fMRI-Based Image Reconstruction Using the Elastic Net

Bachelor thesis:

Jeffrey Lemein (0710326)

Supervisor: Marcel van Gerven

July 16, 2010

Contents

Chapter 1

Introduction

One of the interesting studies in neuroscience is the ability to reconstruct images based on recorded brain activity. Though we are still in the early phase, this technique can offer us a lot of practical and useful applications. Some applications you can think of are reading out someones thoughts or even dreams. Another more useful application of this technique is found in the field of crime fighting. Imagine a victim that is able to memorize the face of a perpatrator, but is not able to describe him (or her) in an accurate way. Reconstructing an image based on the victims brain activity can offer the police some useful clues for finding the criminal. Unfortunately, this application is still far from being brought into reality.

In general, images can be reconstructed (or identified) by using a systematic mapping between visual stimuli and brain activity. In this process, a difference can be made to the type of the reconstruction process: the reconstructions can be reconstructed from visual stimuli to the brain, or from the brain to the visual stimuli. In the first case decoding is achieved by evaluating the mapping. In the second case decoding is achieved via an inversion procedure [9].

Previous fMRI studies have shown that visual features, such as orientation and motion direction [7, 8], and visual object categories [3, 5] can be decoded from fMRI activity patterns by a statistical decoder, which learns the mapping between a brain activity pattern and a stimulus category from a training data set [13]. Details about what fMRI exactly is, will be explained later.

A recent study from Kay et al. [10] has demonstrated that it is possible to identify a presented image among a large number of candidate images using a ’receptive field model’. A receptive field is that region of visual space to which individual neurons or voxels in the brain will respond. This ’receptive field model’ predicts the fMRI activity for the visual image that a person has seen. By comparing this predicted fMRI activity with the real fMRI activities as measured during training sessions, it is possible to accurately find the correct image that the person has seen. But image identification is constrained by the candidate image set. It would be much more interesting if it would be possible to reconstruct images without restricting oneself to a predetermined set.

A study by Thirion et al. [17] showed that it is possible to reconstruct the actual image that was seen, but the reconstructions were not very accurate and the resolution was low. A study by Miyawaki et al. is more promising [13]. They used several decoding techniques to produce high-quality image reconstructions.

They started their experiment by constructing 10 by 10 pixel contrast images. These contrast images were gray (zero contrast) or filled with a flickering checkerboard pattern (full contrast) [9]. The subjects were presented with numerous images (rectangles, crosses and other shapes). While watching the images, fMRI activity was collected from the visual areas V1, V2, V3 (early visual areas) and V4 (higher visual area). They then developed a reconstruction model and trained it on their data. The first stage consisted of predicting local contrast based on linear combinations of voxel responses. A voxel is a contraction of volume and pixel and corresponds with a volume cell in three dimensional space. So, in this case a voxel is a cell in the visual area of the brain. The results were reasonably good and the reason for this is that voxels in early visual areas reliably signal the amount of contrast in their spatial receptive field [10]. The second stage consisted of combining the predicted local contrast into a single image that estimates the pattern of contrast the subject saw [9]. They then tested their reconstruction model using data that was separated from the training data.

This thesis uses the data collected by Miyawaki et al [13]. This data will be used for the same task as they used it for. The difference is that in this thesis elastic net will be used as algorithm to predict the contrast. Also Miyawaki et al. predicted the contrast per local region and combined these regions into a single image. In this experiment the reconstruction image will be generated pixel by pixel. The reconstructions from Miyawaki's reconstructor and the elastic net reconstructor tested in this study can not be compared. This is because Miyawaki et al. used flickering checkerboard patterns and homogeneous gray areas in their training images. This study uses structured images both for training and testing.

The central question in this study is: what is the contribution of the visual cortex and the point in time to the reconstructions made using elastic net? In other words: the reconstructions, produced by elastic net, from several visual areas and time samples are compared with one another.

Chapter 2

Regression

2.1 Introduction to regression

Regression is a technique which tries to estimate the value of a target variable based on a set of known values corresponding to one or more predictor variables. Examples of regression include predicting the fuel usage based on someone's driving style, the unemployment rate given economic factors, the amount of sold ice creams based on weather conditions and estimating the age of a fossil according to the amount of carbon-14 left in the original material [16].

Observations and their corresponding target values can be denoted in a dataset D,

$$D = \{(\mathbf{x}_i, y_i) \mid i = 1, 2, \ldots, N\}. \tag{2.1}$$

Here N denotes the number of observations, x_i corresponds to the set of attributes of the ith observation (also called the explanatory variables) and y_i corresponds to the target (or response) variable of the ith observation. The goal of regression is to find a target function f that best matches the target variable given the input data. To establish an understanding of a best match there has to be a way of telling how good (or bad) a target function maps to the target variable given the set of attributes. This is done by introducing an error function. A commonly used error function is the sum of squared error.

$$\text{Sum Squared Error (SSE)} = \sum_i (y_i - f(\mathbf{x}_i))^2 \tag{2.2}$$

The best matching target function is the one that minimizes the error function. In this thesis the sum of squared error function is used to compute the error between the predicted and original data. Also is it used as a measure for image reconstruction (see section 3.5).

2.2 Linear regression

Linear regression is a specific form of regression in which the task is to learn a linear target function f that best matches the target variable given the set of predictor variables. In this chapter I will explain how linear regression problems can be solved by first explaining it for a single predictor variable and then for the general case with more predictor variables, also called multivariate linear regression.

2.2.1 Linear regression with a single predictor variable

Suppose there is just one predictor variable x. The linear target function f is then a function with two parameters β_0 and β_1, also called regression coefficients.

$$f(x) = \beta_0 + \beta_1 x \tag{2.3}$$

The β_0 is the offset and β_1 is the rate of change. Given a set of values for the predictor variable x and their corresponding values for the target variable y, the task of linear regression is to find a value for β_0 and β_1 so that f minimizes the the sum of squared error.

$$SSE = \sum_{i=1}^{N}[y_i - f(x_i)]^2 = \sum_{i=1}^{N}[y_i - \beta_1 x - \beta_0]^2 \tag{2.4}$$

To find the corresponding values of β_0 and β_1, take the the partial derivatives of SSE, set them to zero and solve the resulting set of linear equations.

$$\begin{aligned}
\frac{\partial SSE}{\partial \beta_0} &= -2\sum_{i=1}^{N}[y_i - \beta_0 - \beta_1 x_i] = 0 \\
&\Rightarrow \beta_0 \cdot N + \beta_1 \sum_{i=1}^{N} x_i = \sum_{i=1}^{N} y_i \\
\frac{\partial SSE}{\partial \beta_1} &= -2\sum_{i=1}^{N}[y_i - \beta_0 - \beta_1 x_i]x_i = 0 \\
&\Rightarrow \beta_0 x_i + \beta_1 \sum_{i=1}^{N} x_i^2 = \sum_{i=1}^{N} x_i y_i
\end{aligned}$$

These equations can be summarized in a matrix equation, known as the normal equation:

$$\begin{pmatrix} N & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix} \tag{2.5}$$

The regression coefficients can now easily be computed by multiplying both sides of the equation with the inverse matrix:

$$\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} N & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix} \tag{2.6}$$

In the next section an example will be presented to show that solving linear equations with one predictor variable is very straight forward.

