Project Final Report

Professor John Canny

CS 194-16 Introduction to Data Science

14 December 2015

Team: Brainiacs

Neha Verma, Sayan Gul, Hammad Bashir, Ernesto Ezequiel Perez

Predicting Brain Activity Patterns from Visual Stimuli

Problem Statement and Background

The two fundamental problems related to brain imaging data are called encoding and decoding. Encoding takes stimuli as input (e.g. natural images) and attempts to predict the brain activity in response to the given stimuli, whereas decoding tries to reconstruct the stimuli based on the measured brain activity. In our analysis, we mainly focused on the encoding problem; however, these two problems are fundamentally equivalent. Formally, the equivalency can be shown by the following equation:

$$P(f(S)|R) \propto P(R|f(S))P(f(S))$$

In this equation S refers to stimuli and R refers to the response. It is important to note that the probability is written in terms of $f(S)$ instead of S. This is because the brain receives the stimuli through sensory organs which apply nonlinear transformations to the raw stimuli. The above equation can be explained in the following way. The probability of a stimulus given a response (which decoding attempts to find) is proportional to the probability of a response given the stimulus (which encoding attempts to find) multiplied by the prior probability of the given stimulus.

The only remaining issue here is that finding the prior probability of a given stimulus requires us to consider a large space of possible stimuli, which makes decoding a more difficult problem than encoding. For example, for our particular instance, the prior would be all natural images, which constitutes an unreasonably large data space.

The data we used in this project was collected by the Gallant Lab at UC Berkeley, and it was provided to us by the course staff. This dataset, along with many other neuroscience datasets, is also accessible to the scientific research community through Collaborative Research in Computational Neuroscience-Data Sharing group located at http://crcns.org/.

The data consists of two datasets which were extracted by recording two subject's brain activity in response to natural images. Brain activity was measured using blood oxygenation level-dependent (BOLD) signals measured by functional magnetic resonance imaging (fMRI) to capture the volumetric absorption of oxygen, which is greater in areas of the brain that show the most activity.

The stimuli data was composed of 1750 500x500 pixel images for the training set and 120 500x500 images for the validation set. The BOLD fMRI data contained information from voxels in seven areas of the brain: V1, V2, V3, V3A, V3B, V4, and the Lateral Occipital area, as well as some voxels that fell outside these areas labeled as "other". The aggregate volume of these Regions of Interest was broken down into 25,915 voxels. Each subject's BOLD fMRI training data consisted of a 1750x25915 matrix, where the i th row corresponded to the voxel response for the i th image, and the j th column corresponded to the data of the j th voxel. Similarly, the validation data consisted of a 120x25915 matrix. The data stored in each element of these matrices was the z-score of a given voxel response to a given image.

We evaluated encoding models using R-squared values, and cross validation error as a measure to choose the best hyperparameters for our models.

This problem is of great significance in many fields such as neuroscience, psychology, medicine and biomedical engineering. Developing better models to encode and decode brain imaging data could lead to a better understanding of how the brain interprets, stores, and processes different stimuli, which could be used in many applications such as brain-machine interfaces. Furthermore, research on the reconstruction of dreams from brain activity, although unsuccessful until now, could shed light on whether dreams are processes different from visual stimuli.

Both encoding and decoding methods are extensively explained in Professor Jack Gallant's talk on fMRI enabled brain reading (2012) . We also used "Identifying Natural Images From Human Brain Activity" by Kay et al. (2008) for ideas on how to approach the data analysis. In this paper, authors used a Gabor pyramid model for transforming the images. Following this, they multiplied the features generated from this transformation by a weight vector to predict the voxel responses. They updated this weight vector using Gradient descent with early stopping. We also explored ideas from several other papers which worked on similar problems. (Chen et al., 2014; Nishimoto et al., 2011; Naselaris et al., 2009).

Methods

Data Pipeline:

- Data Collection:

As mentioned before, the data was provided by the course staff through the Gallant Lab at UC Berkeley. The fMRI data itself was collected at the Brain Imaging Center at UC Berkeley by measuring the BOLD responses of two subjects as they were shown natural images across multiple sessions.

