PROJECT REPORT

ON

PREDICTING IMAGE CATEGORIZATION FROM FMRI DATA

CS 6375 Machine Learning

Term Project

Spring 2017

By

SUSMITHA SRI KADIYALA-sxk163530

SANJANA VIJAYKUMAR – sxv163930

1. INTRODUCTION

Our perceptive abilities are truly incredible. When we see a house, regardless of its color, size, or shape, we know it's a house. As obvious as this seems, most of us will take these abilities for granted. For instance, what if we couldn't distinguish cars and buildings? What if we couldn't even recognize friends and family, and mistook them for strangers? This condition is termed agnosia, and it's easy enough to imagine how different our lives would be if we couldn't perceive properly. With how remarkable our brains are, it's a wonder that we know so little about neuroscience. However, in recent years, much more focus has placed on this field, with object recognition at the forefront. While major theories have developed about how the brain processes information, nothing has been officially proven.

There is evidence that suggests object recognition is attributed to modularization in the ventral temporal area (meaning certain areas of the brain are responsible for recognizing different stimuli), but there are also theories for it being a result of distributed systems (meaning all areas of ventral temporal cortex contribute in object recognition).

2. PROBLEM DEFINITION

2.1 Task Definition

For this project, we decided to implement machine learning algorithms, to see if we could find correlations between specific areas of the brain and object recognition processes. Specifically, we wanted to see if we could predict the visual stimuli a subject was looking at given their fMRI scans while viewing the image. fMRI scans measure blood flow across the brain, which correspond to brain activity/stimulation. If we are correctly able to predict visual stimuli, our results could prove or at least further support the modularity of our brain in object recognition. Creating algorithms to achieve these results could also have far more practical effects as well, such as enabling communication with the physically impaired. If someone's in a coma, fMRI scans could give us crucial information about the state of thoughts of that person.

2.2 Feature Set

The examples in the data were split across four subjects. Each subject was given one of five visual stimulus categories (faces, bodies, houses, words, cars) while performing one of three tasks (Odd Ball, Selective Attention, Working Memory - the details of which are not important to our project) and asked to perform three trials, for a total of 180 examples.

2.3 Data Processing

For each example, the fMRI data we were given was formatted as a 1973 element vector, each representing a voxel (specific area of the brain) and the corresponding amplitude of blood flow to that voxel (brain activity). As a part of processing the data and to allow for equal weighting of each voxel number, we normalized the mean and variance of each voxel, as follows:

1. $\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$
2. Replace each $x^{(i)}$ with $x^{(i)} - \mu$
3. Let $\sigma_j^2 = \sum_i \left(x_j^{(i)}\right)^2$
4. Replace each $x_j^{(i)}$ with $x_j^{(i)}/\sigma_j$

After normalizing the data, we noticed that the normalized data added more variance and noise to each individual example. Therefore, we wanted to eliminate this unwanted noise through data smoothing. Because, our voxels are presumably dependent on their neighboring voxels, we modified the amplitude of a specific voxel to be a weighted average of the closest C voxels by index. More specifically, For every voxel v_i in a data example x:

$$v_i := \frac{\sum_{j=0}^{C} \frac{v_{(i+j)} + v_{(i-j)}}{j+1}}{2C}$$

The noise reduction technique could vastly reduce the amount of deviation in the data, which we hoped would help prevent or limit the magnitude of overfitting for our different algorithms. Our data also had many features (1973), especially when compared to the number of training examples. We ran principal component analysis to reduce the feature

space. However, it did not improve results for any number of principal components. The likely cause of this is the amount of noise
produced by the process of fMRI scanning.

3. ALGORITHMS

3.1 Logistic Regression:

We began by running logistic regression to provide a good baseline in categorizing our fMRI data for future algorithms. Because we knew we would implement a neural network later, we implemented logistic regression as a neural network without any hidden layers, so that we were just optimizing one synapse (or weight vector of the features) with a sigmoid function (z) where:

$$\sigma(z) = \frac{1}{1 + e^{(-z)}}$$

Then, to compute the optimal weight vector, we ran stochastic gradient descent until convergence by:

$$\theta_j := \theta_j + \alpha(y^{(i)} - \sigma(\theta^T x^{(i)}))x_j^{(i)}$$

To classify the testing data, the assignment is simply equal to the sign(θ^Tx). For a given subject, we trained our weights across all examples (all five categories and all three tasks) corresponding to the first two runs and tested on examples corresponding to the third run. Then, we ran our algorithm within all subjects. So, we trained again on the first two runs, but now incorporated all subjects, and then tested on the third run for our examples. We correctly classified about 82.22% of our testing examples. We then decided to run our algorithm across subjects. We trained on all subjects except for one, and then tested on the one we left out.