An example

Based on Tan's example [16], suppose we want to predict the skin temperature of a person during sleep based on the heat flux measurements generated by a heat sensor. By measuring these variables in examples in the real world, we may end up with the data set shown in Table 2.1 and Figure 2.1. Here the skin temperature value is shown corresponding with the amount of heat flux.

The task is now to find a linear target function that best matches the input data given a specific error function. In this case the sum of squared error is chosen to be the error function.

Heat flux	Skin Temperature		Heat flux	Skin Temperature
3.1290	32.3611		3.2782	32.7444
3.6665	32.1991		3.6772	32.1355
3.9895	32.4804		4.0984	31.9924
4.3886	32.3370		4.5158	32.2604
4.7156	32.1000		5.3427	31.9219
5.6559	32.3326		5.7720	31.8205
5.9347	31.8215		6.0629	31.6435
6.2397	31.6295		6.2914	31.6640
6.9193	31.9840		7.0379	31.5379
7.0889	31.4667		7.1230	31.5075
7.1595	31.7320		7.1979	31.3175
7.3562	31.6711		7.6015	31.2252
7.8497	31.4105		7.8515	31.0709
8.1032	31.4879		8.5590	31.4186
8.8710	31.0737		8.9158	31.2648
9.0835	31.1968		9.4115	31.3237
9.4362	31.1121		9.5353	31.2724
9.5680	31.0339		9.7180	30.8945
9.8428	30.9890		10.1676	30.8850
10.3430	31.0625		10.9135	30.7490

Table 2.1: Measurements from examples in the real world

By representing the data as a matrix and using Equation 2.6, the regression coefficients can be computed very easily. In the following equation the resulting regression coefficients are calculated:

$$\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} 40 & 282.4117 \\ 282.4117 & 2183.6 \end{pmatrix}^{-1} \begin{pmatrix} 1264.1 \\ 8884.6 \end{pmatrix} = \begin{pmatrix} 33.1031 \\ -0.2125 \end{pmatrix} \tag{2.7}$$

Now that we have the regression coefficients, the best matching target function is $f(x) = 33.1031 - 0.2125x$. Here $f(x)$ is representing the skin temperature given a heat flux value x. Figure 2.2 shows the target function together with the data set.

2.2.2 Extending linear regression to multivariate linear regression

The previous example showed that solving a linear regression problem with one predictor variable is reasonably straight forward. In reality there are often more predictor variables and that is why we need a way to extend linear regression with a single predictor variable to the multivariate case. Assume for now that we still have one predictor variable. The normal equation shown before (in Equation 2.5) can be constructed easily with just two matrices. One for the predictor variables and one for the target variable.

Let $X = (\mathbf{1}, \mathbf{x})$, where $\mathbf{1} = (1,\ 1,\ \ldots)^T$ is a vector of ones and $\mathbf{x} = (\ x_1, x_2, \ldots, x_N\ )^T$ a vector with the values of the predictor variable. The T means that the corresponding matrix should be transposed. The left-hand side of the normal equation can now be constructed.

$$X^T X = \begin{pmatrix} N & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix} \tag{2.8}$$

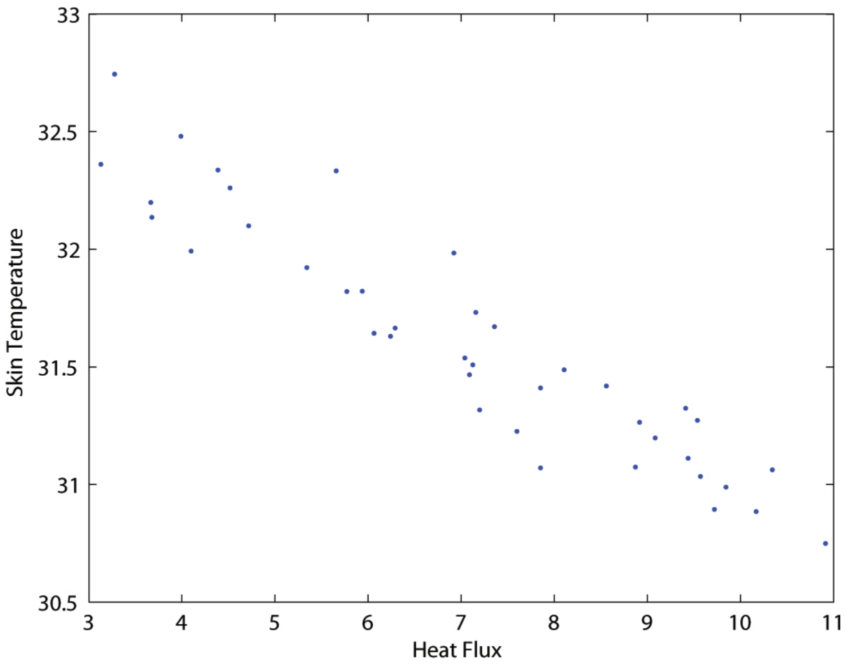

Figure 2.1: Measured skin temperatures given specific heat fluxes

Now let $\mathbf{y} = (\ y_1, y_2, \ldots, y_N\ )$ be the vector containing the values for the target variable. The right-hand side of the normal equation can now be constructed too.

$$X^T y = \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix} \tag{2.9}$$

Given the weight vector $\beta = (\beta_0, \beta_1)^T$ and Equation 2.5, the regression coefficients can be computed with the following formula.

$$\beta = (X^T X)^{-1} X^T y \tag{2.10}$$

The nice thing about this matrix multiplication is that it generalizes to multiple predictor variables, and thus, to multivariate linear regression. To deal with the multivariate case, the predictor matrix $\mathbf{X}$ needs to contain the other predictor variables as well. This is done by adjusting X so that each column (except the first one) contains the values for a different predictor variable. Suppose we have d predictor variables, in which a predictor variable is denoted by a vector $\mathbf{x}_d$. Each predictor variable $\mathbf{x}_d$ contains measurements corresponding with the d-th predictor variable. Then $\mathbf{X}$ is adjusted so that $\mathbf{X} = (\ \mathbf{1}, \mathbf{x}_1, \ldots, \mathbf{x}_d\ )$ contains all the predictor variables. As a consequence, the number of regression coefficients have increased too, so that $\beta = (\ \beta_0, \beta_1, \ldots, \beta_{d+1}\ )$ in which d is the number of predictor variables.

With this extension, it is now possible to compute the weight vector for any number of predictor variables. The only thing that needs to be done is performing the matrix computation as shown in Equation 2.10. With this knowledge it is possible to start reasoning about the simple pixel-based image reconstructor using fMRI data as predictor variables and the pixel color of the seen image as target variable.

