On the Arrangement of Visual Features for Efficient and Accurate Prediction using a Linear Model

Monika Shah

Follow this and additional works at: https://digitalcommons.memphis.edu/etd

Recommended Citation

https://digitalcommons.memphis.edu/etd/2145

This Thesis is brought to you for free and open access by University of Memphis Digital Commons. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of University of Memphis Digital Commons. For more information, please contact khggerty@memphis.edu.
Acknowledgement

I would like to express my sincere gratitude to my adviser Dr. Bonny Banerjee for giving me opportunity to work with him and for his continuous guidance and support. I am very thankful for his motivations, patience and immense knowledge. He has been a constant help for this research and in writing this thesis. I would like to thank my thesis committee members: Dr. Madhusudhanan Balasubramanian and Dr. Aaron L Robinson for their time, encouragement and insightful questions. I am thankful to my funding support for this work from Institute of Intelligence system and Department of Electrical and Computer engineering at University of Memphis. I am grateful to my parents Mr. Umesh Raut and Mrs. Phulpati Raut for their love and support.
Abstract

Assuming the brain is a prediction machine, we investigate the arrangement of 2D visual features (Gabor filters) such that visual prediction can be carried out efficiently. The contributions of this thesis are threefold: (1) we prove that a 1-mode tensor can be predicted from an n-mode and an (n-1)-mode tensors; (2) using a linear model to predict visual features at varying distances from a given location, we show that prediction error increases smoothly and monotonically with increase in distance and increase in degree of sparsity of the prediction model; (3) we show that the visual features can be arranged in a 2D grid such that the total wiring length can be minimized without compromising prediction accuracy. All experiments are carried out using 110 high-resolution natural images.
# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.1 Notation</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Literature Review</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1 Linear Model</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1.1 Linear Regression Model</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1.2 Multiple Linear Regression Model</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1.3 Logistic Regression</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.1.4 Generalized Linear Regression Model</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.1.5 Ridge Regression</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1.6 Support Vector Machine (SVM)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.2 Matrix Factorization</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.2.1 Dictionary Learning</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.2.2 Non-negative Matrix Factorization (NMF)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.2.3 Spherical Clustering</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.2.4 Sparse Coding</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Design Overview</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3.1 A linear prediction model using n-mode correlation tensor</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3.1.1 Inner Product</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3.1.2 Outer Product</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3.1.3 Multidimensional Matrix Transpose Operation</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3.1.4 Tensor Reconstruction</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>3.2 Limitations</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>n-mode tensor to n two-mode tensor decomposition</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4.1 Isserlis' Theorem</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4.2 Independent Variables</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4.3 Conditional Independence(Deduction by Dr. Bonny Banerjee)</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Methodology</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>5.1 Gabor Filter</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>5.2 Feature Extraction</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>5.3 Cross-Correlation</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5.4 Central Patch and Neighbor Patch</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5.5 Model Learning</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>5.6 Optimal Predictor</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>5.7 Prediction</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>5.8 Sparsity</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>Algorithm</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>6.1 Algorithm</td>
<td>19</td>
</tr>
</tbody>
</table>
7 Results and Evaluation

7.1 Analysis of learned model 22
7.2 Prediction Error 22
  7.2.1 Mean Square Error (MSE) 22
  7.2.2 Cosine Similarity 22
  7.2.3 Mean Square Error (MSE) 22
  7.2.4 Sample Standard Deviation 23
7.3 Comparison of prediction error with increase in radius 24
7.4 Hypercolumn 31
7.5 2D Feature Arrangement 31
7.6 Results and Analysis 32
7.7 Prediction Error with increase in sparsity 40

8 Conclusion 42

References 43
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Samples of natural images from the dataset of 110 images</td>
<td>13</td>
</tr>
<tr>
<td>2.</td>
<td>Gabor filter of different orientation for number of feature = 9</td>
<td>15</td>
</tr>
<tr>
<td>3.</td>
<td>Euclidian distance between the original feature vector and the predicted feature vector for $k = 9$ and radius $1,2,4,7,10,13,16,20,25,30,35$ and 40 pixels from the central patch: (a) one non-zero connection strength; (b) 3 non-zero connection strength; (c) 5 non-zero connection strength; (d) 7 non-zero connection strength; (e) 9 non-zero connection strength</td>
<td>25</td>
</tr>
<tr>
<td>4.</td>
<td>Cosine similarity between the original feature vector and the predicted feature vector for $k = 9$ and radius $1,2,4,7,10,13,16,20,25,30,35$ and 40 pixels from the central patch: (a) one non-zero connection strength; (b) 3 non-zero connection strength; (c) 5 non-zero connection strength; (d) 7 non-zero connection strength; (e) 9 non-zero connection strength</td>
<td>26</td>
</tr>
<tr>
<td>5.</td>
<td>Euclidean distance between the original feature vector and the predicted feature vector for $k = 12$ and radius $1,2,4,7,10,13,16,20,25,30,35$ and 40 pixels from the central patch: (a) one non-zero connection strength; (b) 3 non-zero connection strength; (c) 6 non-zero connection strength; (d) 9 non-zero connection strength; (e) 12 non-zero connection strength</td>
<td>27</td>
</tr>
<tr>
<td>6.</td>
<td>Cosine similarity between the original feature vector and the predicted feature vector for $k = 12$ and radius $1,2,4,7,10,13,16,20,25,30,35$ and 40 pixels from the central patch: (a) one non-zero connection strength; (b) 3 non-zero connection strength; (c) 6 non-zero connection strength; (d) 9 non-zero connection strength; (e) 12 non-zero connection strength</td>
<td>28</td>
</tr>
<tr>
<td>7.</td>
<td>Euclidean distance between the original feature vector and the predicted feature vector for $k = 36$ and radius $1,2,4,7,10,13,16,20,25,30,35$ and 40 pixels from the central patch in ??: (a) one non-zero connection strength; (b) 9 non-zero connection strength; (c) 18 non-zero connection strength; (d) 27 non-zero connection strength; (e) 36 non-zero connection strength</td>
<td>29</td>
</tr>
<tr>
<td>8.</td>
<td>Cosine similarity between the original feature vector and the predicted feature vector for $k = 36$ and radius $1,2,4,7,10,13,16,20,25,30,35$ and 40 pixels from the central patch: (a) one non-zero connection strength; (b) 9 non-zero connection strength; (c) 18 non-zero connection strength; (d) 27 non-zero connection strength; (e) 36 non-zero connection strength</td>
<td>30</td>
</tr>
</tbody>
</table>
9. *A hypercolumn consisting of one set of orientation columns and one set of ocular dominance columns in the visual cortex. Electrode 1 penetrating at right angle to the surface encounters neurons responding to the same orientation and the same (right) ocular dominance. Electrode 2 penetrating parallel to the surface encounters neurons responding to the left ocular dominance and to different orientations.*

