The training data was then augmented. A common neuroscience practice, to save computation time, is to decimate the time steps by a factor of five. In other words, starting from the first timestep, only every fifth timestep is kept; the rest are thrown away. We repeated this with the training data five times, starting from the first, second, third, fourth, and fifth timestep. Each training example was thus split into five training examples, for a total of 52,330 training examples. Each validation and test example was decimated by a factor of five (but not augmented) starting from a random timestep between the first and the fifth, inclusive. The 991st time step was ignored in all cases so each of the remaining examples was 198 time steps long.

Additionally, eventual comparisons between EEG data alone, MEG data alone, and fused EEG+MEG data must be made depending on the type of data streams present, not the number of data streams present. To more easily take advantage of the spatial arrangement of the data using a convolutional neural network (more on that later), we want the input to be rearranged into a square. We thus randomly downsample the 71 EEG data to 64 channels, the 306 MEG channels to 64 channels, and the 71 + 306 EEG + MEG data to 32 + 32 EEG and MEG channels. The ratio of EEG and MEG channels is a hyperparameter that can be experimented with in future work.

We started our preprocessing with 19 subjects x six 7.5 minute runs with 404 channels recording an average of 148 events per run for 991 timesteps. We ended with 52,330 training examples, 2,617 validation examples, and 3,271 testing examples that each had 64 EEG channels recording information for 198 time steps.

4. Methods

Neural networks have produced an explosion of cutting-edge results in a variety of fields over the past four years. This has sparked interest in the neuroscience community as to whether artificial neural networks can be used to learn more about biological neural networks (e.g., the human brain) and/or to replace or supplement current neuroscience machine learning problems. This project explores how neural networks can be used to tackle the MEG-EEG fusion problem mentioned earlier.

4.1. Multi-Modal Models

Current multi-modal literature is sparse. There are three main papers that have been published in the past several years covering multi-modal deep autoencoders, multi-modal deep boltzman machines, and multi-modal deep belief networks [9–11]. The simpliest approach is described

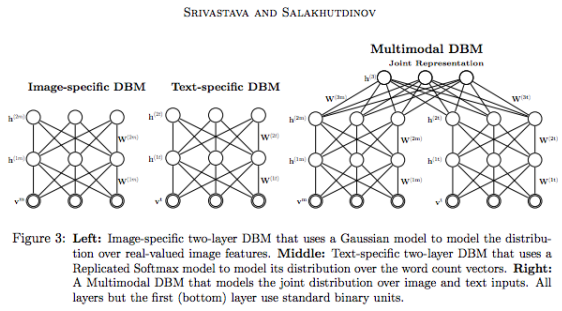


Figure 3: **Left:** Image-specific two-layer DBM that uses a Gaussian model to model the distribution over real-valued image features. **Middle:** Text-specific two-layer DBM that uses a Replicated Softmax model to model its distribution over the word count vectors. **Right:** A Multimodal DBM that models the joint distribution over image and text inputs. All layers but the first (bottom) layer use standard binary units.

Figure 6. Example Multi-Modal Deep Boltzmann Machine architecture. Credit: Srivastava and Salakhutdinov, 2012.

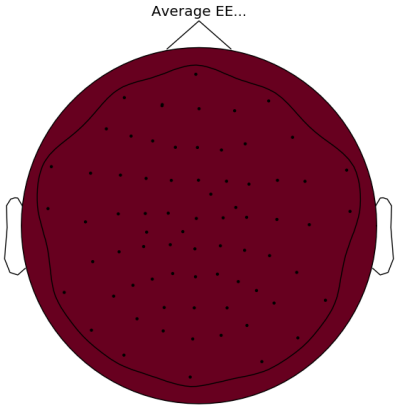


Figure 7. Spatial arrangement of EEG electrodes (MEG electrodes not shown for clarity).

in [10]: pretrain two neural networks, one on EEG data and one on MEG data, then combine the two neural networks using a joint representation layer, and then fine-tune the model using MEG and EEG data concurrently. The model is then able to classify better than if it used either modality alone and still retains the ability to perform classification if one of the modalities is missing. This approach requires designing a neural network architecture that can performs well when given a single modality (either EEG or MEG). This is the goal of the cs231n section of this project: design a neural network architecture that can classify EEG data well that can then be extended into a multi-modal architecture.