- Cleaning and Repair:

The dataset published by the Gallant Lab was already cleaned and preprocessed to a certain degree. Every brain volume that was gathered was aligned spatially to the first session to achieve consistency. After this, the peak BOLD response to each image for each voxel was estimated from the aligned data. Finally, this data was z-scored for each voxel. So in our dataset, a voxel's response to a particular image refers to how many standard deviations above the voxel's mean response the response to the specific image is.

We rescaled the stimuli images to 200x200 pixels, 150x150 pixels and 96x96 pixels. This allowed for faster feature extraction while ensuring that image characteristics were still distinguishable.

For input into a convolutional neural network the images were scaled down to 96x96 to speed up training and also to conform to common network models from Caffe. The voxel data contained NaN values corresponding to lack of data for the voxel in question. These values were dealt with in different ways for different models used which we will discuss in tandem with the models further on. Also, the pre scaled down stimuli file in our dataset was corrupted so we had to reconstruct the 1,750 unique stimuli by going through 17 separate data files and scaling them to the desired size for models.

- Transformation:

For featurization we first tried raw pixel data. Due to the nature of the data and the nonlinear way in which the brain processes images, we later used Gabor filters to approximate this nonlinear transformation. We generated quadrature pairs of Gabor filters at 5 different scales: 200x200, 100x100, 50x50, 25x25 and 10x10, with sets of 6 and 8 orientations. After filtering an image through convolution using each filter in the Gabor filter bank, we obtained a deconstruction of the image consisting of a set with size equal to 5 (number of scales) times the number of orientations (6 or 8) times two (quadrature pairs). From each filtered image we extracted the local energy and mean amplitude, for a total of 120 features per image when using 6 orientations and 160 features when using 8 orientations.

We also obtained transformations in the form of filters by training a Convolutional Neural Network on the training set. We tried various network architectures, giving us kernels of size 64x64, 32x32, 11x11, 5x5 and 2x2. With the number of kernels increasing with the depth of the layer and the size decreasing respectively as well.

We also transformed the data to a per-region collection of voxels instead of 25,000 voxels spread across the brain. To do this we used index to voxel mappings given in the dataset to extract the voxels corresponding to each region.

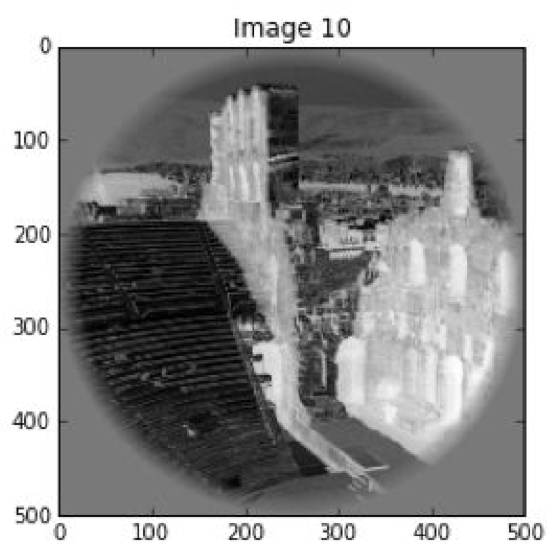
- <u>Analysis:</u>

We first tried a naive approach by using raw pixel data as input (40000 pixels per image) and a voxel set response as output (a 25915 voxel set response per image) for a Linear Regression model. As expected this yielded negative r-squared scores showing the input was highly uncorrelated to the output data. Because of the high dimensionality this was by far the most time inefficient approach. For this model we transformed NaN values to 0, in order to keep as much voxel information as possible.

