Generalized Reconstruction of Visual Stimuli from fMRI Data

Islam, Russ
rmislam@stanford.edu

Kim, Min Cheol
mincheol@stanford.edu

Lee, Steven
slee2010@stanford.edu

Winter 2014-15

Abstract

The relationship between visual information and neural activity is central to the field of computational neuroscience and has the potential to advance the development of brain-machine interfaces. In this paper, we explore methods used to map neural activity, in the form of fMRI signals, to visual stimuli, in the form of black and white 100 pixel images. First, we replicate the current leading technique, which trains many logistic classifiers that map the fMRI data to blocks of pixels in the stimulus image space. Next, we implement novel methods, including a linear classifier, several support vector machines (SVM), and several neural network architectures and compare our results with the current leading technique. We find that while linear classifiers and SVMs are not very effective at modeling the relationship between fMRI activity and visual stimuli, the neural network performs better, although not as well as the leading technique.

1 Introduction

Problem Description and Motivation

The question of how sensory information is encoded in neural signals is central to the field of computational neuroscience. In particular, the mapping between stimulation of the retina and activity in the visual cortex is not well understood, though the literature tells us that the relationship is most likely highly nonlinear and suffers from a substantial amount of noise. This makes machine learning and computer vision techniques particularly well suited to discovering a method to decode visual stimuli from their fMRI patterns.

In 2008, a group of researchers (Miyawaki, et. al.) from the Japanese company ATR (Advanced Telecommunications Research Institute International) published a paper in Neuron that presented their generally successful attempt to apply local decoders trained using multinomial sparse logistic regression to uncover this mapping. Test subjects were shown a series of 10-by-10 pixel images, where each pixel could be either black or white. Scans of their visual cortex were conducted as the subjects watched these images, and after training the reconstruction algorithm on this set, the researchers used the trained algorithm on real-time fMRI data to reconstruct what the test subjects saw as they watched a second set of images. During the training of the algorithm, each pixel in the 10-by-10 stimulus image was given a number between 0 and 1 to represent the "grayness" of that pixel predicted using the entire fMRI scan taken while the subject viewed that image. This was repeated for each of the 100 pixels, and training was done over several images. In fact, Miyawaki, et. al., went beyond this and used not only single pixel patches, but used 1-by-2, 2-by-1, and 2-by-2 pixel patches as well and selected the optimal combination of these four basis patches to reduce the reconstruction error (see original paper for details).

ATR has made their dataset as well as their code for the reconstruction algorithm publicly available, so rather than only replicating their paper, we explored a number of alternative techniques to compare the merits and demerits of each one against the method used by Miyawaki, et. al., and perhaps to reduce the reconstruction error to below what was reported in their paper. Our project is an image reconstruction problem that is cast as a classification problem (each pixel in the presented image is either black or white, and since there are 100 pixels, there are 2^{100} classes). We are motivated by the opportunity to understand more deeply the encoding and decoding schemes used by the brain to represent visual data, and such

an understanding is particularly important for brain-machine interfaces to become a commercially viable technology. We also believe this reconstruction problem is somewhat unusual compared to those presented in class or done in previous years' projects, so the unfamiliarity of this problem formulation is especially exciting.

2 Previous Work

2.1 Review of Previous Work

Miyawaki, et. al.'s seminal work on visual stimuli reconstruction via fMRI trained "local decoders" that assigned weights associating local, small fMRI patches to pixels in the visual stimuli space. The group used sparse logistic regression to optimize these weights on randomly generated visual stimuli and obtained average reconstruction errors ranging approximately between 0.2 and 0.25 (measured by the pixel-by-pixel mean square difference between the estimated image and true image). At their multiscale stage (i.e. using a weighted combination of decoders for various patch sizes), Miyawaki, et. al. obtained a reconstruction error of 0.2. The group was able to demonstrate the efficacy of trained local decoders that are linearly weighted to produce a predicted output visual image. The outline of their method is as follows:

1. Data Preprocessing: After the experiment was conducted on the subject and the fMRI data was recorded, the stimuli images were represented using four different basis pixels. The original stimuli images were 10-by-10 pixels each, with each pixel taking on a value of 0 or 1. This is the 1-by-1 pixel basis representation. The stimuli images were then represented by 1-by-2 pixel blocks, where each block was assigned the average value of its component pixels (i.e., if the block consisted of two white pixels or two black pixels, the entire block was assigned a label of 1 or 0, respectively, if a 1-by-2 block consisted of one white pixel and one black pixel, the entire block was assigned a label of 0.5). These blocks overlapped to produce a total of 90 blocks of size 1-by-2 pixels. A third representation of the stimuli images was formed by 2-by-1 pixel blocks, and finally a fourth representation was formed by 2-by-2 pixel blocks, where each block is labeled by the average intensity of its component pixels. These four representations were used to train four types of local decoders, one type for each representation.