4.2. Baseline

This is a three-way classification problem. Random guessing is expected to predict the correct class 33% of the time. A perfect classification algorithm would be expected to predict the correct class 100% of the time. We need to verify that our classification problem can be approached using machine learning algorithms and to evaluate our neural network performs compared to other machine learning methods. We choose a common, linear machine learning algorithm called a Support Vector Machine (SVM) to serve a simple lower-bound machine learning algorithm.

Another option was to choose a current high-performing algorithm from the field of neuroscience. There is currently no clear state-of-the-art algorithms on the facial recognition task this data focuses on, though there are other tasks that have had neural networks applied to them. Additionally, other machine learning algorithms like Bayesian Network and Markov Chains have been applied to EEG problems. However, all current literature on working with this dataset is related to the multi-modal fusion of MEG and EEG data on the specific problem of trying to increase the accuracy of source localization using one modality as soft constraints on a Bayesian Network. Future work could include using other machine learning methods, choosing different datasets, or attempting to perform source localization with artificial neural networks. For the sake of time, this project is limited to comparing neural networks with an SVM baseline.

4.3. Recurrent Neural Network

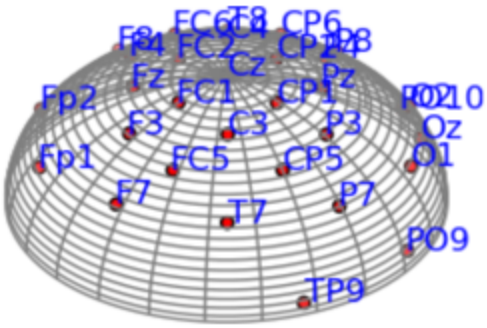
The data is temporally arranged (as time series data) and brain activity at one time step is related to brain activity at the next time step. In other words, information about past timesteps in addition to information at a current time step should help you learn more about what happens next. This type of problem is well-suited for a Recurrent Neural Network.

4.4. Convolutional Neural Network

The electrodes are arranged spatially over the subject's scalp. We can project their locations onto a simplified model of the skull (a hemisphere) and finding the groupings of k electrodes that minimize the total distance between electrodes within each group, as done in [12].

We can then arrange the channels based on their spatial groupings. For example, assume we found the k=4 groupings. We could then group the 16 best groupings (the 64 channels that compose the top 16 k=4 groups) into an 8x8 square, where each of the 16 squares contains the information for the spatially close four channels inside that group. This spatial arrangement makes the data well-suited for a convolutional neural network.

There are also multiple ways of transforming the data using Fast Fourier Transforms. The data over the entire time series can be grouped into a frequency vs power spectral density graph, where the nth frequency bin represents the (n * sampling rate / channel length) frequency (in Hz) and its corresponding power spectral density value that represents the energy at each frequency (in V^2/Hz) over the entire time series. This, however, eliminates the temporal nature of the data, likely reducing the overall predictive power of



(b) Spherical projection viewed from side, front of head at left.

Figure 8. Electrodes projected onto a simplified model of the skull (a hemisphere) (different dataset, source: [12]).

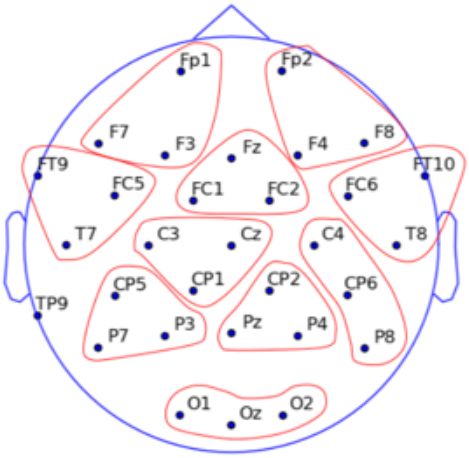


Figure 9. The 2D k=3 groupings of nearby channels (different dataset, source: [12])

the data.

An alternative is to transform the data into a spectrogram, a visual representation of the energy distribution over spectrum of frequencies in the signal over time. This effectively increases the amount of information known at each timestep, theoretically increasing the overall predictive power of the data at the cost of added computational complexity. This would transform the data from 64 channels x 198 time steps to 64 channels x 198 timesteps x N frequency bucket values per timestep. To reduce the number of dimensions of the data cube, the spectrograms of each channel could be concatenated (perhaps by nearest-neighbor groupings) or individual channels could be analyzed at a time.