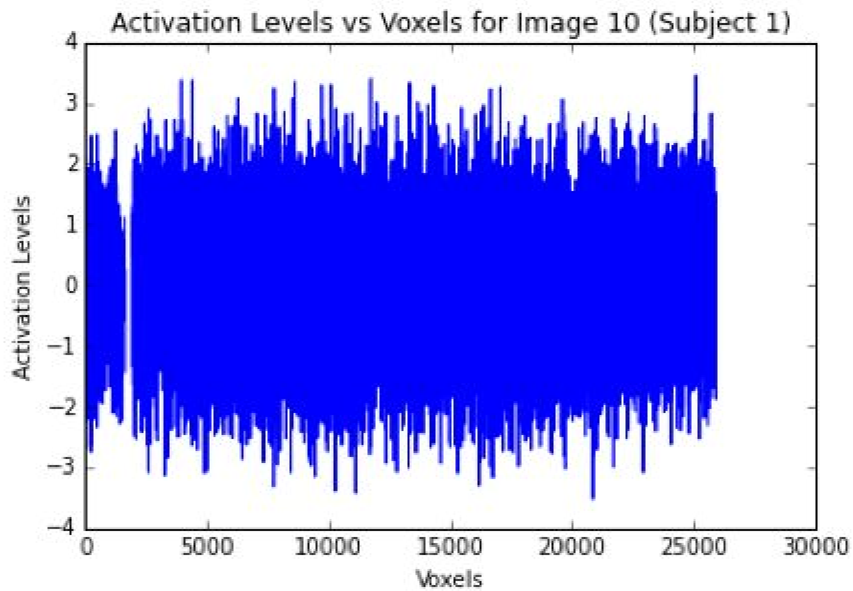
For all other regression approaches, we used Gabor filter featurization to reduce the dimensionality of the input data (120 and 160, depending on number of orientations) and correlated it to the brain activity data. We performed regression on each voxel separately using as input the featurized images and as output the data for the jth voxel. During training and validation we skipped training samples where the voxel contained NaN values. We followed this methodology for Linear Regression, SVR and Neural Network models.

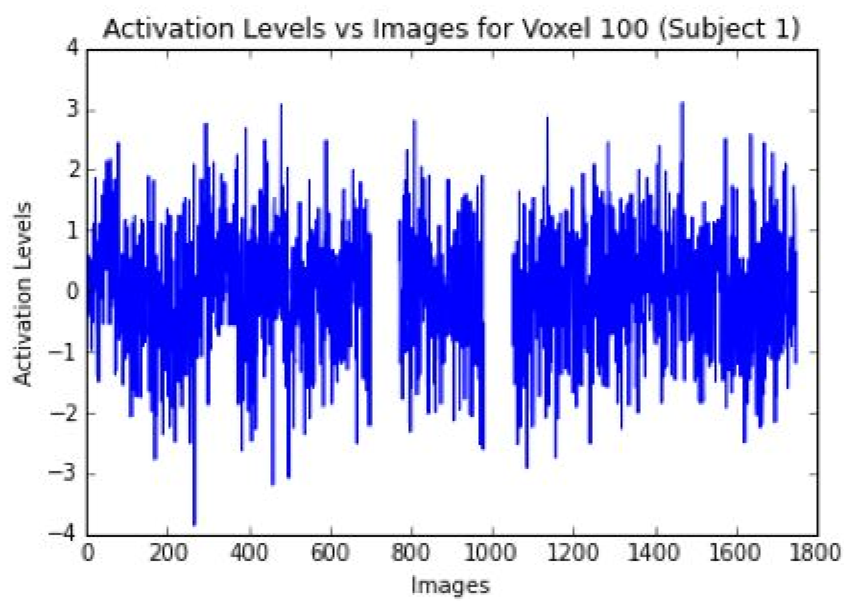
- <u>Visualizations:</u>

We started our visualizations by using matplotlib to show the image stimuli which were shown to subjects. These images were in multiple different “.mat” files, which we combined into one. The images looked as following:

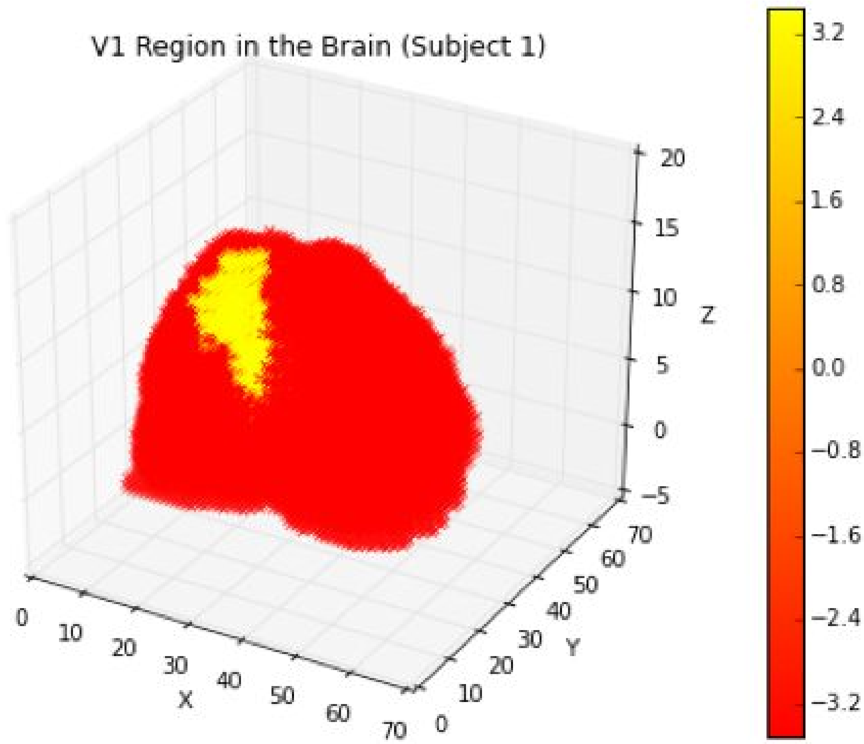


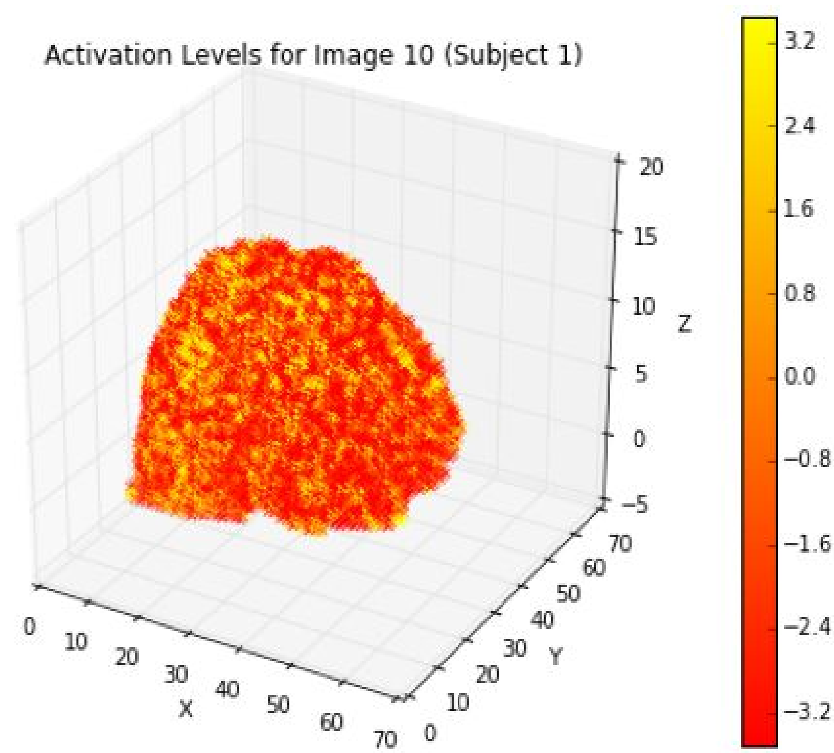
We attempted to visualize the estimated response data in two main ways. We tried to plot the different voxel responses for a given image, and we tried to plot the different responses by a given voxel to different images. These plots usually didn't pertain too much information:





Next we tried a different approach. We used the index mapping from voxels to a 64x64x18 cube that was provided in the dataset in order to visualize the activations on the brain itself. This gave us much better visualizations that were much easier to interpret. We were also able to use this approach to visualize regions of interest:





Parameters

The best results were obtained by rescaling images to 200x200 pixels, filtering with Gabor filter banks with scales: 200x200, 100x100, 50x50, 25x25 and 10x10, with 6 orientations, and extracting mean amplitude and local energy. The best model was Linear Regression from the scikit learn python library. We tried a custom made Neural Network (implemented originally for digit recognition, cs189 Fall 2015) with one hidden layer consisting of 50, 100 and 200 nodes. We tried Sigmoid, softmax and ReLu activation for the hidden layer and tanh and linear activation for the output layer. We tried Support Vector Regression from the scikit learn python library with rbf and polynomial kernels.