2. Local Decoders: Each block in each of the four representations of the stimuli images had an associated local decoder. That is, the 1-by-1 pixel block representation had 100 local decoders, the 1-by-2 representation had 90 local decoders, the 2-by-1 representation had 90 local decoders, and the 2-by-2 representation had 81 local decoders, yielding a total of 361 local decoders that were independently trained. Each local decoder was trained to predict the intensity value of its associated block in a stimulus image using all voxels in the associated fMRI scan. For the 1-by-1 representation, each block (each pixel) was assigned a label of either 0 or 1, so the problem reduced to binary classification, but for the 1-by-2 and 2-by-1 representations each block could take a value of 0, 0.5, or 1, and for the 2-by-2 representation values of 0, 0.25, 0.5, 0.75, and 1 were possible, so for these latter three representations the problem was multiclass classification. For this reason Miyawaki, et. al., used multinomial logistic regression to assign weights to voxels for each local decoder. Rather than using maximum likelihood to find these weights, they used sparse logistic regression to force only a small number of voxels to have nonzero weights. This was done to maintain biological realism, since experiments have shown that localized visual stimuli (e.g., an image with only one white pixel) result in sparse neural activity as seen in fMRI scans.

3. Combination Coefficients: Because the representations for the 1-by-2, 2-by-1, and 2-by-2 pixel blocks overlap, the outputs of the local decoders must be multiplied by coefficients and then summed to produce the value of each pixel in the reconstructed stimulus image. For instance, the first pixel (top left of the image) has four local decoders associated with it, one for each representation. Each of these four local decoders outputs a single number that represents the prediction for the intensity value of the first pixel. Each of these four numbers is multiplied by a different coefficient and summed to produce the final prediction for the value of the first pixel. Least-squares was used across all training observations of the first pixel to calculate the optimal coefficients. The pixel in the second row and second column of the image has 9 associated local decoders (due to overlap, this pixel lies within 9

blocks across the four representations), and similarly least-squares is performed to calculate the optimal values for these 9 coefficients. Once the optimal coefficients are calculated for each pixel in the stimulus image, the algorithm is complete.

Similar work has since focused on reconstructing motion from fMRI scans. Nishimoto, et. al. utilized a motion-energy model that predicted a change in visual stimuli (i.e. a frame-by-frame difference) using fMRI voxels recorded from a subject viewing short video clips. Using this time-sensitive local decoder, Nishimoto, et. al. achieved approximately 0.3 reconstruction accuracy for moving frames.

2.2 Novelties in Our Work

After replicating the results from Miyawaki, et. al., we attempt to achieve similar or higher reconstruction accuracies using an array of machine learning techniques. We simplify the model used, and frame the fMRI to predicted stimuli problem as a basic machine learning algorithm where output and input pixels are independent. The novel techniques we apply include a linear classifier, SVM, and Neural Network. These techniques require significantly less computational time (on the order of minutes to hours) compared to the time required for techniques used in Miyawaki, et. al. (on the order of days) on a single machine. This significant decrease in computational time makes real-time fMRI to stimulus image reconstruction more feasible.