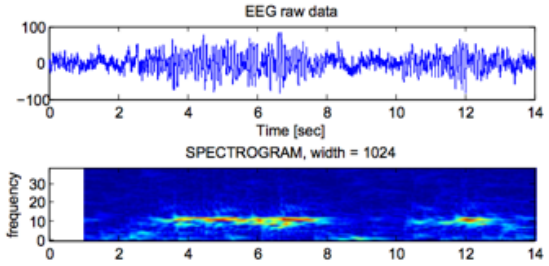


Figure 10. Example transformation of raw data into spectogram. Image credit: http://www.timely-cost.eu/sites/default/files/ppts/2ndTrSc/Niko Busch - Time frequency analysis of EEG data.pdf

4.5. Recurrent Convolutional Neural Network

The neural network model could alternatively take advantage of both the spatial and the temporal structure of the data and perform both spatial convolutions and recurrences. It could arrange the channels spatially, as in the CNNs, and then look at each of the timesteps individually. That is, the model could consider, at each timestep, an 8 channel x 8 channel data square. This data structure is similar to a video: a series of spatially arranged data squares (i.e. images) that are related to each other over time and looked at one frame at a time (i.e. frames in a video). This means the model could take advantage of all techniques currently being used to analyze videos, such as recurrent convolutional neural network models as described in [13], to analyze the EEG data.

Additionally, if the spectrogram format is found to increase the predictive power of the data at an acceptable computational cost, each ”image” could be a spectrogram representation, and the recurrent convolutional neural network model could look at an 8 channel x 8 channel x N frequency bucket values 3D data cube at each timestep.

4.6. Dimensionality Reduction

A popular dimensionality reduction technique is Principal Component Analysis (PCA), which uses an orthogonal transformation to transform your original, potentially correlated data points into a set of linearly uncorrelated data points (principal components). The number of principal components is strictly equal to or less than the original number of dimensions. This is useful for decreasing computational cost, but eliminates any spatial and temporal structure of the data. It was found to be unnecessary for the SVM baselines and as such was not used.

There is, however, an interesting spinoff of PCA that incorporates expert neuroscience knowledge: RCA [14]. This technique uses the same techiniques as PCA to significantly reduce the number of EEG or MEG channels in a dataset while keeping the total number of timesteps the same. In

other words, RCA could transform the original 71 EEG channels into ¡10 EEG uber-channels, each with a modified value that depends on what channels were combined where. This could be incorporated into future work if spatially arranging the data does not significantly increase predictive power.

5. Results

All work was done using Python, Keras, Theano, and a personal laptop. Data was stored in the hdf5 file format.

5.1. Baseline

We chose to do a simple linear model, a Support Vector Machine, as a baseline. To transform the data into a one-dimensional input, we simply concatenated all of the feature vectors. Here are the results.

Model	L1, SqH, P	L2, H, D	L2, SqH, P	L2, SqH, D
SVM Iter 1	45.9%	43.8%	42.5%	44.2%
SVM Iter 2	45.6%	42.9%	42.5%	43.0%
SVM Iter 3	44.6%	41.0%	42.5%	42.9%

Table 1. SVM baseline results The first two rows are 10 max iterations. The third row is 3 max iterations, to see how quickly the models approach their max value. SqH = Squared Hinge Loss, H = Hinge Loss, P = Primal kernel, Dual = dual kernel.

The L1 penalty, squared hinge loss, primal kernel SVM model consistently performed the best. We have our baseline: we want our neural network models to perform better than (hopefully significantly better than) 45%.

5.2. Recurrent Neural Network

We started by trying to take advantage of the temporal nature of the data by implementing a Recurrent Neural Network. Here are the results. The first row of accuracy results is training set accuracy and the bottom row is validation set accuracy (labels removed so the tables fit inside the column).

We began with a thrice-stacked Long-Short Term Memory (LSTM) model that used a categorical crossentropy loss and a RMSprop optimizer. We used the same hidden dimension (50) for each layer and set the initial learning rate (.01), rho (.9), epsilon (1e-06), batch size (128), and number of epochs (5) for the first model.

The model learns, but there's a big difference between training accuracy and validation accuracy, suggesting overfitting. Before we tackled overfitting, we wanted to see if increasing the learning rate by an order of magnitude helped the validation accuracy.

Messing with the learning rate didn't help the accuracy, although it did eliminate overfitting altogether. We turned it back down to .01 and focused on overfitting. We added dropout (p=.5) after each LSTM layer to in an attempt to reduce overfitting.

Ep1	Ep2	Ep3	Ep4	Ep5
41.3%	45.3%	49.8%	56.1%	61.4%
38.4%	39.6%	39.1%	38.4%	38.3%

Table 2. Training accuracy on top and validation accuracy on bottom, labels removed for formatting. 3 LSTM layers with hidden dimension (50), learning rate (.01), rho (.9), epsilon (1e-06), batch size (128), number of epochs (5).

