

Motivic analysis of neuronal responses to visual stimuli

A Project Report

submitted by

ATHUL VIJAYAN

*in partial fulfilment of the requirements
for the award of the degree of*

MASTER OF TECHNOLOGY



**DEPARTMENT OF ENGINEERING DESIGN
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.**

April 2016

THESIS CERTIFICATE

This is to certify that the thesis entitled **Motivic analysis of neuronal responses to visual stimuli**, submitted by **Athul Vijayan**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Hema A. Murthy
Research Guide
Assistant Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

**I would like to thank everyone who helped me.

ABSTRACT

KEYWORDS: Markov Decision Processes, Symmetries, Abstraction

TABLE OF CONTENTS

| | |
|--|------|
| ACKNOWLEDGEMENTS | i |
| ABSTRACT | ii |
| LIST OF TABLES | v |
| LIST OF FIGURES | vi |
| ABBREVIATIONS | vii |
| NOTATION | viii |
| 1 Introduction | 1 |
| 2 Background and Previous work | 2 |
| 2.1 Visual pathway in brain | 2 |
| 2.2 Experiment setup | 2 |
| 2.2.1 Sinusoidal grating visual stimuli | 2 |
| 2.2.2 Natural videos visual stimuli | 2 |
| 2.3 Orientation and Directional selectivity of neurons in V1 | 2 |
| 3 Analyzing neuronal properties | 3 |
| 3.1 Quantifying Orientation and Directional selectivity | 3 |
| 3.2 Modeling neuronal response | 6 |
| 3.3 Finding similarly tuned neurons | 7 |
| 3.4 Study of principal components | 9 |
| 3.5 Study of Reliability | 11 |
| 4 Searching for Motifs | 13 |

| | | |
|----------|--|-----------|
| 4.1 | Cross-Correlation Function | 14 |
| 4.2 | ACFGram | 16 |
| 5 | Rough Longest Common Subsequence (RLCS) | 19 |
| 5.1 | Longest Common Subsequence (LCS) | 19 |
| 5.2 | RLCS | 20 |
| 5.2.1 | Motivic analysis of Neuronal signals with RLCS | 24 |
| 5.2.2 | Normalized RLCS score as a correlation metric | 25 |
| 6 | Inferences and Future work | 26 |
| A | Neural visual pathway | 27 |

LIST OF TABLES

| | |
|---|----|
| 5.1 Some examples of LCS in string comparison | 20 |
|---|----|

LIST OF FIGURES

| | |
|---|----|
| 3.1 Responses $R(\theta_k)$ of a simple neuron for each θ_k plotted in both orientation and direction space. Equal lobes in direction space shows direction is irrelevant. | 4 |
| 3.2 Responses $R(\theta_k)$ of a complex neuron for each θ_k plotted in both orientation and direction space. Unequal lobes in direction space shows one direction is preferred than its opposite. | 5 |
| 3.3 Responses $R(\theta_k)$ of a an orientation insensitive neuron for each θ_k plotted in both orientation and direction space. Similar responses to all orientations shows absence of selectivity. | 5 |
| 3.4 Fit of orientation and direction tuning curves of a neuron. The distinct peak in the orientation tuning curve shows selectivity to orientation θ_{pref} . Different σ_1 and σ_2 in direction tuning curve shows direction sensitive cell. - Thus a complex cell | 7 |
| 3.5 Fit of orientation and direction tuning curves of a neuron. The distinct peak in the orientation tuning curve shows selectivity to orientation θ_{pref} . Similar σ_1 and σ_2 in direction tuning curve direction of stimuli is irrelevant. - Thus a simple cell | 8 |
| 3.6 Pairwise Pearson correlation of neurons. Both axes are sorted in decreasing order of OSI. | 9 |
| 3.7 Orientation tuning curves of some neurons retrieved from correlation study. All of them have a same preferred orientation. | 10 |
| 3.8 Principal Component Analysis of response of neuron to drifting sinusoidal grating stimuli. | 11 |
| 3.9 Reliability measure R_A for different neurons | 12 |
| 4.1 Cross-Correlation function between a target neuron response and a manually chosen subsequence from template neuron. | 15 |
| 4.2 Extracting frame from template based on frame width and ending. | 16 |
| 4.3 ACFGram plots for template and target responses from same neuron. | 17 |
| 4.4 ACFGram plots for template and target responses from different neurons. | 18 |

ABBREVIATIONS

| | |
|-------------|----------------------------------|
| RLCS | Rough Longest Common Subsequence |
| LCSS | Longest Common Segment Set |

NOTATION

| | |
|--------------|---|
| $\rho(a, b)$ | Pearson correlation between a and b |
| V1 | Primary Visual Cortex |

CHAPTER 1

Introduction

CHAPTER 2

Background and Previous work

2.1 Visual pathway in brain

2.2 Experiment setup

2.2.1 Sinusoidal grating visual stimuli

2.2.2 Natural videos visual stimuli

2.3 Orientation and Directional selectivity of neurons in V1

CHAPTER 3

Analyzing neuronal properties

Characteristics of neurons in the V1 are discussed in the background section. In this chapter, we use experimental data to demonstrate the claimed properties of neurons in V1. In the experiment, a drifting sinusoidal grating video is shown to awake mice and neuronal responses were recorded. See Section 2.2.1 for detailed experiment setup.