10. Node-feature arrangement for feature, k = 9 using winner-take-all approach: (a) radius = 1; (b) radius = 2; (c) radius = 4; (d) radius = 7; (e) radius = 10; (f) radius = 13. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.

11. Node-feature arrangement for feature, k = 9 using winner-take-all approach: (a) radius = 16; (b) radius = 20; (c) radius = 25; (d) radius = 30; (e) radius = 35; (f) radius = 40. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.

12. Node-feature arrangement for feature, k = 12 using winner-take-all approach: (a) radius = 1; (b) radius = 2; (c) radius = 4; (d) radius = 7; (e) radius = 10; (f) radius = 13. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.

13. Node-feature arrangement for feature, k = 12 using winner-take-all approach: (a) radius = 16; (b) radius = 20; (c) radius = 25; (d) radius = 30; (e) radius = 35; (f) radius = 40. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.

14. Node-feature arrangement for feature, k = 36 using winner-take-all approach: (a) radius = 1; (b) radius = 2; (c) radius = 4; (d) radius = 7; (e) radius = 10; (f) radius = 13. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.
15. Node-feature arrangement for feature $k = 36$ using winner-take-all approach: (a) radius = 16; (b) radius = 20; (c) radius = 25; (d) radius = 30; (e) radius = 35; (f) radius = 40. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.

16. Prediction error vs sparsity for 9 features: (a) MSE for radius = 1; (b) Cosine Similarity for radius = 1

17. Prediction error vs sparsity for 12 features: (a) MSE for radius = 1; (b) Cosine Similarity for radius = 1

18. Prediction error vs sparsity for 36 features: (a) MSE for radius = 1; (b) Cosine Similarity for radius = 1
Chapter 1

Introduction

Image prediction and image reconstruction has become an essential task in the field of computer vision. At present, there are abundant of research that focuses on predictive models. A large number of linear predictive models and non-linear predictive models exists that are developed for prediction of data. Our goal in this research is to come up with the simplest predictive model and do the analysis of the visual feature arrangement such that the model can give better prediction accuracy.

Apart from computational models, human cognition has been processing optical sensory data and predicting image features formed on retina ever since optical system got evolved for visual purpose. The stark similarity in data processing arrangement in human brain and computationally developed algorithm is conceivable. The field is new and very few works have been done in this regard. We developed the most fundamental linear prediction model and showed the correlation of the features with the hypercolumn[2] arrangement of visual cortex.

This thesis focuses on developing a linear predictive algorithm that predicts the features of neighbor patches (NP) of an image given a central patch (CP). We prove and develop an algorithm to predict 1-mode tensor from n-mode tensor and n-1 mode tensor. We use a linear predictive model to predict visual features at varying distances from a given location. Using the prediction results, we show that as the distance of the prediction location from a given location increases the prediction error increases.

The hypercolumn structure of visual cortex consists of orientation column and ocular dominance column. A hypercolumn of orientation column is perpendicular to a hypercolumn of ocular dominance columns. It consists of vertical arrangement of cells that all responds to cells in the same orientation in the same retinal location. In this research we use a simple linear model to generate a
hypercolumn structure which helps in accurate prediction. We also show that the visual features can be arranged in a 2D grid such that the total wiring length can be minimized without compromising prediction accuracy.

1.1 Notation

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x)</td>
<td>random variable</td>
</tr>
<tr>
<td>(\mathbf{x})</td>
<td>random vector</td>
</tr>
<tr>
<td>(X)</td>
<td>matrix</td>
</tr>
<tr>
<td>(\mathbf{W})</td>
<td>tensor</td>
</tr>
<tr>
<td>(n)</td>
<td>number of neighbor patches</td>
</tr>
<tr>
<td>(N)</td>
<td>number of patches</td>
</tr>
<tr>
<td>(k)</td>
<td>number of features</td>
</tr>
<tr>
<td>(\text{CN})</td>
<td>central node</td>
</tr>
<tr>
<td>(\text{NN})</td>
<td>neighbor node</td>
</tr>
<tr>
<td>(\mathbf{f})</td>
<td>feature vector</td>
</tr>
<tr>
<td>(\mathbf{M})</td>
<td>mean cross-correlation matrix</td>
</tr>
<tr>
<td>(\mathbf{f}_0)</td>
<td>feature vector of CN</td>
</tr>
<tr>
<td>(\mathbf{f}_i)</td>
<td>feature vector of NN</td>
</tr>
<tr>
<td>(\mathbf{M}_0)</td>
<td>mean auto-correlation matrix of features of CN</td>
</tr>
<tr>
<td>(\mathbf{M}_i)</td>
<td>mean cross-correlation matrix between features of CN and (i^{th}) NN</td>
</tr>
<tr>
<td>(\mathbf{M}_{\text{win}})</td>
<td>mean correlation matrix for winner-take-all</td>
</tr>
</tbody>
</table>
Chapter 2

Literature Review

2.1 Linear Model

Let \( Y \in \mathbb{R}^{m \times n} \) is linearly related to \( k \) independent variables \( X \in \mathbb{R}^{m \times k} \) through the parameters \( B \in \mathbb{R}^{k \times n} \) [3, p. 34]

\[
Y = XB + E
\]  

(2.1)

where \( X \) is a design matrix of \( m \) observations on each of the \( k \) explanatory variables, \( E \in \mathbb{R}^{m \times n} \); \( e_i \) is the difference between the true and observed realizations of \( y_i \).