Ep1	Ep2	Ep3	Ep4	Ep5
33.4%	33.1%	33.1%	33.7%	33.4%
33.6%	33.6%	34.8%	34.8%	34.8%

Table 3. Training accuracy on top and validation accuracy on bottom, labels removed for formatting. 3 LSTM layers with hidden dimension (50), learning rate (.1).

Ep1	Ep2	Ep3	Ep4	Ep5
40.5%	43.2%	45.1%	46.5%	48.3%
38.9%	40.7%	39.2%	39.9%	40.2%

Table 4. Training accuracy on top and validation accuracy on bottom, labels removed for formatting. 3 LSTM layers with hidden dimension (50) and dropout, learning rate (.01).

The gap between training and validation accuracies dropped significantly, but the overall accuracy was still low. We increased the hidden dimension of each layer to 100 in an attempt to increase the amount of information the model can store about the data to try to increase accuracy.

Ep1	Ep2	Ep3	Ep4	Ep5
39.6%	43.1%	44.7%	47.4%	50.6%
40.8%	41.5%	39.0%	41.1%	39.2%

Table 5. Training accuracy on top and validation accuracy on bottom, labels removed for formatting. 3 LSTM layers with hidden dimension (100) and dropout, learning rate (.01), rho (.9), epsilon (1e-06), batch size (128), number of epochs (5).

The increase in hidden dimensions didn't change the accuracy significantly. We reduced the model to one layer to see what affect that would have on accuracy.

Ep1	Ep2	Ep3	Ep4	Ep5
41.3%	48.5%	56.3%	62.9%	68.0%
39.6%	39.1%	39.4%	38.4%	38.2%

Table 6. Training accuracy on top and validation accuracy on bottom, labels removed for formatting. 1 LSTM layer with hidden dimension (100) and dropout, learning rate (.01).

The decrease in depth didn't decrease the accuracy significantly, but did exaberate the already-worrying overfitting problem. We returned to three layers and introduced L2 regularization to the weights, activity, and biases in an attempt to further decrease overfitting.

Ep1	Ep2	Ep3	Ep4	Ep5
37.7%	37.3%	37.4%	37.6%	37.7%
38.0%	35.4%	37.7%	36.7%	36.1%

Table 7. Training accuracy on top and validation accuracy on bottom, labels removed for formatting. 3 LSTM layers with hidden dimension (100) and dropout, learning rate (.01), regularization strength (.01).

Turning on L2 regularization eliminated overfitting, but didn't improve accuracy. It is unclear why the RNN model performed poorly. The success of LSTM models in literature and previous Kaggle competitions suggests this may be the result of poorly chosen hyperparameters. Future work will attempt to replicate past success on training RNNs on EEG data on other problems and run RNN code on a GPU cluster to accelerate training before coming back to co continue the RNN hyperparameter search.

5.3. Convolutional Neural Network

Instead of attempting to jump straight into optimizing a large neural network, we began working with CNNs by transforming the data into a spectrogram and focusing on a specific channel we knew had predictive power: Channel 65, the visual cortex-focused electrode graphed in Figure 4. We wanted to ensure this model had strong predictive power, so we didn't decimate the data. The width of the convolutions would then mean temporally adjacent data was considered together. Additionally, we thought part of the overfitting above may have been from the highly correlated nature of the augmented training data: splitting one example into five examples by decimating it may have just created five highly correlated training examples that contributed to overfitting. We were left with the original 10,466 training examples, 2,617 validation examples, and 3,271 testing examples. Each example was a data square 1 channel deep x 928 time steps long x 33 frequency bucket values tall. We then ran this through an untrained VGG CNN architecture with filters of size (3, 1).

Ep1	Ep2	Ep3	Ep4	Ep5
41.3%	48.5%	56.3%	62.9%	68.0%
39.6%	39.1%	39.4%	38.4%	38.2%

Table 8. Training accuracy on top and validation accuracy on bottom, labels removed for formatting. VGG-like convnet with (3, 1) filters.