3 Technical Details

3.1 Summary

First we perform principal components analysis (PCA) to reduce the dimension of the input space from 6046 to 50. Using the training set, we trained a neural network for each pixel that takes the 50-dimensional voxel data as input and outputs either 0 or 1 to denote the intensity of the pixel. We train a separate neural network for each pixel, yielding a total of 100 neural networks. As described in section 2.1, Miyawaki, et. al., represented the visual stimulus images using 1-by-1 (original), 1-by-2, 2-by-1, and 2-by-2 pixel blocks, yielding a total of 361 blocks for each image. For each pixel block in the 1-by-2 case, we then train a neural network that takes the 50-dimensional voxel data as input and outputs a predicted value for that pixel block, and we repeat this for the 2-by-1 and 2-by-2 cases. This yields 361 neural networks. When reconstructing the image for a test example, we perform PCA on the test voxel data to obtain a 50-dimensional input. For each pixel, we feed this input into all neural networks that are associated with a pixel block that contains our pixel of interest. We then multiply each of the outputs of the neural networks by a coefficient (which was calculated using least-squares over the training set), and then sum these terms to produce our final predicted output for our pixel of interest. Performing this for all pixels yields a complete reconstructed image for our particular test input example.

3.2 Detailed Description

Dataset

The data is obtained from the website brainliner.jp, where ATR has posted publicly available fMRI data. A test subject was presented a series of 10-by-10 pixel images, where each pixel is either black or white. As the subject viewed these simple visual stimuli, fMRI scans of his visual cortex were conducted (6046 voxels). For the training of the reconstruction algorithm, during each of 20 runs the subject was shown 22 randomly generated images, and 3 scans were taken for each image. The resultant algorithm was used to decode in real time what the subject saw as he viewed 10 images of geometric shapes or letters per run for 12 runs. 6 scans were taken for each of these 120 images (the fMRI data was block-averaged in the original paper by Miyawaki, et. al.).

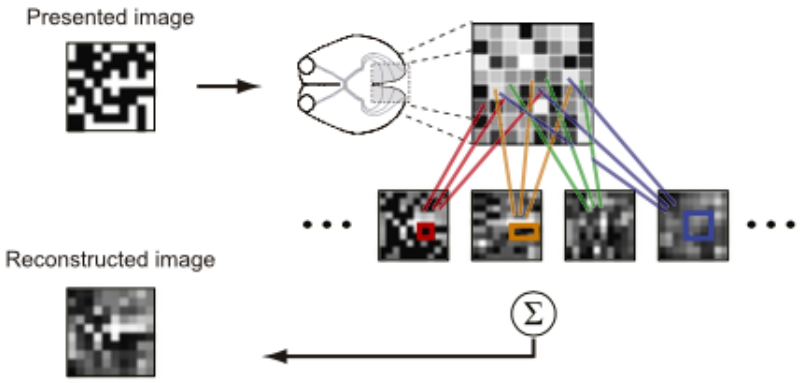

Figure 1: Experimental schematic from Miyawaki, et. al. 10-by-10 pixel images are presented to a participant, from which fMRI readings are obtained. fMRI readings are segmented into patches which are used to reconstruct the original presented image.

Techniques Explored

1. Linear Classifier: As a baseline technique we trained a linear classifier with fMRI scans and presented images represented as column vectors. The input and output variables are the column vector of an fMRI image and the column vector of the presented image, respectively. We implemented a simple least-norm solution in closed form.

2. Support Vector Machine: We replaced the least-norm solution with an SVM classifier for each pixel (a total of 100 SVMs). Each SVM uses the fMRI data for the associated image to predict whether a pixel will have a value of 0 or 1. We trained SVMs with a variety of kernels and regularization parameters to see which provides the best performance.

3. Neural Networks: We trained several neural network architectures with the details explained below.

Note: We used libraries to implement techniques such as SVM and neural network training, since pre-existing libraries like those of MATLAB are optimized to compute these algorithms very quickly (and our dataset is quite large). Data preprocessing, implementation of simple algorithms like least-norm, training and cross-validation, the specific architecture of our neural network, and data postprocessing was our own novel code, while we relegated the actual calculation of the SVM decision boundary (for instance) to libraries.

Evaluation Metrics

Our techniques were assessed using the mean-squared error (MSE) between predicted images and images that were actually shown. We define MSE as:

$$MSE = \frac{1}{|I|} \sum_{i \in I} ||\hat{i} - i||$$

Where I is the set of 10-by-10 stimulus images in our test set and $\hat{i}$ is our estimate of image i. We also considered using other evaluation metrics as well, such as the structural similarity index (SSIM).

Our training set was the same 20 runs used by Miyawaki, et. al., for their training set, and likewise our test set was their test set of 12 runs of images of geometric shapes and letters. By emulating the training conditions of the original paper, we were able to directly compare our results to those reported by Miyawaki, et. al.