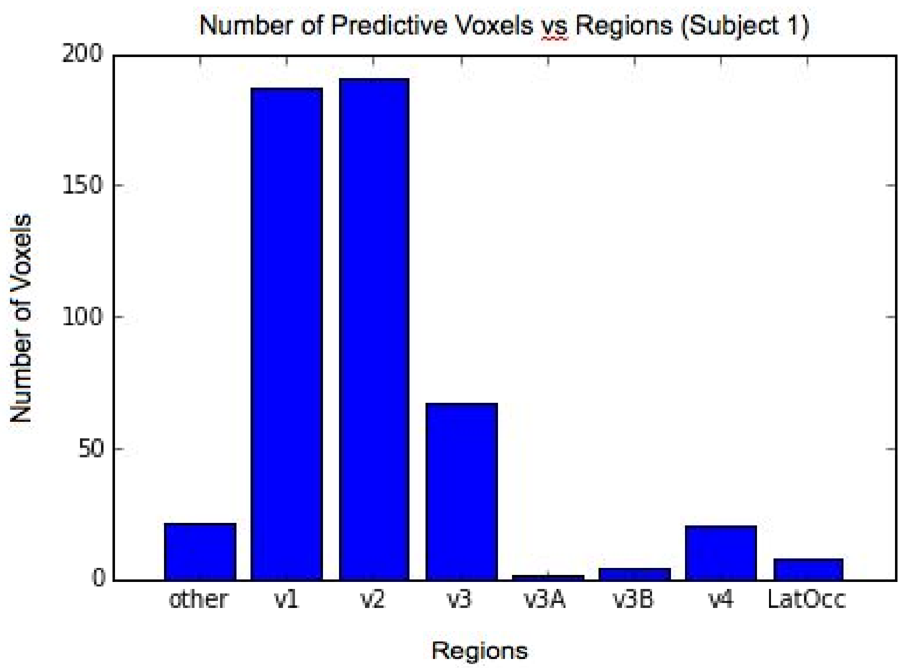
Our convolutional neural networks were implemented using the NoLearn library in python which is a easy to use wrapper around the Theano based neural network library Lasagne. Input images were scaled to 96x96. We tried multiple network layouts as well various hyperparameter configurations to prevent overfitting and introduce some regularization. In the end we settled on 5 convolutional layers with pooling layers stacked in between, in addition to a dropout layer before our last 1D layer. All hidden layers used leaky ReLUs. Lastly, we added L2 regularization to the network. Deeper networks would have been more effective (as most industry research has shown) but our lack of access to more powerful GPUs prevented this. Deeper networks were used but took too long to train to prove useful for iterating over multiple designs. We trained our networks using ADAGRAD on the visual cortex data.

Prior work and other other groups

It is difficult to compare the performance of our model to prior work performed by Gallant Lab since they devised their own performance measure as opposed to using r-squared value as we did. We had a different approach to the dataset than our classmates, as they seemed to focus on finding a semantic representation for the image stimuli and comparing the response across subjects, whereas we focused on finding a good encoding model to predict brain activity given a visual stimuli.