2.1.1 Linear Regression Model

A simple linear regression model is defined as [3, p. 7]

\[
y = \beta_0 + \beta_1 X + e
\]  

(2.2)

where \( y \) is a dependent variable and \( X \) is an independent variable. \( \beta_0 \) and \( \beta_1 \) are the intercept term and the slope parameter respectively and are called as regression parameter, \( e \) represents the difference between the true and observed realization of \( y \).

2.1.2 Multiple Linear Regression Model

A multiple linear regression model can be defined as [3, p. 33]

\[
y = X_1\beta_1 + ... + X_p\beta_p + e
\]  

(2.3)

where \( y \) is a dependent variable and it is linearly dependent on \( p \) independent variables \( X_1, ..., X_p \) respectively. \( e \) represents the difference between the true and observed realization of \( y \).
2.1.3 Logistic Regression

Logistic regression can be defined as\cite[p. 6]{4}

\[
\pi(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}
\] (2.4)

where \( \pi(x) = P[y|x] \) represents the conditional probability of \( y \) given \( x \) while using the logistic distribution. This transformation is defined as

\[
g(x) = \ln\left[\frac{\pi(x)}{1 - \pi(x)}\right]
\] (2.5)

The transformation, \( g(x) \) is linear in its parameters.

2.1.4 Generalized Linear Regression Model

A generalised linear model is defined by three components; a random function, a systematic component and a link function. A random component specifies a distribution for \( Y|X \). The probability density function of the distribution can be assumed as\cite[p. 413]{3}

\[
f(y; \theta, \phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right)
\] (2.6)

where \( \theta \) is natural parameter, \( \phi \) is dispersion parameter and \( a,b,c \) are functions. The systematic component relates a parameter \( \eta \) to the predictors \( X \) as

\[
\eta = \beta^T X = \beta_1 X_1 + ... + \beta_p X_p
\] (2.7)

The link function connects the systematic and the random component through a link function \( g \) as

\[
g(\mu) = \eta
\] (2.8)

where \( \mu = E(Y|X) \)
2.1.5 Ridge Regression

The ridge estimate of linear model in equation (1) is

\[ \hat{B} = (X^T X + kI)^{-1}X^T Y \] (2.9)

where \( k \) is a positive number \( > 1 \). Ridge regression is used when there is multicollinearity in the multiple regression data. The least square estimates of these data are unbiased but their variance is very high. Ridge regression reduces the standard error by adding bias in the regression estimate.

2.1.6 Support Vector Machine (SVM)

In SVM, the data is mapped into a higher dimensional input space and an optimal separating hyperplane is constructed in this space. The SVM classifier is defined as\(^5\)

\[ y(x) = \text{sign}\left[ \sum_{i=1}^{N} \alpha_i y_i \psi(x, x_i) + b \right] \] (2.10)

where \( \psi(x, x_i) = x_i^T x \), \( \alpha_i \) are positive real constant and \( b \) is a real constant. The classifier is constructed as,

\[ w^T \phi(x_k) + b \geq 1, \text{if } y_k = +1 \] (2.11)

\[ w^T \phi(x_k) + b \leq 1, \text{if } y_k = -1 \] (2.12)

where \( \phi(.) \) is a non-linear function which maps the input space into a higher dimensional space.

2.2 Matrix Factorization

The computation of \( X \) and \( B \) in equation (1) is referred to as matrix factorization.
2.2.1 Dictionary Learning

Dictionary learning is a matrix factorization problem that aims to represent the signal, \( Y \) as the linear combination of dictionary, \( X = [x_1, \ldots, x_k] \in \mathbb{R}^{m \times k} \). The dictionary \( X \) is learned along with a matrix of decomposition coefficient, \( B = [\beta_1, \ldots, \beta_n] \in \mathbb{R}^{k \times n} \).

2.2.2 Non-negative Matrix Factorization (NMF)

It is a matrix factorization problem such that,

\[
Y = XB
\]  \hspace{1cm} (2.13)

where \( Y \in \mathbb{R}^{m \times n} \) is approximately factorised into \( X \in \mathbb{R}^{m \times k} \) and \( B \in \mathbb{R}^{k \times n} \) [6]. Generally, \( k \) is chosen such that it is smaller than \( m \) or \( n \) making the dimensions of \( X \) and \( B \) matrix smaller than the original matrix \( Y \). The resulting matrices are the compressed version of original data matrix.

2.2.3 Spherical Clustering

Spherical Clustering can be defined as,

\[
Y = XB
\]  \hspace{1cm} (2.14)

where, \( \beta_i \in \mathbb{R}^k \) are called code vectors and \(|\beta_i|_0 \leq 1\), i.e. a single code vector may have at most single non-zero element and \(|x_j|_2 = 1\) [7].