4 Experiments

1. Replication of Previous Results

   (a) Setup: We trained local decoders as described in Miyawaki, et. al. for patch sizes of 1×1, 1×2, 2×1, and 2×2. Each local decoder maps concatenated pixel patches in the fMRI space to a single pixel in the stimulus image, and is computed with logistic regression. To predict a pixel in the stimulus image, a weighted combination

   (b) Results: See figure 2 for our replication of results described in Miyawaki, et. al. When using only 1×1 decoders, we received an MSE of 0.1200. When using a weighted combination of 1×1 and 1×2 decoders, we received an MSE of 0.1003.

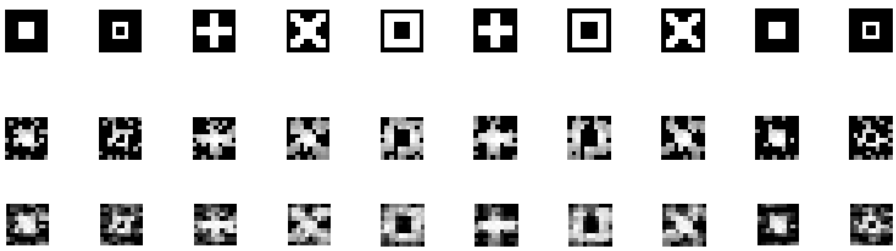

Figure 2: Visualization of original stimulus image (top row) vs. reconstructed image using only 1×1 decoders (middle row) and using both 1×1 and 1×2 decoders (bottom row).

2. Results of Linear Classifier

   (a) Setup: Let Y be our 1320-by-100 matrix containing the training set of images, and let A be our 1320-by-6046 matrix containing the associated fMRI training set data. Assuming that $Y = AX$, we are trying to find a 6046-by-100 matrix X that represents the least-norm linear transformation from A to Y. If we let B be our 720-by-6046 matrix containing the fMRI data for the test set, then $Z = BX$ will give us our estimates for our test images. However, we will force every element in Z that is greater than 0.5 to be 1, and all other elements will be 0. We can then directly compare our estimated test images to the actual test images, casting the algorithm as a classification problem where each pixel can fall into one of two classes: black (0) or white (1). Note that because each pixel can only be either 0 or 1, the MSE between an estimated image and an actual image is actually identical to the classification error averaged over all pixels, which is why we report the MSE and not the RMSE below. The reported MSE was averaged over all images (i.e., the classification error averaged over all pixels and averaged over all images).

   (b) Results:

Method	Training MSE	Test MSE
Least-norm	0	0.4928

   (c) Comments: Although the least-norm solution is naive, it provides a baseline for further experiments. The average classification error is just under 0.5, so it performs about as well as random chance (0.5). Clearly the least-norm solution fails to explain the transformation between A and Y well.

3. Results of SVMs

   (a) Details: We trained an SVM for each pixel (a total of 100 SVMs) to predict whether that pixel will have a 0 or a 1 as its value. For the linear SVM, we used a range of regularization parameters

λ ranging between 0.01 and 100 and selected the value that minimized error. We used a variety of kernels: linear, quadratic, third-order polynomial, radial basis function, and a multilayer perceptron kernel with [-1 1] scale.

(b) Results:

Method	Training MSE	Test MSE
Linear SVM	0	0.4305
Quadratic SVM	0.1906	0.4931
Polynomial SVM	0.4927	0.5273
RBF SVM	0	0.5120
MLP SVM	0.5002	0.2750

(c) Comments: The MLP kernel by far offered the best performance on the test set, although it performed much less well on the training set, which is concerning. The second best test set performance was seen by the linear kernel, although it seemed to overfit the training data due to its high test error compared to training error. However, if the images produced by the MLP SVM are viewed, it is seen that the predicted intensity are almost always zero for each pixel. A trivial algorithm that always predicts zero for every pixel would achieve an MSE similar to that reported above for the MLP SVM test MSE. Regularization was performed as well, but the MSE values did not change significantly from those reported above. Thus, at least from what we can gather from our experiments, the SVM produces a trivial solution. However, our experiments below show that neural networks attempt to model the complex nonlinear relationship between fMRI data and stimulus images with greater success.