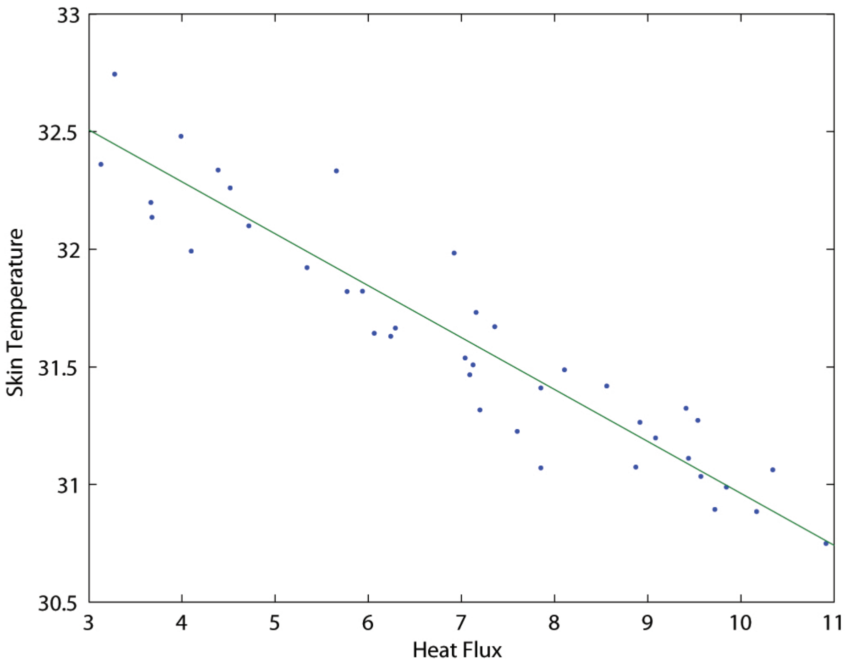

Figure 2.2: Best matching target function using SSE as error function

2.3 Ridge regression (L2)

2.3.1 The problem of collinearity

A major flaw in Equation 2.10 is the problem that occurs when two columns (or vectors) in matrix X are linearly dependent. This problem is called collinearity. When collinearity occurs within a matrix X the inverse, X^{-1} can not be computed. Think about this for a minute. When a matrix is multiplied by it's inverse, the result is the identity matrix I. But an identity matrix doesn't have columns that are linearly dependent on each other. In fact, it's the perfect example of a matrix that doesn't have linear relationships between columns. So, it's not possible to construct an inverse matrix of a matrix in which collinearity occurs. Because of this, $(X^T X)^{-1}$ can not be computed so that the weight vector result in invalid values.

2.3.2 Ridge regression as a solution against collinearity

Ridge regression tries to solve the problem of collinearity by introducing a (regression) matrix $\Gamma = \lambda I$ that is added to $X^T X$, where I denotes the identity matrix and λ the regularization parameter that is used to regulate the amount of offset. The goal is to get a matrix with columns that are less linearly dependent so that the inverse can be computed. As a result, ridge regression produces weight parameters with a small bias, whereas ordinary linear regression produces unbiased estimators. But according to [1], the variances of these new parameters (using ridge regression) are smaller than those of the ordinary linear regression. They might even outperform the unbiased parameters.

You could see the regression matrix as a (linearly independent) offset that is added to $X^T X$ to produce a matrix with linearly independent columns. As said before, the identity matrix is the perfect example of a matrix with independent columns. Therefore, this matrix is often chosen to be added to the regression matrix[1]. Let β_{L2} be the set of regression coefficients corresponding with the coefficients produced by ridge regression. These regression coefficients can then be computed as follows.

$$\beta_{L2} = (X^T X + \lambda \cdot I)^{-1} X^T y \tag{2.11}$$

The scalar variable λ, called the regularization parameter, is multiplied with the identity matrix to regulate the amount of offset being added: too small and it can not fight collinearity, too large and the bias of the parameters becomes too large to be able to compute reasonable weights. According to [1] the optimal value for λ can not be calculated and the best way to achieve this optimal value is trying it out and stick with the best result.

2.4 Lasso regression (L1)

Lasso regression is a regression method that stands for *least absolute shrinkage and selection operator* and enjoys some of the favourable properties of both subset selection and ridge regression [18]. The effect of lasso regression is that some regression coefficients will be pushed to zero and others will be shrinked. With this approach the bias increases a bit, but the variance will be smaller. And so it results in a more stable reconstruction model. Additionally it is possible to get an insight in the most important coefficients, and in the field of image reconstruction using fMRI data, also in the most important brain regions used to represent the seen image in the brain.

Lasso regression minimizes the sum of squared error with a bound on the sum of the absolute regression coefficients. Let β_{L1} be the set of regression coefficients that are computed with lasso regression. These lasso regression weights can then be characterized as follows.

$$\beta_{L1} = \{\beta_0, \beta_1, \ldots, \beta_d\} \text{ where } \sum_{i=0}^{d} |\beta_i| \leq b \tag{2.12}$$

Here b is corresponding with the bounds on the sum of the absolute regression coefficients (i.e. the sum of the absolute regression coefficients must not exceed bounds b). The algorithm to compute the lasso regression coefficients is explained clearly in the articles by Tibshirani and others [18, 4].

2.5 Elastic net algorithm

Ridge regression is known to shrink the coefficients of correlated predictors to each other while lasso regression selects a subset of predictors and assumes that a lot of predictors have a coefficient close to zero [4]. Elastic net is designed to combine these two measures as the *elastic net penalty* P. The entire family of P_α creates a useful compromise between ridge and lasso regression [4].

The elastic net solves the following problem:

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} \left[\frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \mathbf{x}_i^T \beta)^2 + \lambda P_\alpha(\beta) \right] \tag{2.13}$$

where β_0 and β are the regression coefficients and $P_\alpha(\beta)$ is:

[1] The assumption here is that all variables are treated as being independant of each other. And in the case of reconstructing images from fMRI data, we assume that the explanatory variables of the fMRI data are independent.

$$P_\alpha(\beta) = \sum_{j=1}^{p} \left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right] \text{ with } j = 1, \ldots, p \tag{2.14}$$

As has been said before, P_α is the elastic net penalty and α can be used to get a compromise between the ridge regression penalty ($\alpha = 0$) and the lasso regression penalty ($\alpha = 1$). If you choose $\alpha = 1 - \epsilon$ for some small $\epsilon > 0$, then the elastic net results in lasso regression but removes degeneracies caused by extreme correlations [4]. Note that the ridge and lasso regression parameter do not necessarily have to be expressed with α. It's perfectly legal to choose two different values that do not sum up to one. This is what is done in this thesis. More on this later in Chapter 4.

The idea behind computing the regression coefficients using elastic net is that the regression coefficients are set to an initial value of zero. Elastic net then continually tries to optimize the coefficients until the change of the coefficients is smaller than a predetermined toleration value. Choosing a small toleration value causes the algorithm to take longer to find the best values for the coefficients. A detailed explanation of this algorithm is explained in [4].