3.2 Neural Network:

We implemented a three-layer neural network (input, hidden layer, output) through backpropagation training, compared to logistic regression which is essentially a two-layer network (input, output). To test how well our regression fits the data, we considered pairs of voxels as features, which would be more appropriate for non-linear data. This also makes sense on a broader level because areas close together in the brain are likely to have similar functionality, if we assume the brain is at least somewhat modularly structured. At each intermediate level h_j and corresponding weight vector v_j:

$$h_j = \sigma(v_j^T x)$$

Then, to produce a final score:

$$score = w^T h$$

The assignment of the testing example x given the final weight vector w is equal to:

$$y^{(i)} = sgn(\sigma(w^T h))$$

The average accuracy of classification obtained is 79.44%.

3.3 K-Nearest Neighbors:

For K Nearest Neighbors, given an unclassified example, we search for the most similar example in the dataset and classify the test example as its nearest neighbor. When determining the nearest neighbor, we used the Euclidean distance to measure this closeness where

$$d(i,j) = ||x^{(i)} - x^{(j)}||_2^2.$$

We extend this to look at the k-closest neighbors and choose the image category that appeared most frequently within the k-closest neighbors for a range of k-values. More specifically, we selected the top k-closest neighbors to the current example and then selected the image category of one of the examples randomly with probability weights equal to 1/d(i,j). The average classification rate obtained is 23.61%.

3.4 Gaussian Naive Bayes:

Our implementation of the Gaussian Naïve Bayes algorithm attempts to categorize a new vector of fMRI data by choosing the image category that had the highest probability given the newly observed training data. One strong assumption Naive Bayes makes is that for a given image category, it assumes each feature is independent from every other feature, thus making the likelihood of the data:

$$p(y) = \prod_{x \in X} p(x, y)$$

Furthermore, since p (x, y) = p(x|y) *p(y) and in our case each image category was equally likely to occur thus making p(y) a constant, and p(x) is unchanging, we can simplify p(y|x) =p(x|y) p(y)/p(x) to p(y|x) = p(x|y) for all voxels. Now for the probability of a voxel's amplitude given a category, we assumed the voxel's amplitudes were normally distributed within an image category and therefore the calculated probability of voxel x having amplitude μ as:

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

For binary classification when dividing our splitting our dataset by trial run, we observe testing errors of less than 20% overall, and error rates of less than 10% when also conditioning an individual subject. For general image classification, the result was a testing error less than 40%. Average classification rate obtained is 83.75%.

4. CONCLUSION

Our feature set contained 1973 different voxels, however, many of these voxels we determined early on were similar and therefore can be grouped together as more dense features. We ran PCA initially to try and remove this variability across similar voxels, but there was extra noise due to task and subject variability, and so we could not identify these groupings. We could identify that voxel groupings were better features when we used a neural network. Like many other fMRI studies, our data has shown to be highly subject dependent as all our classification algorithms worked better when we had training data and test data from the same subject. We attribute the differences across subjects to two primary causes. The success rates we achieved are on par with many of the current classification algorithms in the field, but we were limited by the amount of data we had, more data could have pushed this classification rate up or at least, in the case of our neural network, establish a more concrete and confident bound on what classification rates can be achieved by better fitting the data.

5.REFERENCES

[1] Cox DD, Savoy RL. Functional Magnetic Resonance Imaging (fMRI) "Brain Reading": Detecting and Classifying Distributed Patterns of fMRI Activity in Human Visual Cortex. NeuroImage, 2003.

[2] LaConte S, Strother S, Cherkassky V, Anderson J, Hu X. Support vector machines for temporal classification of block design fMRI data. NeuroImage. 2005

[3] Haxby JV, Gobbini MI, Furey ML, Ishai A, Schouten JL, Pietrini P. Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science. 2001

[4] Mitchell TM, Hutchinson R, Niculescu RS, Pereira F, Wang X, Just M, Newman S. Learning to Decode Cognitive States from Brain Images. Machine Learning. 2004

[5] Trask Andrew. A Neural Network in 11 lines of Python (Part 1) http://iamtrask.github.io/ 2015/07/12/basic-python-network/

[6] Liang Percy. CS 221 Course Material

[7] Ng Andrew. CS 229 Course Material.