4. Results of Neural Networks

(a) Probability-outputting Neural Networks: Our initial approach with the neural networks was to predict one pixel at a time using the fMRI data. We implemented a neural network with various architectures, all ultimately predicting the value of one pixel based off 50 or 100 largest principal components of the training data. The general neural net architecture is summarized in figure 3. Some of the architectures we used were [10], [10 10], [15 15], where each number represents the number of hidden nodes in each hidden layer. We predicted each pixel using 20 neural nets and averaged their outputs; this made the process much slower but improved the consistency of our results. After the nets were trained and the test data was evaluated to yield a score for predicting 0 (ϕ_0) and a score for predicting 1 (ϕ_1), the actual pixel value p was calculated using the following formula, which represents the probability of the pixel having an intensity value of 1:

$$p = \frac{\phi_1}{\phi_0 + \phi_1}$$

Some of our best performing and most consistent results are shown in figure 4.

(b) Optimally Combined Four-Basis Neural Networks: We then represented the stimulus images using four different sizes of pixel blocks (1-by-1, 1-by-2, 2-by-1, and 2-by-2 pixels), just as was done in the Miyawaki, et. al., paper. This yielded a total of 361 features for each stimulus image, up from the original 100 pixels for each stimulus image. We trained 361 neural networks to predict each one of these features. Rather than predicting probabilities as in the previous technique, in this technique each neural networks predicted the intensity of the pixel from a set of class values (0 or 1 for the 1-by-1 pixel block case, 0, 0.5 or 1 for the 1-by-2 and 2-by-1 case, and 0, 0.25, 0.5, 0.75, and 1 for the 2-by-2 case). We then used least squares on the neural networks associated with each pixel to find the optimal combination of the outputs of those neural networks. For instance, for the top left pixel (the first pixel), the associated blocks consist of one 1-by-1 block, one 1-by-2 block, one 2-by-1 block, and one 2-by-2 block. For a pixel in the middle of the image, there are 9 associated blocks (because blocks overlap). These class predictions were then optimally combined with coefficients that were determined using least-squares on the training data. To reconstruct images from test set fMRI data, the data underwent PCA to reduce the dimensionality to 50 and then fed into each of the 361 neural networks. For each pixel, the associated neural network

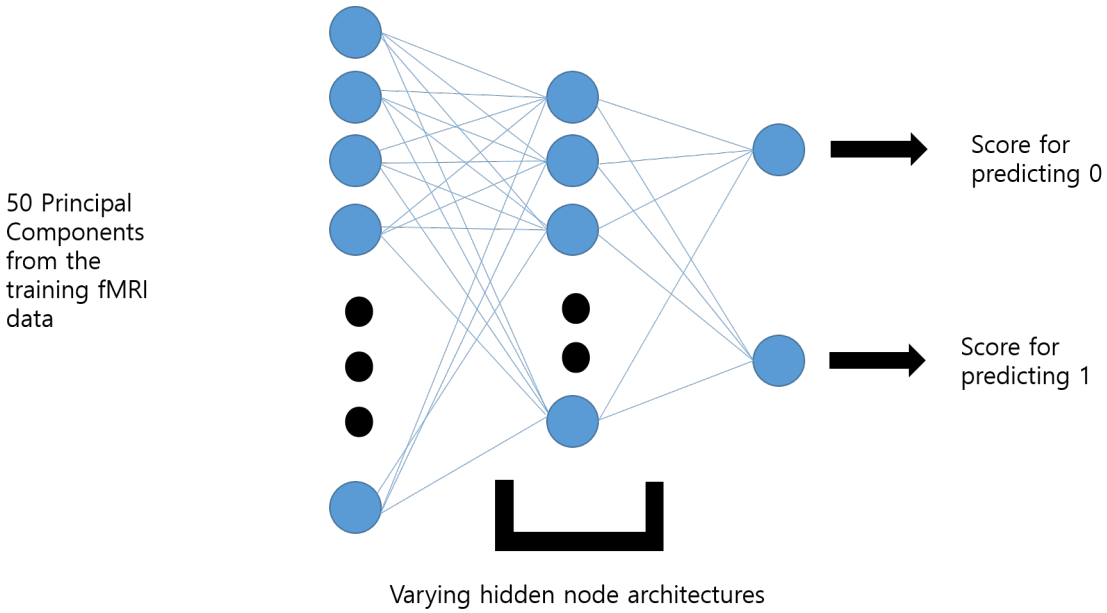

Figure 3: Architecture of a single probability-outputting neural network.

outputs were combined using the previously calculated optimal coefficients to produce a single intensity value for each pixel. A few representative images are shown in figure 5.