Chapter 3

Data used for this Research

3.1 What is fMRI and how is it measured?

fMRI stands for functional Magnetic Resonance Imaging and is a specialized type of MRI scan. It measures the change in blood flow related to neural activity in the brain. fMRI scans are performed with a Magnetic Resonance (MR) scanner. The technique to detect changes in blood flow using magnetic resonance imaging is called blood oxygenation level dependent (BOLD) imaging. Changes in blood flow and blood oxygenation in the brain are closely linked to neural activity [15]. Active neurons lead to a regional increase in oxygenated blood but this is not accompanied with a corresponding increase in oxygen utilization. This difference in oxygen supply and consumption underlies the BOLD signal (blood-oxygen-level dependence) [14, 11]. Since blood oxygenation varies according to the levels of neural activity, these differences can be used to detect brain activity [2]. This effect will occur approximately one to five seconds after the subject has seen the image and it will remain at it's peak for four to five seconds, before falling back to the baseline blood flow. This effect of increased blood flow leads to local changes in the relative concentration of oxyhemoglobin and deoxyhemoglobin and changes in local blood volume and local blood flow to the brain. This process is called hemodynamic response. In Figure 3.1 the hemodynamic response function is shown.

3.2 The visual cortex (V1, V2, V3 and V4)

The term visual cortex is referring to the primary visual cortex (also known as V1) and extrastriate visual cortex areas such as V2, V3 and V4. There is a visual cortex for each hemisphere of the brain where the left hemisphere visual cortex receives from the right visual field and the right hemisphere visual cortex from the left visual field. The visual cortex is sensitive to visual stimuli. In this paper, these stimuli are constrast-defined images.

Nearly all visual information enters the cortex via area V1 [12]. This area is located in the back of the brain. V1 contains cells that react to stimuli that are localized in space, orientation and frequency. Therefore, V1 is good in processing lines, rectangles and edges. When looking at this study, V1 is particularly good in reacting to contrast. So this is very useful for the contrast defined images.

Visual area V2 receives input from visual area V1. According to Hegdé et al. [6], V2 cells respond well to some complex stimuli. These stimuli consist of grating and contour stimuli. Approximately one-third of the V2 cells showed significant differential responsiveness to various complex shape characteristics and many were also selective for the orientation, size and spatial frequency of the shape [6]. These results indicate that V2 cells explicitly represent complex shape information [6].

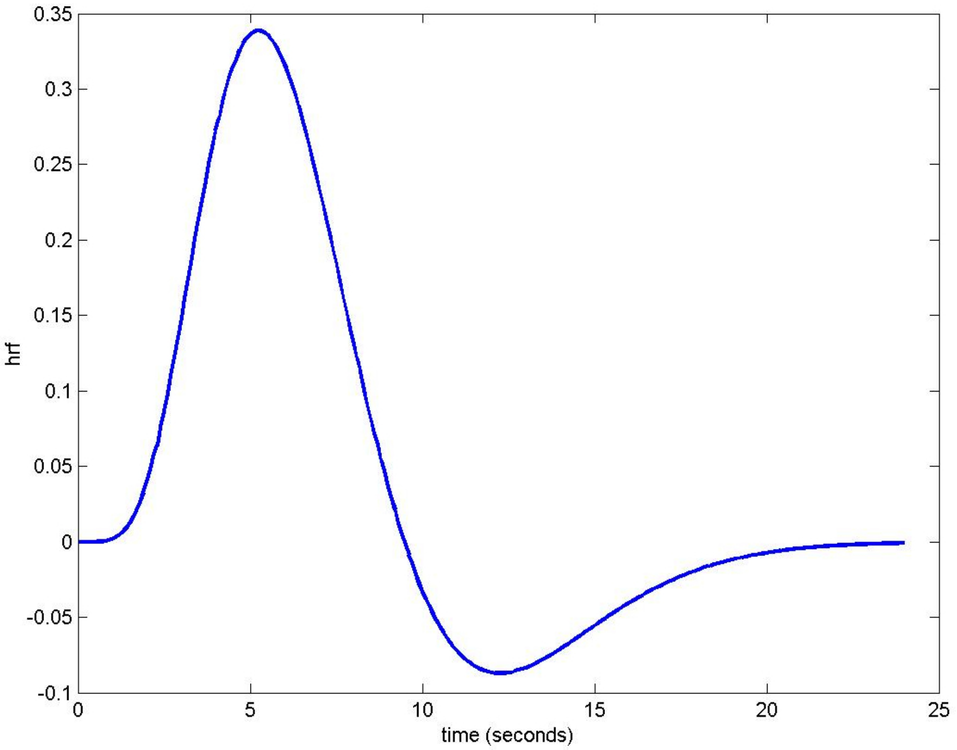

Figure 3.1: The hemodynamic response function. It shows the BOLD response of a certain region in the brains over time when a subject sees an image.

Visual area V3 receives part of its input from area V2 and from a layer in area V1 [12]. The function of visual area V3 is not really clear, Most cells in V3 are selective for orientation and many are also reactive to motion and to depth. Just a small set of cells are color sensitive.

Also visual area V4 is still reasonably unexplored. This area receives input from regions of area V2, but also from areas V1 and V3 [12]. V4 contains many cells that are color sensitive. This can indicate that it is used to distinguish colors in images. In this study, contrast images are used that contain no colors other than black and white. In visual area V4, there are also cells with complex spatial and orientation tuning, suggesting that the area is also important for spatial vision.

3.3 Data used in this research

In this research fMRI data is used from Miyawaki et al. [13]. This data is retrieved by rear-projecting visual stimuli onto a screen placed in the scanner bore using a gamma-corrected LCD projector. The subjects used to gather fMRI responses were male adults with normal or corrected-to-normal visual acuity. Four of these subjects were screened for head motion in preliminary scans and two of them that showed the least head motion underwent the full experimental procedure. The subjects gave written informed consent [13].

3.3.1 Collection of data

The fMRI responses of the visual cortex are measured using three types of experiments:

- The random image session (see Section 3.3.2),

- The figure image session (see Section 3.3.3),
- The conventional retinotopy mapping session.

In these sessions, subjects viewed the stimulus sequence while maintaining fixation. To prevent subjects from losing fixation, the color of the fixation spot changed from white to red 2 seconds before each stimulus block started. To ensure alertness, subjects were instructed to detect the color change of the fixation spot that occurred after a random interval of 3-5 seconds from the beginning of each stimulus block.

The conventional retinotopic mapping is used to determine to which visual area the voxels belong. The retinotopy mapping session followed the conventional procedure using a rotating wedge and an expanding ring of flickering checkerboard. The data were used to delineate the borders between visual cortical areas and to identify the retinotopy map on the flattened cortical surfaces. The retinotopic mapping was only used to relate the conventional retinotopy and the location of voxels selected by our method.

In this thesis, only the data from the figure image session is used in the experiments. For completeness, the other experiments performed by Miyawaki et al. [13] are described as well.

3.3.2 The random image session

In the random image session, each run contained 22 stimulus blocks. These were shown for 6 seconds followed by 6 seconds of rest. Extra rest is added before the first stimulus block (28 seconds) and at the end of each run (12 seconds). Each stimulus block is an image consisting of 12 by 12 small square patches. These images were presented on a gray background with a fixation spot to prevent subjects from moving their eyes away from the image. Each patch was either a flickering checkerboard or a homogeneous gray area, with equal probability. Each stimulus block had a different arrangement of random patches. To avoid the effect of the border frames of each stimuli, the central 10 x 10 area is used for analysis. This random image session consisted of 20 runs, so 440 different random patterns were presented to each subject.