2.2.4 Sparse Coding

Each \( y \) contains a sparse representation over \( \beta \) over \( X \) if \( y \) can be represented as \( \kappa \) atoms in \( X \) and \( \kappa << k \). Given \( Y \), sparse coding can be defined as a matrix factorization problem such as

\[
\min_{X \in X, B \in B} ||Y - XB||_F^2
\]  \hspace{1cm} (2.15)
where $||.||_F$ denoted the matrix Frobenius norm. $X$ and $B$ are convex set, $X$ is bounded and $X$ is a set of matrices in $\mathbb{R}^{m \times k}$ whose columns are on the surface of the unit sphere of the $l_2$ norm and $B \in \mathbb{R}^{k \times n}$. 
Chapter 3
Design Overview

3.1 A linear prediction model using n-mode correlation tensor

3.1.1 Inner Product

The inner product \( \langle \mathcal{A}, \mathcal{B} \rangle \) between two tensors \( \mathcal{A}, \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_n} \) is defined as

\[
\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1, i_2, \ldots, i_n} a_{i_1,2,\ldots,n} b_{i_1,2,\ldots,n}
\] (3.1)

3.1.2 Outer Product

Outer product \([8]\) of \( n \) vectors is defined by,

\[
\mathcal{W} = E[u^{(1)} \odot u^{(2)} \odot \ldots \odot u^{(n)}]
\] (3.2)

where, \( u^{(1)} \in \mathbb{R}^{n_1}, \ldots, u^{(n)} \in \mathbb{R}^{n_n} \) and \( \mathcal{W} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_n} \). The entries of \( \mathcal{W} \) are completely determined by \( w_{i_1i_2\ldots i_n} = u_{i_1}u_{i_2}\ldots u_{i_n} \).

3.1.3 Multidimensional Matrix Transpose Operation

A multidimensional matrix transpose is indicated by the letter \( "T" \) followed by the dimensions being transposed in parentheses, all of which is a superscript to the multidimensional matrix being transposed. The transpose of the \( x \)th dimension and \( y \)th dimension of multidimensional matrix \( \mathcal{A} \) is indicated by \( \mathcal{A}^{T(x,y)} \). A multidimensional matrix transpose can be done for any two dimensions. Let \( x \) and \( y \) be two dimensions being transposed. If \( \mathcal{B} = \mathcal{A}^{T(x,y)} \) then \( b_{ijk\ldots y\ldots x\ldots q} = a_{ijk\ldots x\ldots y\ldots q} \).

Transpose of \( x \) and \( y \) dimension of \( \mathcal{W} \) \([9]\),

\[
\mathcal{W}^{T(x,y)} = E[u^{(y)} \odot u^{(2)} \odot \ldots \odot u^{(x)} \odot \ldots \odot u^{(n)}]
\] (3.3)

where, \( x = 1 \) and \( y = 2,3,\ldots,n \).
3.1.4 Tensor Reconstruction

A \( u^{(i)} \) tensor can be reconstructed from n-mode outer product tensor 
\( \mathcal{W}^{T(x,y)} \) by computing dot product of \( \mathcal{W}^{T(x,y)} \) with n-1 dimensional outer product tensor(\( X \)).

where, \( X = u^{(2)} \circ u^{(3)} \circ ... \circ u^{(x)} \circ ... \circ u^{(n)} \)

\[
\hat{u}^{y} = \langle \mathcal{W}^{T(x,y)}, X \rangle
\]  \hspace{1cm} (3.4)

\[
\hat{u}^{y} = \langle (u^{(y)} \circ u^{(2)} \circ ... \circ u^{(x)} \circ ... \circ u^{(n)}), (u^{(2)} \circ u^{(3)} \circ ... \circ u^{(x)} \circ ... \circ u^{(n)}) \rangle
\]  \hspace{1cm} (3.5)

where \( u^{(i)} \) is normalized using L2 norm. Using equation A.3.7 [10], equation (3.5) can be reduced to

\[
\hat{u}^{y} = u^{(y)} \langle u^{2}, u^{2} \rangle \langle u^{3}, u^{3} \rangle ... \langle u^{x}, u^{x} \rangle ... \langle u^{n}, u^{n} \rangle
\]  \hspace{1cm} (3.6)

The tensor \( u^{i} \) is normalized using L2 norm, so \( \langle u^{i}, u^{i} \rangle = 1 \). Therefore,

\[
\hat{u}^{y} = u^{(y)} \cdot 1 \cdot 1 \cdot ... \cdot 1 \cdot ... \cdot 1
\]  \hspace{1cm} (3.7)

\[
= u^{(y)}
\]  \hspace{1cm} (3.8)

3.2 Limitations

The size of an n-mode tensor \( \mathcal{W} \) is \( k^n \), where there are n modalities, each is of k dimensions on average. This makes the model unusable for practical purposes. One way to reduce this size is by approximating it as n 2-mode tensors.
Chapter 4

n-mode tensor to n two-mode tensor decomposition

4.1 Isserlis’ Theorem

Isserli’s theorem[11] is a formula that allows to compute higher-order moments of the multivariate normal distribution in terms of its covariance matrix. If \((x_1, x_2, ..., x_n)\) is a zero mean multivariate normal random vector, then

\[
E[x_1 x_2 ... x_n] = \sum_{p \in P_2} \prod_{i,j \in p} E[x_i x_j]
\] (4.1)

where the sum is over all the pairings of \(\{1,...,n\}\), i.e. all distinct ways of partitioning \(\{1,...,n\}\) into pairs \(\{i,j\}\), and the product is over the pairs contained in \(p\).

For 4\(^{th}\) order moment,

\[
E[x_1 x_2 x_3 x_4] = E[x_1 x_2] E[x_3 x_4] + E[x_1 x_3] E[x_2 x_4] + E[x_1 x_4] E[x_2 x_3]
\] (4.2)

If \(n\) is odd, then there does not exist any pairing of \([i,j]\). Under this hypothesis, Isserlis’ theorem implies that:

\[
E[x_1 x_2 ... x_n] = 0
\] (4.3)

Limitations: Isserlis’ theorem assumes that \((x_1, x_2, ..., x_n)\) are obtained from a joint multivariate normal distribution using marginal distribution.