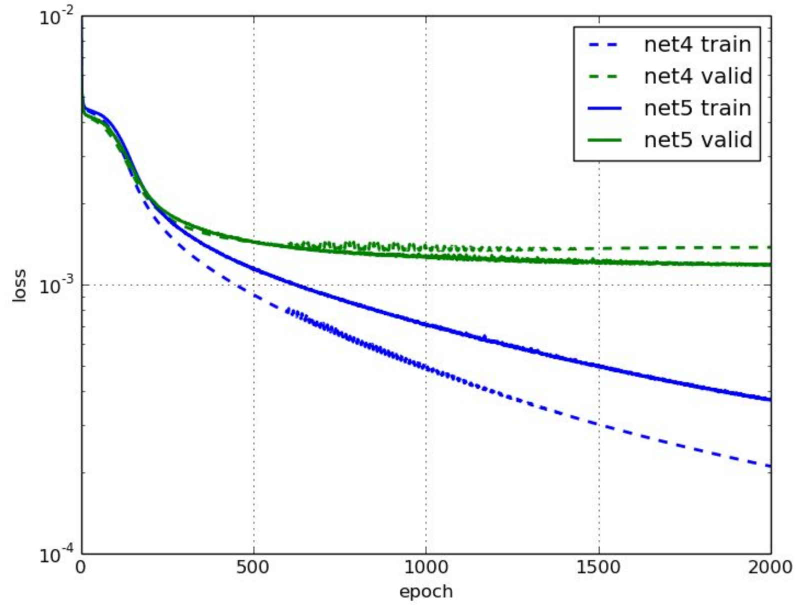
Results

Our best result based on r-squared values was achieved through Linear regression of featurized images. The highest r-square value achieved was of 0.44 corresponding to voxel 15,872. For brain activity and similar data an r-square value of 50% is common, since it is harder to do regression on this kind of data. The following graph shows the distribution of the 500 voxels with greatest r-squared value by region of interest (Subject 1). Based on this we came to the conclusion that the regions V1 and V2 play the most important role in predicting brain activity.

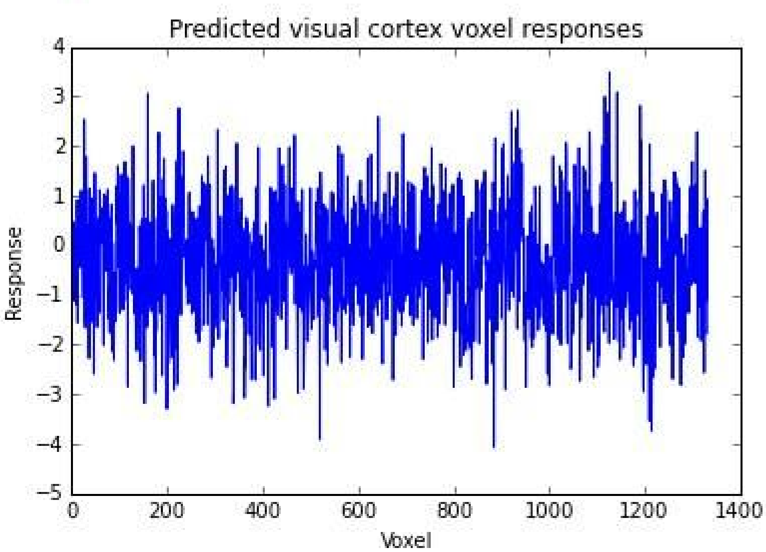


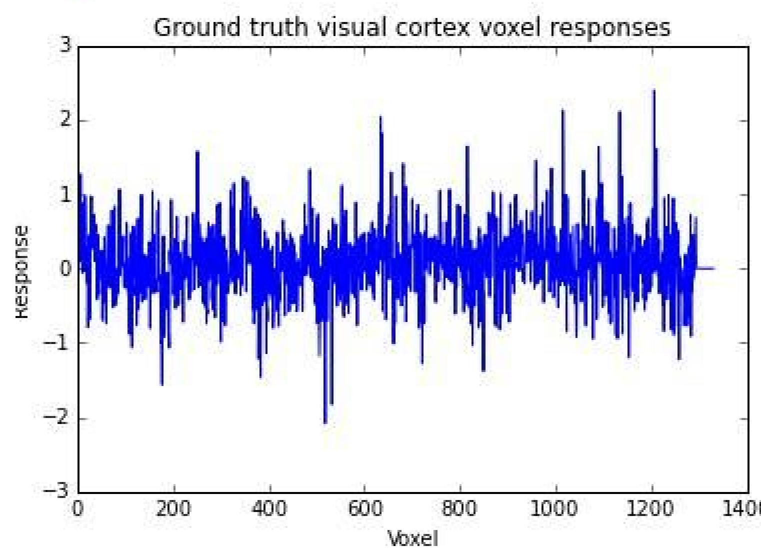
The convolutional neural network we implemented in NoLearn had an average r-squared value of .42 across all voxels. It models the overall structure of the response fairly well but lacks accuracy on the intermediary data in between larger activations. Our model took 12 hours to train on a GPU with 512 CUDA cores. The following chart shows the training history of an initial net we used without any dropout or regularization. Adding dropout and regularization led to less overfitting and better loss

(Dotted is old model)



Here is a predicted response to a particular stimuli on subject one:





We see that the structure of the response is fairly well modeled (by the primary large spike activations). In fact activations above a threshold of |1.5| were predicted with 80% accuracy, demonstrating that the CNN picks up on the overall structure of a brain response.