3.3.3 The figure image session

In the figure image session, each run contained 10 stimulus blocks. Each stimulus block was showed for 12 seconds followed by 12 seconds of rest. Extra rest periods were added, as in the random image session. Stimulus images consisted of flickering checkerboard patches, as in the random image session, but formed geometric shapes (squares, small frames, large frames, plus and 'X') or alphabetic symbols ('n', 'e', 'u', 'r', 'o', 'n').

In each run five geometric shapes or five alphabets were presented, and each image was repeated twice. Subject S1 performed four geometric-shape runs and four alphabet runs, while S2 performed four geometric-shape runs and three alphabet runs.

3.4 Experiments

In this thesis a couple of experiments will be performed:

- Comparing reconstructions of contrast defined images looking at the difference in the used visual areas V1, V2, V3 and V4.
- Comparing reconstructions of contrast defined images looking at the difference in used time samples.

- Use elastic net to find the most important regression coefficients (and thus the most important regions in the brain).

3.5 Evaluation

Evaluation is the key point in almost any research and therefore this research is no exception. When images are reconstructed, it is important to be able to compare and evaluate reconstructions with each other. Here images are evaluated using the sum of squared error (SSE, see Equation 2.2) using cross validation. This measure is chosen because it is the same measure as the error function we used before in linear regression. And because we are trying to minimize the sum of squared error, it is logical to use the same measure when evaluating reconstructed images.

Chapter 4

Simple Pixel-Based Image Reconstructor

The simple pixel-based image reconstructor is a simple way of reconstructing images from fMRI data. It is simple because it consists solely of basic multivariate linear regression techniques. The idea behind this reconstructor is that images are reconstructed pixel by pixel. This is done by computing the regression coefficients for every pixel using the fMRI data as the predictor variables and the pixel color of all the samples as target variable. The algorithm used for reconstructing the images is elastic net. For the experiments a specific small constant value of $1e^{-3}$ is chosen for ridge regression parameter λ. This value is used for ridge regression and will stay the same during all experiments[1].

Because elastic net tries to find regression coefficients by computing β over and over again until the changes are smaller than a specific threshold, the reconstruction process is very slow. To address this issue, the threshold is chosen to be $1e^{-2}$, which is quite large. As a result, the reconstructions are not optimal, but they are still reasonably good.

4.1 Determining the appropriate ν value

The first step in this research was finding out the appropriate ν value to use with the image reconstructions. As is explained in Section 2.5, the elastic net algorithm has two input parameters. These are λ and ν. The first one is used in ridge regression to solve the problem of collinearity that can occur in the training data whereas ν is used for lasso regression. This value determines the number of predictor variables that will be used in the reconstruction process. In other words, the most important regression coefficients get a weight different than zero. This opens the possibility to pinpoint the areas in the brain that are used to see the images. In Table 4.1 an overview is presented with the number of predictor variables that are nonzero and the sum of squared error for the image, given their corresponding visual cortex. If the SSE is 1.0, this is equivalent with a single black pixel that is supposed to be white, and vice versa. The values are corresponding with twenty different images in the set that are trained using leave one out[2]. The values in the table are averaged over these images.

[1] The purpose for choosing a specific λ is to prevent the problem of collinearity. The only thing needed to prevent this is a small offset. So in this case a constant value of $1e^{-3}$ will suffice.

[2] These results were obtained using leave one out on twenty different images. Reconstructing the images using elastic net takes a considerable amount of time. Therefore, only twenty images are tested using leave one out and the resulting values are averaged over these images. Twenty images are tested because the set of images consist of twenty unique images.

ν-value		v1	v2	v3	v4
$1e^{0}$	# of nonzeros:	1 (0%)	1 (0%)	1 (0%)	1 (0%)
	sse per image:	11.22	11.22	11.22	11.22
$1e^{-1}$	# of nonzeros:	7 (1%)	9 (1%)	5 (0%)	6 (1%)
	sse per imagel:	7.46	7.74	8.26	8.30
$1e^{-2}$	# of nonzeros:	46 (4%)	50 (5%)	47 (4%)	56 (10%)
	sse per image:	3.62	3.67	4.18	5.10
$1e^{-3}$	# of nonzeros:	208 (20%)	196 (19%)	210 (17%)	187 (33%)
	sse per image:	1.72	1.87	2.05	3.44
$1e^{-4}$	# of nonzeros:	544 (53%)	543 (52%)	598 (48%)	422 (76%)
	sse per image:	1.23	1.27	1.50	2.99
$1e^{-5}$	# of nonzeros:	982 (96%)	1002 (96%)	1163 (94%)	554 (99%)
	sse per image:	1.29	1.44	1.45	3.06
$1e^{-6}$	# of nonzeros:	1018 (100%)	1046 (100%)	1237 (100%)	558 (100%)
	sse per image:	1.34	1.55	1.47	3.09
	# of predictors:	1018	1046	1238	558

Table 4.1: The SSE and number of predictor variables set to nonzero based upon the value of ν. λ is chosen the same ($1e^{-3}$). All brain volumes were used. The SSE values are corresponding with the averaged sum of SSE values for each pixel in 20 different unique images.

As can be seen from the analysis of the regression coefficients, the value of ν can be set to $1e^{-3}$ for the best results. These best results are dependent on how many of the predictor variables are used. The best thing is to have just a few coefficients set to nonzero, because this gives insight in the important coefficients and thus the important regions in the brain. A ν-value greater than $1e^{-3}$ causes too few predictor variables to be used in the reconstruction process. Choosing a ν-value smaller than $1e^{-3}$ causes too many predictor variables to be used. This reduces the effect of the lasso regression because lasso regression is used to find sparse solutions. So the best value for ν is $1e^{-3}$. A value of $1e^{-4}$ is also reasonable but not really favorable because then approximately half of the coefficients are used in the reconstruction process. Due to this amount of coefficients, it is difficult to find the most important coefficients. It would be very unlikely that half of the coefficients are important because it then includes noise and overestimates the importance of some coefficients.

4.2 Comparing reconstructions between the visual cortices and points in time

One important question to ask is which visual area contributes the most to the reconstructed image and in what amount. It is known that visual area V1 contains data for space, left/right orientation, et cetera. Visual areas V2, V3 and V4 are less understood and it would be great to see if these areas contribute to the reconstructions in a positive (or negative) way. To gain an insight into these visual areas, the SSE values of the reconstructions are calculated and shown in Table 4.2. In Figure 4.1 some sample reconstructions are shown.

time samples/visual cortex	V1	V2	V3	V4
t1 (after 2 sec)	3.55	4.13	7.59	9.91
t2 (after 4 sec)	3.68	4.61	6.33	10.49
t3 (after 6 sec)	4.00	4.55	6.88	10.73
t4 (after 8 sec)	6.26	5.83	9.42	11.84
t5 (after 10 sec)	11.16	11.28	10.25	14.14
t6 (after 12 sec)	11.62	11.42	10.57	13.58

Table 4.2: SSE values for the visual areas V1, V2, V3 and V4 for each specific sample in time. The SSE values are generated by averaging the SSE values over 20 different images and correspond to the SSE value for the whole image. Here a ν-value of $1e^{-3}$ is used, like has been found in Section 4.1.