4.2 Independent Variables

The expected value of the product of the two random variables is equal to the product of their expectations if the random variables are independent. Let \(x\) and \(y\) be independent random variables. Then, the two random variables are mean
independent, which is defined as,

\[ E[xy] = E[x]E[y] \tag{4.4} \]

If random variables \(x_1, x_2, \ldots, x_n\) are mutually independent, then,

\[ E\left[ \prod_{i=1}^{n} x_i \right] = \prod_{i=1}^{n} E[x_i] \tag{4.5} \]

4.3 Conditional Independence (Deduction by Dr. Bonny Banerjee)

Let \(x\) and \(y\) be two random variables.

\[ E[xy] = E[xE[y|x]] \tag{4.6} \]

where \(x_i\) random variables are conditionally independent given a random variable \(x\).

\[ E[x_1 x_2 \ldots x_i \ldots x_n] = E[x E[x_1|x]E[x_2|x] \ldots E[x_i|x] \ldots E[x_n|x]] \tag{4.7} \]

where, \(i = 1, 2, \ldots, n\). We need to deduce equation (4.7) in terms of expectation of product of two variables.
Chapter 5
Methodology
In this research, we predict neighbor patches of an image given a central patch. An
arbitrary patch is considered as the central patch. The patches at radius r and at
particular angular direction from the central patch is considered as the neighbor
patches. We developed a dataset of natural images of minimum size 1024×1024
pixels and maximum size 3000× 3000 pixels. Patches of size 21×21 pixels are
obtained from the input images. Neighbor patches are obtained at distances
distances 1,2,4,7,10,13,16,20,25,30,35 and 40 pixels from the central patch. We
apply gabor filter on the patches to obtain the local features.
Figure 1. Samples of natural images from the dataset of 110 images
5.1 Gabor Filter

Gabor filters is a linear filter used for texture analysis. Frequency and orientaion representation of gabor filter resembles the human visual system. They are commonly used for texture representation and discrimination. A Gabor filter \( g(u,v) \) can be represented as the Gaussian function modulated by a complex sinusoidal signal\[12\], i.e.

\[
g(u,v) = \frac{1}{2\pi\sigma_u\sigma_v} e^{\left[ -\frac{1}{2} \left( \frac{\tilde{u}^2}{\sigma_u^2} + \frac{\tilde{v}^2}{\sigma_v^2} \right) \right]} e^{2\pi j W \tilde{u}} \tag{5.1}
\]

\[
\begin{align*}
\tilde{u} &= u\cos\theta + v\sin\theta \\
\tilde{v} &= -u\sin\theta + v\cos\theta
\end{align*}
\tag{5.2}
\]

where \( \sigma_u \) and \( \sigma_v \) are the scaling parameters of the filter, \( \theta \) specifies the orientation of the gabor filters and \( W \) is the radial frequency of the sinusoid. In this research, the parameters selected for the gabor filter are; orientation, \( \theta = [0^\circ, 20^\circ, 40^\circ, 60^\circ, 80^\circ, 100^\circ, 120^\circ, 140^\circ, 160^\circ] \) when 9 number of features are selected.

5.2 Feature Extraction

Gabor filter is applied on the small patches of the natural images to extract the features i.e. we extract local features from the images. The negative features are eliminated by subtracting the minimum feature from the features vector. A vector \( x \in \mathbb{R}^{k \times 1} \) of small patch can be normalized as

\[
x_{\text{norm}} = \frac{x}{||x||_2} \tag{5.3}
\]

where \( ||x||_2 = \sqrt{\sum_{i=1}^{k} x_i^2} \). The gabor filters are vectorized into column vector and are normalized using equation (5.3) so that the range of data is within \([-1, 1]\).

A Bank of gabor filter is applied on the small patches to obtain the features.
as

\[ f_\theta = \langle x, g(u, v) \theta \rangle \]  \hspace{1cm} (5.4)

where, \( f_i \) is the feature obtained by the dot product of an input patch \( (x) \) with a gabor filter \( (g(u, v)_i) \) of \( i^{th} \) orientation.

\[ \text{Figure 2. Gabor filter of different orientation for number of feature = 9} \]

5.3 Cross-Correlation

Correlation is an important measure of the degree to which two or more random variables have some relationship. For example, if \( x \) and \( y \) are correlated, then outcomes of \( x \) might be used to predict outcomes of \( y \). The cross-correlation of random variables \( x \) and \( y \) is a measure of the linear relationship between \( x \) and \( y \) \[13, \text{p. 282}\].

\[ R_{xy} = E[xy] \] \hspace{1cm} (5.5)

where, \( E \) is the expectation.