6. Conclusion

The goal of my cs231n project was to architect an artificial neural network model that could perform a three-class classification problem on EEG data and that could be extended into a multi-modal architecture. Neither my recurrent neural network models nor my convolutional neural network model outperformed my SVM baseline, and I was never able to implement a recurrent convolutional neural network model. Previous success of RNNs and CNNs and rCNNS in related problems and in related datasets, however, convince me that this is because of my choice of hyperparamters, which were limited due to the sheer amount of time preprocessing this data required and my newbiness to both neuroscience projects and machine learning projects. I believe the best model to analyze EEG data is a recurrent convolutional neural network model, and I believe with another quarter's worth of work I will be able to implement some high-performing models and begin experimenting with multi-modal architectures.

7. Acknowledgements

This is a continuation of a Fall 2015-2016 CS 221 final project and is currently being pursued in conjunction with a CS 199 independent study with Professor Mykel Kochenderfer. Thanks to Professor Anthony Norcia (Psychology) for MEEG processing advice. Data preprocessing was done with mne-python, an open-source python MEG and EEG data processing library [15]. All code is publicly available under the MIT license here [16].

References

[1] S. Bunge and I. Kahn, "Cognition: An overview of neuroimaging techniques," in *Encyclopedia of Neuroscience*, L. R. Squire, Ed., Oxford: Academic Press, 2009, pp. 1063–1067, ISBN: 978-0-08-045046-9. DOI: http://dx.doi.org/10.1016/B978-008045046-9.00298-9. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780080450469002989.

[2] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography-theory, instrumentation, and applications to noninvasive studies of the working human brain," *Reviews of Modern Physics*, vol. 65, pp. 413–497, Apr. 1993. DOI: 10.1103/RevModPhys.65.413.

[3] B. Burle, L. Spieser, C. Roger, L. Casini, T. Hasbroucq, and F. Vidal, "Spatial and temporal resolutions of eeg: Is it really black and white? a scalp current density view," *International Journal of Psy-*

chophysiology, vol. 97, no. 3, pp. 210–220, 2015, On the benefits of using surface Laplacian (current source density) methodology in electrophysiology, ISSN: 0167-8760. DOI: http://dx.doi.org/10.1016/j.ijpsycho.2015.05.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167876015001865.

[4] S. Baillet, L. Garnero, G. Marin, and J.-P. Hugonin, "Combined meg and eeg source imaging by minimization of mutual information," *Biomedical Engineering, IEEE Transactions on*, vol. 46, no. 5, pp. 522–534, May 1999, ISSN: 0018-9294. DOI: 10.1109/10.759053.

[5] D. Sharon, M. S. Hämäläinen, R. B. Tootell, E. Halgren, and J. W. Belliveau, "The advantage of combining meg and eeg: Comparison to fmri in focally-stimulated visual cortex," *Neuroimage*, vol. 36, no. 4, pp. 1225–1235, Jul. 2007, 17532230[pmid], ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2007.03.066. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2706118/.

[6] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, "Brain-computer interfaces in medicine," *Mayo Clin Proc*, vol. 87, no. 3, pp. 268–279, Mar. 2012, 22325364[pmid], ISSN: 0025-6196. DOI: 10.1016/j.mayocp.2011.12.008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3497935/.

[7] e. a. Wakeman, *A multi-subject, multi-modal human neuroimaging dataset*. [Online]. Available: https://openfmri.org/dataset/ds000117/ (visited on 12/10/2015).

[8] D. G. Wakeman and R. N. Henson, "A multi-subject, multi-modal human neuroimaging dataset," *Scientific Data*, vol. 2, Jan. 2015, Data Descriptor. [Online]. Available: http://dx.doi.org/10.1038/sdata.2015.1.

[9] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[10] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *Journal of Machine Learning Research*, vol. 15, pp. 2949–2980, 2014.

[11] ——, "Learning representations for multimodal data with deep belief nets," in *International Conference on Machine Learning Workshop*, 2012.

[12] I. Walker, "Deep convolutional neural networks for brain computer interface using motor imagery," *IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE DEPARTMENT OF COMPUTING*, 2015. [Online]. Available: http://www.doc.ic.ac.uk/~mpd37/theses/DeepEEG_IanWalker2015.pdf.

[13] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3367–3375.

[14] J. P. Dmochowski, A. S. Greaves, and A. M. Norcia, "Maximally reliable spatial filtering of steady state visual evoked potentials," *Neuroimage*, vol. 109, pp. 63–72, 2015.

[15] M. Developers, *A community-driven software package designed for for processing electroencephalography (eeg) and magnetoencephalography (meg) data*. [Online]. Available: http://martinos.org/mne/stable/index.html (visited on 12/10/2015).

[16] P. Warren, "Neurofusion," [Online]. Available: https://github.com/pawarren/neurofusion.