4.3 Finding the important regression coefficients

Another interesting thing to see is what coefficients contribute the most to the reconstructed image. To gain an insight, six points are chosen out of the 100 pixels of which the images consist. These points are called A, B, C, D, E and F and are shown in Figure 4.2. The image shown in this figure is the average image of all the images which are used during the training session. The average image is useful to see what value the pixels have on average. And by looking at these pixels, we could expect that the pixels denoted by points 'A and F', 'B and E' and 'C and D' would have similar values for the regression coefficients. In Figure 4.3 the coefficients for each of these points are shown in a plot.

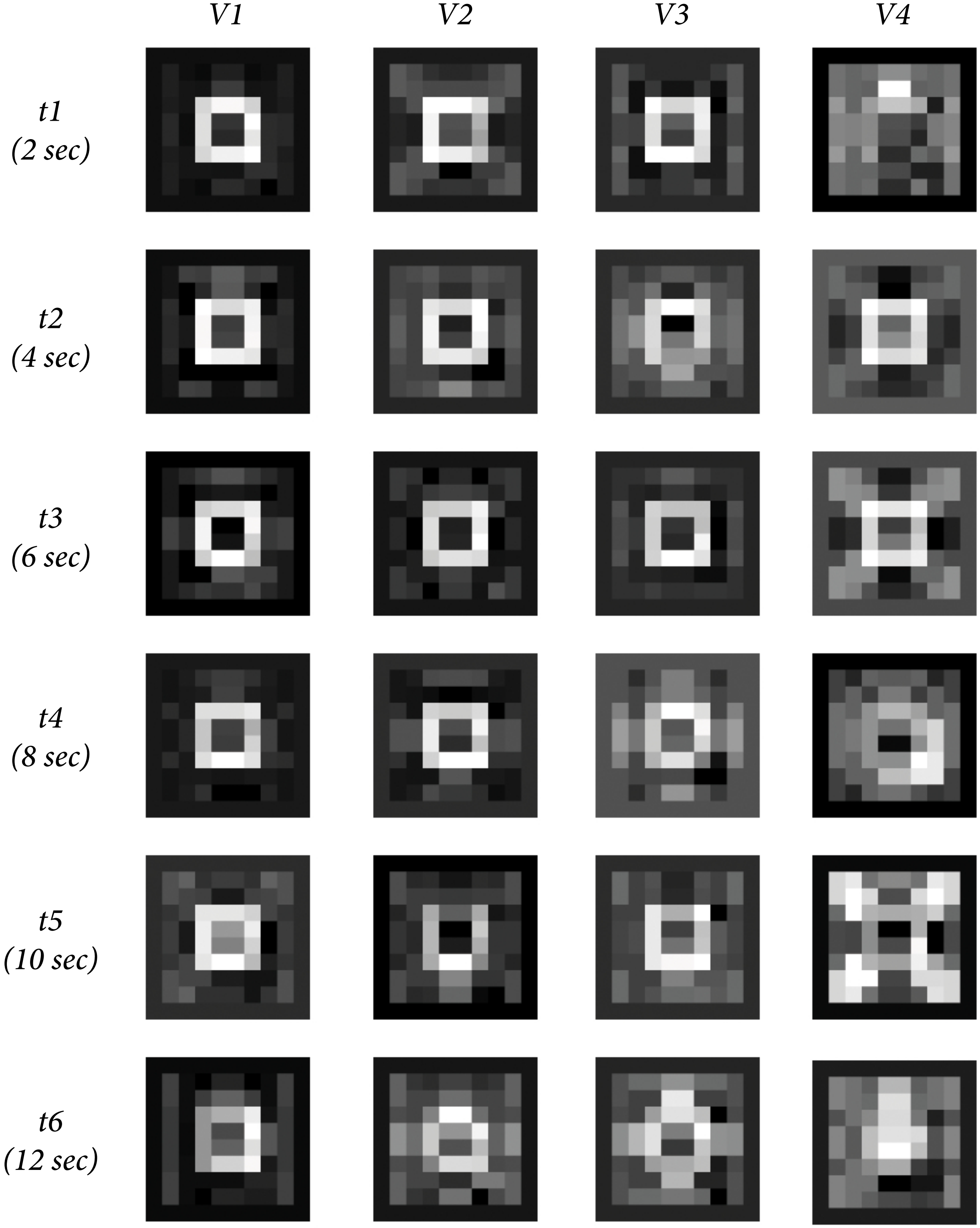

Figure 4.1: Reconstructions of a specific image based on a visual area and time sample. The visual areas are shown horizontally and the time samples vertically. For example, the upper right reconstruction corresponds with the reconstruction in which the algorithm only uses data from visual area V4 and only uses data from the first time sample.

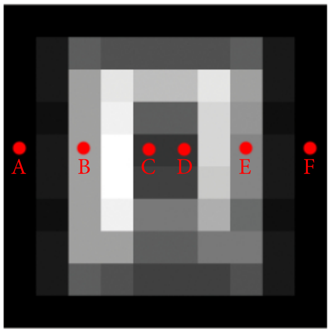

Figure 4.2: The average reconstruction of all samples in the training set. This image contains six points corresponding with a particular pixel. These points are compared against each other in Section 5.3.