For two random vectors \( x = (x_1, ..., x_m)^T \) and \( y = (y_1, ..., y_n)^T \), each containing random elements whose expected value and variance exist, the cross-correlation matrix of \( x \) and \( y \) is defined by \[14, \text{p. 337}\]

\[ R_{xy} = E[xy^T] \] \hspace{1cm} (5.6)

5.4 Central Patch and Neighbor Patch

A patch in an arbitrary location is considered as a central patch and patches at distance \( r \) and at certain angular directions are considered as the neighbor patches. The orientation of the gabor filter for different number of features are as follows
Table 1. Orientation of the Gabor filter for feature 9, 12 and 36 respectively

<table>
<thead>
<tr>
<th>Feature</th>
<th>Angular Direction (Degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0 : 20 : 340</td>
</tr>
<tr>
<td>12</td>
<td>0 : 15 : 345</td>
</tr>
<tr>
<td>36</td>
<td>0 : 5 : 355</td>
</tr>
</tbody>
</table>

5.5 Model Learning

We train a model such that it consists of the connection strength between the features of central patch and the features of neighbor patches. The model is represented by the expectation of correlation matrix between the central patch and the neighbor patch for total number of patches in the training data. The expected correlation matrix is obtained as,

\[
M_0 = \frac{\sum_{j=1}^{N} R_{f_{j,0}f_{j,0}}}{N} \tag{5.7}
\]

\[
M_i = \frac{\sum_{j=1}^{N} R_{f_{j,0}f_{j,i}}}{N}, i = \{1, 2, ..., n\} \tag{5.8}
\]

where \(f_{j,0}\) is the features of central patch and \(f_{j,i}\) is the features of neighbor patch of \(j^{th}\) observation. \(M_0\) is the expected auto-correlation matrix of features of central patch, and \(M_i\) is the expected cross-correlation matrix between features of central patch and \(i^{th}\) neighbor patch.

Each element in the expected auto-correlation matrix represents the connection strength between the features of central patch. Each element in the expected cross-correlation matrix represents the connection strength between the features of central patch and the features of neighbor patches.

5.6 Optimal Predictor

We obtain the optimal predictor as

\[
W_i = M_0^{-1}M_i \tag{5.9}
\]
\[ W_i = (E[f_0f_0^T])^{-1} E[f_0f_i^T] \] (5.10)

where \( f_0 \in \mathbb{R}^{k \times 1} \) is feature vector of central patch and \( f_i \in \mathbb{R}^{k \times 1} \) is feature vector of neighbor patch. \( W_i \) is the optimal predictor. It is used to predict the \( i^{th} \) neighbor patch from any central patch.

### 5.7 Prediction

We use optimal predictor to predict the features of neighbor patch given the features of central patch as

\[ \hat{f}_i = W_i f_0 \] (5.11)

where \( W_i \) is the optimal predictor for the \( i \)th neighbor and \( f_0 \) is the features of central node.

### 5.8 Sparsity

Sparsity is a measure of how sparse the matrix is. We introduce sparsity in the optimal predictor matrix by keeping more strongly connected correlation values and make the rest zero. We observe the changes in the prediction error with respect to change in sparsity. We measure the degree of sparsity as

\[ \text{sparsity} = (1 - \frac{\text{nonzerofeatures}}{\text{totalfeatures}}) \times 100 \] (5.12)

We introduce five orders of sparsity for all the features as

<table>
<thead>
<tr>
<th>Sparsity</th>
<th>Full</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>27</td>
<td>18</td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

The maximum sparsity is termed as winner-take-all where the maximum connection strength between the features of central patch and the features of...
neighbour patch are stored and the rest of the connection strength in the expected correlation matrix are discarded and are replaced by zero. It is observed that the diagonal elements of the expected correlation matrix are the maximum connection strength. The first, second and third sparsity are introduced in the expected correlation matrix by keeping the $p^{th}$ maximum connection strength, where $p$ is each element in table 2.
Chapter 6
Algorithm

6.1 Algorithm

1. The input images are of high resolution and the minimum intensity of each image is zero.

2. Small patches are obtained from the image, and each patch is converted into a column vector, \( x \)

3. Generate a gabor filter bank and convert them into column vectors, \((g(u, v))_i\), where \( i \) is the orientation of the gabor filter.

4. Normalize the patch and gabor filter as

\[
\tilde{x} = \frac{x}{||x||_2} \quad (6.1)
\]

\[
\tilde{g}(u, v) = \frac{g(u, v)}{||g(u, v)|_i||_2} \quad (6.2)
\]

5. Feature vectors are obtained by the dot product of normalized patch and normalized gabor filters

\[
f = \langle \tilde{x}, \tilde{g}(u, v) \rangle \quad (6.3)
\]

6. Convert feature vector into positive values by subtracting each element by its minimum element and normalize it as

\[
\tilde{f} = \frac{f_i - \min(f)}{||f_i - \min(f)||_2} \quad (6.4)
\]

7. Expected Correlation Matrix

\[
M_0 = \frac{\sum_{j=1}^{N} R_{\tilde{f}_i,0} \tilde{f}_{i,0}}{N} \quad (6.5)
\]
\[ M_i = \frac{\sum_{j=1}^{N} R_{\tilde{f}_j, 0 \tilde{f}_j, i}}{N}, \quad i = \{1, 2, ..., n\} \]  \hspace{1cm} (6.6)

where \( n \) is the number of neighbors and \( N \) is the total number of patches.

8. Optimal Predictor

\[ W_i = M_0^{-1} M_i \]  \hspace{1cm} (6.7)

9. Prediction

\[ \hat{f}_i = W_i^T \tilde{f}_0 \]  \hspace{1cm} (6.8)

where \( \tilde{f}_0 \) is the feature of the central patch.

10. Prediction error

(a) Mean square error (MSE)

\[ MSE = \frac{\sum_{i=1}^{N} ||f_i - \hat{f}_i||_2^2}{N} \]  \hspace{1cm} (6.9)

(b) Cosine Similarity

\[ CosineSimilarity = \left| \left\langle \frac{x}{||x||_2}, \frac{\hat{x}}{||\hat{x}||_2} \right\rangle \right| \]  \hspace{1cm} (6.10)

11. Feature Arrangement

\[ \hat{M}_{ij} = \frac{M_{ij}}{max M_{0j}} \]  \hspace{1cm} (6.11)

where \( i \in \{0, 1, ..., n\} \) and \( j \in \{1, ..., k\} \) is each row of \( M_i \), where \( n \) is the number of neighbor, and \( M_0 \) is the expected auto-correlation matrix of central patch and \( M_i \) is the expected correlation matrix of central patch and \( i^{th} \) neighbor patch where \( i \geq 0 \).
where $m_r \in \max(\hat{M}_{ij})$, $g(u, v)_r$ is the gabor filter, $r \in \{1, \ldots, k\}$ and $k$ is the number of features.
Chapter 7
Results and Evaluation

7.1 Analysis of learned model

In the experiment, it is observed that connection strength between the features of central patch and the features of neighbor patch is maximum in the diagonal elements and the connection strength decreases gradually from the diagonal elements to the non diagonal elements.