(c) Fitting Networks: Our last technique involved using function fitting neural networks to learn a relationship between the 50-dimensional PCA-processed fMRI data (the input) and the 100-dimensional visual images (the output). Instead of training a separate network for each pixel, our networks outputted 100-dimensional vectors (thereby allowing the network to consider relationships among all pixels). We trained four neural networks, one for each size of pixel block. We then used least-squares to find the optimal combination coefficients, which we used to reconstruct images using the test set input data. Each network had one hidden layer, and we experimented with 5, 10, 15, and 25 hidden neurons in the hidden layer. Sample reconstructed images are shown in figure 6.

Results:

Number of Hidden Neurons	Training MSE	Test MSE
5	0.2406	0.2753
10	0.2386	0.2841
15	0.2375	0.2949
20	0.2353	0.2851
25	0.2330	0.3040

5 Conclusions

Although our neural networks greatly outperformed our SVMs, both techniques shed light on the nature of the data, especially when the logistic regression model of the original Miyawaki, et. al., paper is considered. We summarize our key conclusions from our research as follows:

1. A weighted linear model does not effectively describe the relationship between the fMRI images and stimulus images. The results from various linear models, including the linear classifier and support

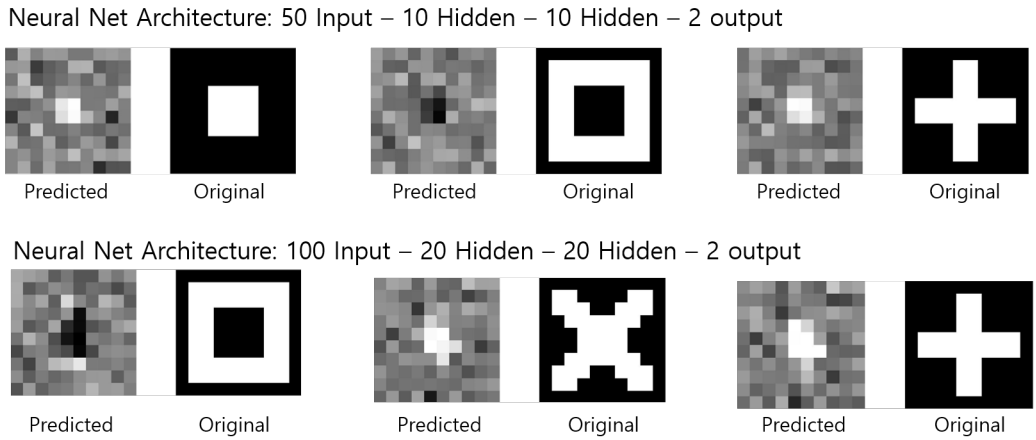

Figure 4: Original and predicted images produced by the probability-outputting neural networks during the testing session.

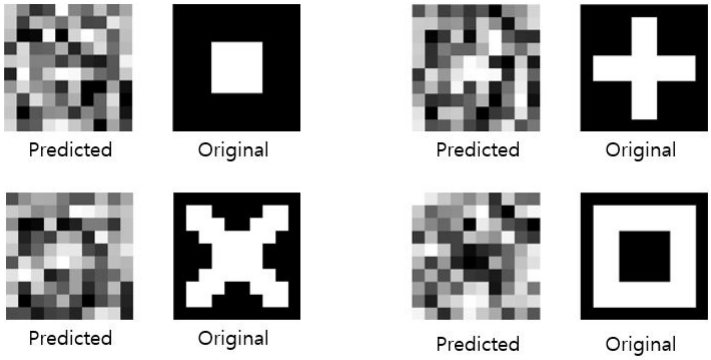

Figure 5: Original and predicted images produced by the optimally combined four-basis neural networks during the testing session.

vector machine with linear kernel yielded high MSE (between 0.43 and 0.50) that are marginally better than guessing the output pixel. However, these results served as baselines to which comparison with other methods that utilize spatial information can be made.