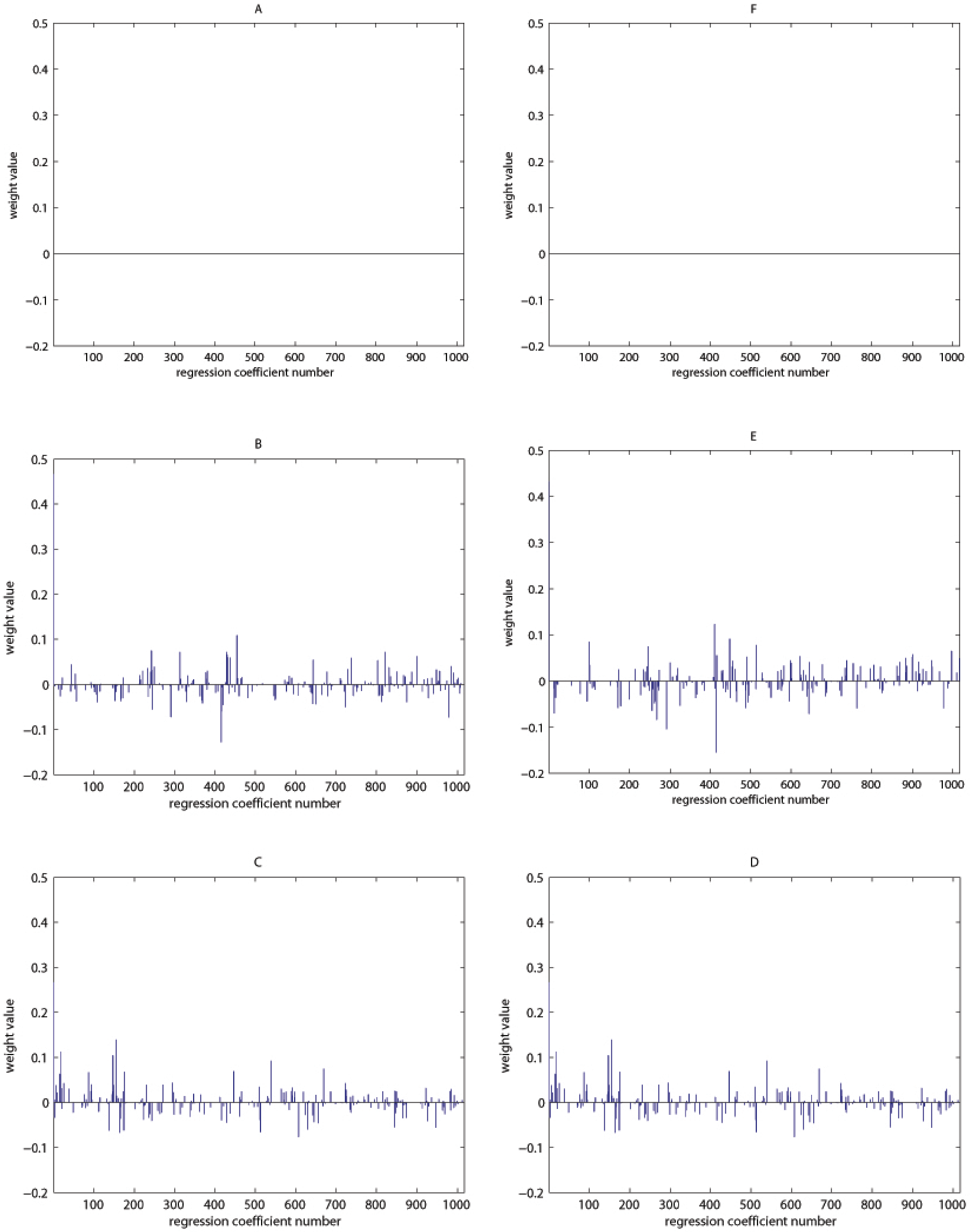

Figure 4.3: The plots of the coefficients corresponding with each point shown in Figure 4.2. The image that is reconstructed is the same as the reconstructed image in Figure 4.1. Here visual area V1 is used using all time samples.

Chapter 5

Discussion

In this chapter conclusions will be presented accompanied with discussions about the data shown in Chapter 4.

5.1 Comparing the reconstructions for the visual cortex

What you can also see in Table 4.1 is that when the value of ν is equal or greater than $1e^0$, only one predictor variable is nonzero and this is the offset. As a result the reconstructed image is equal to the reconstruction which gives the least error. Even greater values of ν will not decrease nor improve the reconstructions, because only the offset is left for the reconstruction and it is not really surprising that the image reconstructor can not perform better.

Values of ν equal or smaller than $1e^{-5}$ generally use all the available coefficients. This does not result in bad reconstructions. They often reconstruct better, because all coefficients are used so that the model has a lot of information that can be used to carefully reconstruct the images. This is also what can be seen by looking at the SSE values for small ν-values in Table 4.1. The reconstruction quality tends to reach it's optimum for ν-values from $1e^{-4}$ and smaller. For the purpose of finding the most important brain regions, a subset of all coefficients is needed. Additionally, a drawback of using all coefficients is that too much detail is stored in the model which can result in overfitting. Also lasso regression is used and it would be redundant if there was no need to reduce the number of coefficients used during the reconstruction process. Lasso regression shrinks all coefficients and pushes them to zero if they do not contribute enough to the reconstructions. This results in less coefficients being used for the reconstructions. This is more efficient and gives interpretable models. A drawback is the risk of underfitting.

A thing to note is that in Table 4.1 the coefficients for visual cortices V1, V2 and V3 are getting sparse faster than the coefficients for visual cortex V4. The reason for this may be that V4 reacts less strong to contrast differences and therefore needs more coefficients to produce good reconstructions.

5.2 Comparing the reconstructions for different time samples

Given the hemodynamic response function (shown in Figure 3.1), the oxygen level should be at its peak between time sample t2 and t3. And because the contrast defined images are shown to the subjects for a continuous period of 12 seconds, the reconstruction quality should not drop significantly after time sample t3. However, this is not what can be seen in Table 4.2. The results show that time sample t1 has the best reconstruction quality. Also the example reconstructions in Figure 4.1 show different results than expected. In these reconstructions, the best reconstructions are at time sample t3, but the reconstructions will not stay the same for later time samples. On the contrary, reconstructions for time sample t4 and later all show a decrease in quality. However, this is just one example out of 480 samples.

What also can be seen in Table 4.2 and in the reconstructed images in Figure 4.1 is that visual area V1 (together with area V2) is performing a lot better than the visual areas V3 and V4. This is what is expected, because Miyawaki concluded that V1 contains the most reliable information [13]. Additionally, higher visual areas are less well understood. However, the reconstructions from visual area V3 and V4 are not really bad: it is still possible to recognize the image that is supposed to be reconstructed. The potential reason for this is that the reconstruction process is not that difficult, because the regression coefficients are trained on 20 different contrast defined images. If they were trained on thousands of different images, chances are higher that the reconstructions are a lot worse. An important thing to note, is that the reconstructed images are all seen in the training set. So the reconstruction task essentially becomes a classification task. When fMRI data is shown for an image that has not been seen in the training set, the reconstruction would probably not even look like it.

5.3 Comparing the values for the regression coefficients for various points

In Figure 4.3 it is clear to see that the beta values are different for the selected pixels. This is what is expected because elastic net tries to compute a particular model for each pixel. Point A and F are similar, because they both are pixels that remain black for all samples in the data set. Accordingly, one would expect that the regression coefficients are (nearly) the same. This is indeed what can be seen in the plots for pixel A and F. The line remains zero for all regression coefficients.

For the other pixels, the model changes accordingly to find the best values for the regression coefficients given the restrictions from lasso regression. Looking at points B and E. These pixels get a color different than black in some (or all) of the samples, but the points are similar because they are mostly symmetrical. This also holds for points C and D. Therefore, you would expect that these symmetrical points have nearly the same regression coefficients. As can be seen in Figure 4.3, the plots indeed show some similarity. This can be seen by looking at some of the eye catching properties of the pattern and by looking at the correlation values shown in Table 5.1. The pixel based image reconstructor really computes a different model for different pixels. An exception to this are points C and D, which happen to be very similar.

	B	C	D	E
B	1.0000	0.3770	0.3770	0.6661
C	-	1.0000	1.0000	0.3254
D	-	-	1.0000	0.3254
E	-	-	-	1.0000

Table 5.1: The correlation values between the points B, C, D and E shown in Figure 4.2. The correlation between points A and F are not shown, because the correlation with these coefficients can't be computed because these coefficients consist solely of zeros. Half of the table is filled with dashes to keep the table easy to read. The values for the empty part are the same as the filled in part, because the table is symmetrical.