7.2 Prediction Error

We calculate prediction error using two metrics:

7.2.1 Mean Square Error (MSE)

\[
MSE = \frac{\sum_{i=1}^{N} ||x_i - \hat{x}_i||^2_2}{N}
\]  

(7.1)

where \(x_i\) is the true features of the neighbour patches and \(\hat{x}_i\) is the predicted features of the neighbor patches.

7.2.2 Cosine Similarity

\[
\text{CosineSimilarity} = \left\langle \frac{x}{||x||_2}, \frac{\hat{x}}{||\hat{x}||_2} \right\rangle
\]

(7.2)

We obtain the mean and the standard deviation of the mean square error and cosine similarity.

7.2.3 Mean Square Error (MSE)

Sample Mean of random variables \(\{x_1, ..., x_N\}\) is [13, p. 232]

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

(7.3)
7.2.4 Sample Standard Deviation

Sample Standard Deviation \( \sigma_x \) of a random variable \( x \) is the positive square root of its variance \( \sigma_x^2 \) [13, p. 233],

\[
\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2
\]  \hspace{1cm} (7.4)

1 The intermediate calculation of mean can be eliminated by further deducing equation (7.4) so that it will require the cumulative sum of squares of all the samples and the cumulative sum of the samples.

\[
\sigma_x^2 = \frac{1}{N} \left[ \sum_{i=1}^{N} x_i^2 - 2\bar{x} \sum_{i=1}^{N} x_i + \bar{x}^2 \right] \]  \hspace{1cm} (7.5)

\[
\sigma_x^2 = \frac{1}{N} \left[ \sum_{i=1}^{N} x_i^2 - 2\bar{x}^2 + \bar{x}^2 N \right] \]  \hspace{1cm} (7.6)

\[
\sigma_x^2 = \frac{1}{N} \left[ \sum_{i=1}^{N} x_i^2 \right] - 2\bar{x}^2 + \bar{x}^2 \frac{N}{N} \]  \hspace{1cm} (7.7)

\[
\sigma_x^2 = \frac{1}{N} \left[ \sum_{i=1}^{N} x_i^2 \right] - 2\bar{x}^2 + \bar{x}^2 \]  \hspace{1cm} (7.8)

\[
\sigma_x^2 = \frac{1}{N} \left[ \sum_{i=1}^{N} x_i^2 \right] - \bar{x}^2 \]  \hspace{1cm} (7.9)

\[
\sigma_x = \sqrt{\frac{1}{N} \sum_{i=1}^{N} x_i^2 - \left( \frac{\sum_{i=1}^{N} x_i}{N} \right)^2} \]  \hspace{1cm} (7.10)

2 We use equation (7.10) to calculate the standard deviation of error.

\[1\text{http://datagenetics.com/blog/november22017/index.html} \]

\[2\text{http://datagenetics.com/blog/november22017/index.html} \]
7.3 Comparison of prediction error with increase in radius

We predict the neighbor patches of an image from its central patch. We obtain the prediction error by gradually increasing the distance of the neighbor patches from the central patch. It is observed that the MSE increases and cosine similarity decreases gradually with the increase in the distance of the neighbor patches from the central patch. This result refers that the prediction of the closer neighbor patches from the central patches are better and the prediction degrades for the distance neighbor patches.
Figure 3. Euclidian distance between the original feature vector and the predicted feature vector for $k = 9$ and radius 1,2,4,7,10,13,16,20,25,30,35 and 40 pixels from the central patch: (a) one non-zero connection strength; (b) 3 non-zero connection strength; (c) 5 non-zero connection strength; (d) 7 non-zero connection strength; (e) 9 non-zero connection strength
Figure 4. Cosine similarity between the original feature vector and the predicted feature vector for $k = 9$ and radius 1, 2, 4, 7, 10, 13, 16, 20, 25, 30, 35 and 40 pixels from the central patch: (a) one non-zero connection strength; (b) 3 non-zero connection strength; (c) 5 non-zero connection strength; (d) 7 non-zero connection strength; (e) 9 non-zero connection strength
Figure 5. Euclidean distance between the original feature vector and the predicted feature vector for $k = 12$ and radius $1, 2, 4, 7, 10, 13, 16, 20, 25, 30, 35$ and $40$ pixels from the central patch: (a) one non-zero connection strength; (b) 3 non-zero connection strength; (c) 6 non-zero connection strength; (d) 9 non-zero connection strength; (e) 12 non-zero connection strength
Figure 6. Cosine similarity between the original feature vector and the predicted feature vector for $k = 12$ and radius $1, 2, 4, 7, 10, 13, 16, 20, 25, 30, 35$, and $40$ pixels from the central patch: (a) one non-zero connection strength; (b) 3 non-zero connection strength; (c) 6 non-zero connection strength; (d) 9 non-zero connection strength; (e) 12 non-zero connection strength
Figure 7. Euclidean distance between the original feature vector and the predicted feature vector for \( k = 36 \) and radius 1, 2, 4, 7, 10, 13, 16, 20, 25, 30, 35 and 40 pixels from the central patch in ??: (a) one non-zero connection strength; (b) 9 non-zero connection strength; (c) 18 non-zero connection strength; (d) 27 non-zero connection strength; (e) 36 non-zero connection strength.
Figure 8. Cosine similarity between the original feature vector and the predicted feature vector for $k = 36$ and radius 1,2,4,7,10,13,16,20,25,30,35 and 40 pixels from the central patch: (a) one non-zero connection strength; (b) 9 non-zero connection strength; (c) 18 non-zero connection strength; (d) 27 non-zero connection strength; (e) 36 non-zero connection strength.
7.4 Hypercolumn

A hypercolumn [2] of orientation column consists of a full set of orientation column over a cycle of $180^\circ$. Each column consists of vertical arrangements of neurons of same orientation.