2. The use of PCA significantly improved results by reducing noise in the input fMRI data. Although PCA was necessary because training neural networks on the full 6064-dimensional input data was prohibitively slow, the first 50 principal components captured 93% of the variance in the data, which justifies the use of PCA. Shorter neural network training times allowed us to optimize the networks over various hyperparameters.

3. A large number of neurons in the hidden layer of each neural network seemed to be optimal. This indicates that the input-output relationship of the data is highly nonlinear, which is apparent from the poor performance of linear methods and of the SVMs. In the original paper multiclass sparse logistic regression is used to create decision boundaries in the input space, but further optimization of our neural networks could lead to reconstruction errors even smaller than those produced by Miyawaki, et. al. Although it is difficult to optimize neural networks in a systematic way, their high degree of flexibility makes them particular suited to this research problem.

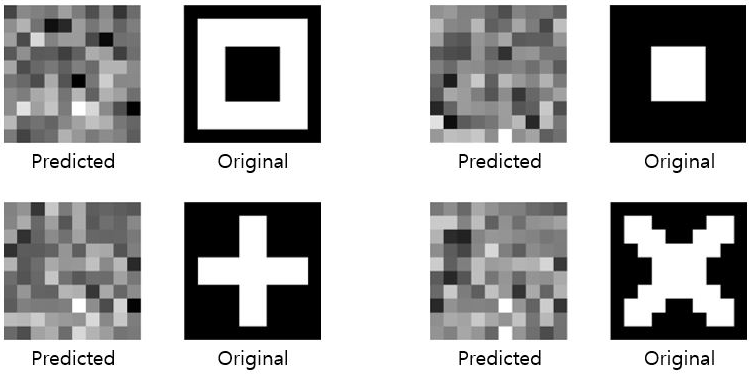

Figure 6: Original and predicted images produced by the function fitting neural networks during the testing session.

4. Miyawaki, et. al., noted that the use of four different pixel block sizes (1-by-1, 1-by-2, 2-by-1, and 2-by-2 pixel blocks), along with their least-squares optimal combination, substantially improved reconstruction performance. The use of blocks larger than 1-by-1 allowed their algorithm to consider correlations between neighboring pixels.

5. Future Work: The techniques used by Miyawaki, et. al., as well as our own techniques viewed the data as vectors rather than as matrices. That is, each three-dimensional fMRI scan was concatenated into a 6046-element vector, and similarly the stimulus images were concatenated into a 100-element vector. Although positional image is still maintain in this format, information specific to image data is lost. For instance, a more sophisticated algorithm could use the histogram of gradients (HoG) technique to extract useful information from the fMRI voxel data, which would then be used in a neural network or in logistic regression. Furthermore, we could have simplified the fMRI data using Eigenfaces (SVD-based) or Fisherfaces (LDA-based) to reduce the dimensionality of the input space in a visually meaningful manner. Another possible direction would be to use minimum mean square error (MMSE) estimation to compute the optimal combination coefficients for the four sizes of pixel blocks. Whereas least-squares makes no assumptions about the underlying distribution of the data, MMSE makes use of the (sample) variance of the data to provide an estimate more precise than that provided by least-squares. This approach would be justified if the fMRI data is approximately normally distributed. If some voxels are noisier than others, MMSE would alter the least-squares estimate to account for this. Finally, for the optimally combined four-basis neural network, we could have calculated the probability of the intensity being 1 for each pixel block (each feature) from each of the 361 neural networks, instead of simply assigning a class label to each pixel block.

References

1. Miyawaki, et. al. "Visual Image Reconstruction from Human Brain Activity Using a Combination of Multiscale Local Image Decoders." Neuron, Volume 60 Issue 5, pp. 915-929.

2. Kamitani, Yukiyasu. "Decoding Early Visual Representations from fMRI Ensemble Responses." In *Visual Population Codes: Toward a Common Multivariate Framework for Cell Recording and Functional Imaging.*, edited by Nikolaus Kriegeskorte and Gabriel Kreiman. 2011.

3. Nishimoto, et. al. "Reconstructing Visual Experiences from Brain Activity Evoked by Natural Moves." Current Biology, Volume 21, pp. 1641-1646.