5.4 Final conclusions

The reconstructions from V1, V2, V3 and V4 were all reasonably good. In all reconstructions it was possible to identify the original image. However, what can be seen in the tables with SSE values (Table 4.1 and Table 4.2) is that reconstructions from visual areas V1 and area V2 generally have the same SSE values. Visual area V3 has a slightly higher SSE and visual area V4 performs the worst by having significantly higher SSE values. This is also what can be seen for the 24 example reconstructions in Figure 4.1. By using visual areas V1 and V2 for the reconstructions, the reconstructed images resemble each other significantly. The reconstructions from V3 are slightly worse and V4 performs the worst. So for reconstructing contrast defined images, visual areas V1 and V2 should be used.

The comparison of the time samples versus reconstruction quality in Table 4.2 showed that the SSE values increase per time sample. Time sample t1 has the best reconstructions and time sample t6 the worst. This is not what is expected, because the subjects are shown the contrast defined images for a continuous period of 12 seconds, which should result in stable SSE values after time sample t3, because this is were the peak for the hemodynamic response function is.

What is also shown in Section 4.3 is a way to find the most relevant regression coefficients. This is illustrated by first determining the best ν-value by selecting the ν-value that results in reconstructions that only use a few regression coefficients. These regression coefficients correspond to a particular voxel in the brain. By analyzing these voxels, it is possible to find the relevant spots in the brain that are used for seeing images.

5.5 Future research

In this study elastic net was used to reconstruct fairly simple images. There were only 20 contrast defined images. For future research, this can be extended to using more images, or using random images to train and use the contrast defined images to test. By having enough random images, it should be possible to find relationships between brain activity and the pixels from an image. Other ways to extend this research is using something different than simple contrast defined images. It would be great to use real world pictures with color, though we are not at that point right now. Somewhere in between would be more realistic.

What can be studied as well are the precise differences between the visual areas or what a visual area exactly responds to. In this study, the visual areas were only used to to generate reconstructions. A thorough study of a particular visual area can be useful to gain insight in the reason why this visual area is good or bad for reconstructions.

Chapter 6

Appendix

```
clear all;

lambda = 1e-3;
nu = 1e-3;

%which visual area to use
v1 = true;
v2 = false;
v3 = false;
v4 = false;

%walk over all time samples
for t=1:6
    timesamples = [t];

    read_neurondata;

    ninput = size(testdata,2);
    npixels = size( testdesign, 2 );

    %get unique indices for images
    [a,unique_indices,c] = unique(testdesign, 'rows');
    nsamples = size(unique_indices);

    initial_beta0 = 0;
    initial_beta = zeros(ninput,1);

    options = struct('offset', 1, 'maxiter', [100000], 'tol', [1e-2]);

    normalizeData;

    reconstructedImage = zeros(nsamples(1),100);
    storedBetas = zeros(nsamples(1), 100);
    sse_image = zeros(1, nsamples(1));

    %walk over all unique indices
    for i=1:20
        %retrieve position of unique index
        n = unique_indices(i);

        for pixelNr=1:npixels
            X = testdata';
            X(:,n) = []; %leave one out

            Y = testdesign(:,pixelNr)';
            Y(n) = []; %leave one out

            %compute weights using elastic net
            [beta, beta0] = elastic(X,Y,nu,lambda,options,initial_beta, ...
                 initial_beta0);
            storedBetas(i,pixelNr) = ninput - sum([beta0 beta'] == 0) + 1;

            mriData = testdata(n,:);
            reconstructedImage(i,pixelNr) = beta0 + mriData * beta;
        end

        %adjust values to fit between 0 and 1
        minimum = min(reconstructedImage(i,:));
        maximum = max(reconstructedImage(i,:));
        spread = maximum-minimum;
        reconstructedImage(i,:) = ...
             (reconstructedImage(i,:)-minimum)./spread;

        %reconstruct image for all figures
        %draw reconstructed image to the screen
        clf, subplot(121), imagesc( reshape( testdesign(n,:),10,10), ...
             [min( testdesign(n,:) ), max( testdesign(n,:) )] ); ...
             title('origineel');
        colormap gray; axis square; axis off; drawnow
        subplot(122), imagesc( reshape(reconstructedImage(i,:),10,10), ...
             [min(reconstructedImage(i,:)), ...
             max(reconstructedImage(i,:))]); title('reconstructed');
        colormap gray; axis square; axis off; drawnow

        %compute SSE for single pixels and for the nth image
        sse_per_pixel = (testdesign(n,:) - reconstructedImage(i,:)).^2;
        sse_image(i) = sum(sse_per_pixel);
    end

    avg_zeros = mean(storedBetas(:,45));
    avg_zeros_perc = round(100 * avg_zeros / (ninput+1));
    avg_sse = mean(sse_image(:));
    fprintf('average sse per image: %f\n\n', avg_sse);
end
```

Bibliography

[1] http://www.aiaccess.net/English/Glossaries/GlosMod/e_gm_ridge.htm.

[2] http://www.fmrib.ox.ac.uk/education/fmri/introduction-to-fmri/what-does-fmri-measure/.

[3] D. D. Cox and R. L. Savoy. Functional magnetic resonance imaging (fMRI) brain reading: detecting and classifying distributed patterns of fMRI activity in human visual cortex. *Neuroimage*, 19:261–270, 2003.

[4] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

[5] J. V. Haxby, M.I. Gobbini, M. L. Furey, A. Ishai, J. L. Schouten, and P. Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293:2425–2430, 2001.

[6] J. Hegdé and D. C. Van Essen. Selectivity for complex shapes in primate visual area V2. *Journal of Neuroscience*, 20:1–6, 2000.

[7] Y. Kamitani and F. Tong. Decoding the visual and subjective contents of the human brain. *Nature*, 8:679–685, 2005.

[8] Y. Kamitani and F. Tong. Decoding seen and attended motion directions from activity in the human visual cortex. *Current Biology*, 16:1096–1102, 2006.

[9] K. N. Kay and J. L. Gallant. I can see what you see. *Nature Neuroscience*, 12:245–246, March 2009.

[10] K. N. Kay, T. Naselaris, R. J. Prenger, and J. L. Gallant. Identifying natural images from human brain activity. *Nature*, 452:352–355, 2008.

[11] S. Laureys, M. Boly, and G. Tononi. *Functional Neuroimaging in The Neurology of Consciousness: Cognitive Neuroscience and Neuropathology.* Academic Press-Elsevier, 2009.

[12] G. Mather. The visual cortex. http://www.lifesci.sussex.ac.uk/home/George_Mather/Linked%20Pages/Physiol/Cortex.html.

[13] Y. Miyawaki, H. Uchida, O. Yamashita, M. Sato, Y. Morito, H. C. Tanabe, N. Sadato, and Y. Kamitani. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron*, 60:915–929, 2008.

[14] M. E. Raichle and M. A. Mintun. Brain work and brain imaging. *The Annual Review of Neuroscience*, 29:449–476, 2006.

[15] C. S. Roy and C. S. Sherrington. On the regulation of the blood-supply of the brain. *Journal of Physiology*, 11:85–158, January 1890.

[16] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

[17] B. Thirion, E. Duchesnay, E. Hubbard, J. Dubois, J. B. Poline, D. Lebihan, and S. Dehaene. Inverse retinotopy: inferring the visual content of images from brain activation patterns. *Neuroimage*, 33:1104–1116, 2006.

[18] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1996.