![Figure 9](image)

Figure 9. A hypercolumn consisting of one set of orientation columns and one set of ocular dominance columns in the visual cortex. Electrode 1 penetrating at right angle to the surface encounters neurons responding to the same orientation and the same (right) ocular dominance. Electrode 2 penetrating parallel to the surface encounters neurons responding to the left ocular dominance and to different orientations[1]

We show this orientation hypercolumn structure computationally using the 2D feature arrangement matrix.

7.5 2D Feature Arrangement

We arrange the visual features (Gabor filters) with respect to the angular directions of neighbor patches and analyse the connection strength between them. This arrangement shows the connection strength between the visual features using gabor filter.

The 2D feature arrangement is done for winner-take-all. In winner-take-all, each row contains the connection strengths of that are most strongly connected features. The expected correlation matrices are normalized by dividing with the
maximum connection strength between the features of central patch respectively as

\[
\hat{M}_{ij} = \frac{M_{ij}}{\max M_{0j}} \quad (7.11)
\]

where \( i \in \{0, 1, ..., n\} \) and \( j \in \{1, ..., k\} \) is each row of \( M_i \), where \( n \) is the number of neighbors, and \( M_0 \) is the expected auto-correlation matrix of central patch and \( M_i \) is the expected correlation matrix of central patch and \( i^{th} \) neighbor patch where \( i \geq 0 \).

The connection strength is represented using gabor filters as

\[
m_r g(u, v)_r \quad (7.12)
\]

where \( m_r \in \max (\hat{M}_{ij}) \), \( g(u, v)_r \) is the gabor filter, \( r \in \{1, ..., k\} \) and \( k \) is the number of features.

### 7.6 Results and Analysis

The top row of the feature arrangement matrix in figure 10 represents the central patch and the rest of the rows represent the angular direction of each neighbor. Each column in the arrangement matrix represents the gabor filter orientation. The intensity of the gabor filter correlation strength or the connection strength between the features of central patch and the neighbor patch.
We have plotted the arrangement of features using gabor filter for 12 different radius. It can be observed from the figures above that most of the column have features of same orientation. The features that are strongly connected or brighter are the ones that are of same orientation and angular direction. The gabor features are more strongly connected for smaller radius and as the radius increases the connection strength between the features becomes weaker.
Figure 10. Node-feature arrangement for feature, $k = 9$ using winner-take-all approach: (a) radius =1; (b) radius =2; (c) radius =4; (d) radius =7; (e) radius =10; (f) radius =13. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.
Figure 11. Node-feature arrangement for feature, $k = 9$ using winner-take-all approach: (a) radius = 16; (b) radius = 20; (c) radius = 25; (d) radius = 30; (e) radius = 35; (f) radius = 40. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.
Figure 12. Node-feature arrangement for feature \(k = 12\) using winner-take-all approach: (a) radius = 1; (b) radius = 2; (c) radius = 4; (d) radius = 7; (e) radius = 10; (f) radius = 13. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.
Figure 13. Node-feature arrangement for feature $k = 12$ using winner-take-all approach: (a) radius = 16; (b) radius = 20; (c) radius = 25; (d) radius = 30; (e) radius = 35; (f) radius = 40. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.
Figure 14. Node-feature arrangement for feature \( k = 36 \) using winner-take-all approach: (a) \( \text{radius} = 1 \); (b) \( \text{radius} = 2 \); (c) \( \text{radius} = 4 \); (d) \( \text{radius} = 7 \); (e) \( \text{radius} = 10 \); (f) \( \text{radius} = 13 \). Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.
Figure 15. Node-feature arrangement for feature $k = 36$ using winner-take-all approach: (a) radius $= 16$; (b) radius $= 20$; (c) radius $= 25$; (d) radius $= 30$; (e) radius $= 35$; (f) radius $= 40$. Top row represents central patch; 2-19 row represents the neighbor patches; Each column represents a gabor filter of specific orientation; the intensity of the gabor filter represents the connection strength between the features.
7.7 Prediction Error with increase in sparsity

Figure 16. Prediction error vs sparsity for 9 features: (a) MSE for radius =1; (b) Cosine Similarity for radius =1

Figure 17. Prediction error vs sparsity for 12 features: (a) MSE for radius =1; (b) Cosine Similarity for radius =1

Figure 18. Prediction error vs sparsity for 36 features: (a) MSE for radius =1; (b) Cosine Similarity for radius =1
These results show that with increase in the degree of sparsity the mean square error increases monotonically and cosine similarity decreases monotonically.
Chapter 8

Conclusion

This thesis shows the representation of hypercolumn structure using simple linear model. A visual feature arrangement is developed which helps in accurate prediction. We also show that as the distance of predicted patch in an image increases, the prediction accuracy decreases. We show that the developed node feature arrangement resembles the orientation hypercolumn in visual cortex. We introduce different degree of sparsity in the learned model to analyse the change in prediction accuracy with the change in sparsity.

As a future work we want to apply this model in different types of data such as speech data and observe if there is analogous hypercolumn structure in other cortical areas.
REFERENCES