Tools

The tools we have used can be listed in the following groups:

- Data interface: HDF5 for Python (h5py)

  The data provided to us was very large in size, stored in .mat format. We chose h5py to extract the data since it's well documented and easy to use.

- Exploratory Data Analysis and Data Manipulation: SciPy, NumPy

  We used both packages in doing numerical analysi. We chose these libraries for their power, ease-of-use, and our past familiarity with them.

- Image Processing: Scikit-Image, OpenCV

  We decided to use Gabor filtering to transform our raw images. This was motivated by the scientific evidence that suggests neurons in the visual cortex have Gabor-filter type receptive fields. We tried both Scikit-Image and OpenCV Gabor filter generators, as both of them are well documented and easy to use for image processing. Our final implementation made use of OpenCV to generate the Gabor filter banks.

- General Machine Learning Models: Scikit-Learn

  We used several scikit-learn's ML models (refer to Parameters section).

- Neural Network Models: PyBrain, Custom Implementation, NoLearn, Theano, Lasagne, Caffe

  We first tried PyBrain for Neural Networks, but it was very unintuitive and lacked good documentation. We then tried a custom implementation we had coded during the semester for cs189 class, and modified it accordingly to perform regression.

We utilized NoLearn (on top of a theano based lasagne) and Caffe for convolutional neural networks. We used NoLearn for its easy to use scikit style layer declaration that allowed us to create CUDA optimized neural networks (in Theano) with an easy API. We used Caffe for its extensive Model Zoo of pretrained models for feature extraction.

- Visualization: Matplotlib

  We used Matplotlib to do visualizations of our findings. Matplotlib gives a concrete set of tools for mapping both 2D and 3D spaces, with a variety of options for subplots, legends, scales, and graph appearances and was easy to use in conjunction with our python machine learning stack.

Lessons Learned

We learned that complexity does not always mean better performance. Out of all the approaches we tried for general machine learning models, it was surprising to find out that Linear Regression gave us the best results. Using Gabor filtered images as opposed to raw pixel data emphasized the importance of featurizing input data in a way that strongly relates it to output data. It was also surprising to see that out of the seven regions of interest only two have a strong influence in brain activity prediction.

Using convolutional networks we were able to achieve very good results as it can be seen in our results section. One surprising result was the higher amplitude in our predicted signal. Exploring this result, we realized that current imaging technologies lack the necessary spatial and temporal resolution to measure brain activity in the necessary level of detail.

In conclusion, we can confidently say that neuroscientific research is one of the most promising fields as we are combining improving machine learning methods with the advancements in brain imaging.

Team Contributions

- Sayan: Sayan's main contributions were in providing domain knowledge and planning general approaches during analysis. He also worked on data processing and visualizations.

- Neha: Neha’s main contributions were in presentations. She designed the poster, as well as the slide show for our presentation.
- Hammad: Creating convolutional neural network models, training, testing, and optimization.
- Ernesto: Linear Regression, SVR and Neural Network models. Gabor filter featurization.

References

Chen, M., Han, J., Hu, X., Jiang, X., Guo, L., & Liu, T. (2014). Survey of Encoding and Decoding of Visual Stimulus via FMRI: An Image Analysis Perspective. *Brain Imaging and Behavior*, *8*(1), 7–23. http://doi.org/10.1007/s11682-013-9238-z

Gallant, J. L. (2012). fMRI Enabled Brain Reading. Retrieved from https://www.youtube.com/watch?v=ZZSJqsMtk1k

Kay, K. N., Naselaris, T., Prenger, R. J., & Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, *452*(7185), 352–355. http://doi.org/10.1038/nature06713

Naselaris T., Prenger R. J., Kay K. N., Oliver M. & Gallant J. L. (2009). Bayesian reconstruction of natural images from human brain activity. Neuron 63, 902–915

Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., & Gallant, J. L. (2011). Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, *21*(19), 1641–1646. http://doi.org/10.1016/j.cub.2011.08.031