Average response of a neuron is modeled as a function of orientation using a Gaussian function. Similarly, average response to various directions are modeled using a mixture of Gaussian functions. Root mean square of residuals are used as a goodness of fit measure.

Neurons in V1 are classified into simple, complex and unselective cells. Classifying a neuron into either of this class is useful as we can find the population of similar cells in the brain. Rather than thresholding OSI and DSI, We have used a k-means clustering algorithm with more features to classify cells.

3.1 Quantifying Orientation and Directional selectivity

Modern imaging technologies allow examining responses of neurons with clarity. Even though it was known that neurons in V1 are selective to orientation, we need robust metrics to quantify degree of selectivity and preferred orientation.

Plotting responses to each stimuli direction in a vector space provides an intuition about the characteristics. In orientation space, responses of two opposite directions are averaged. Angle varies from 0° to 180° in orientation space while angle angle changes from 0° to 360° in direction space. Figure** shows responses plotted in orientation and direction space for a simple neuron. Length of the vector sum is a good metric for amount of selectivity and preferred orientation. Normalized length of vector sum in orientation space is defined as OSI(Orientation Selectivity Index).

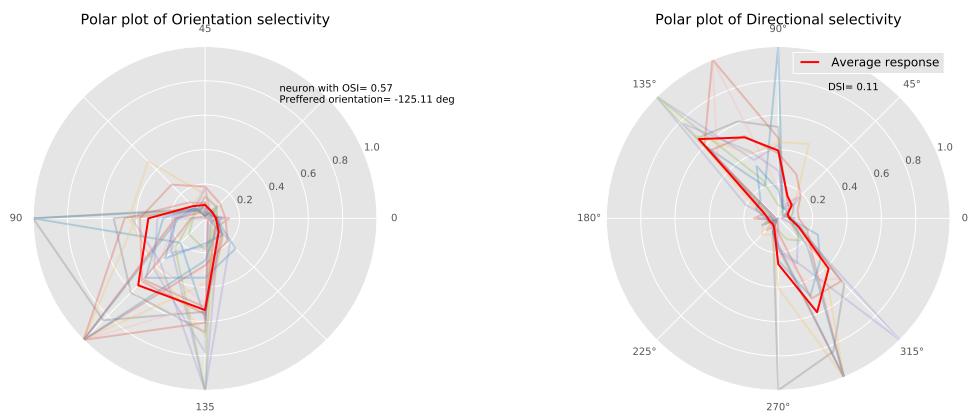
$$OSI = \left| \frac{\sum_k R(\theta_k) \exp(2i\theta_k)}{\sum_k R(\theta_k)} \right|$$

Similarly, Normalized length of vector sum in direction space is defined as DSI(Direction Selectivity Index).

$$DSI = \left| \frac{\sum_k R(\theta_k) \exp(i\theta_k)}{\sum_k R(\theta_k)} \right|$$

Where $R(\theta_k)$ is the average response to angle θ_k .

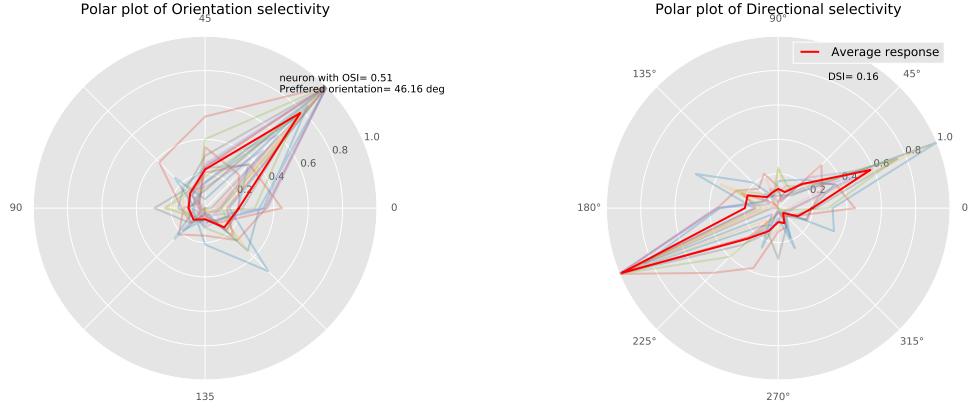
A simple neuron is expected to have high OSI but low DSI. Polar plots of responses in orientation and direction space is shown in Figure 4.4



(a) Responses plotted in orientation space (b) Responses plotted in direction space

Figure 3.1: Responses $R(\theta_k)$ of a simple neuron for each θ_k plotted in both orientation and direction space. Equal lobes in direction space shows direction is irrelevant.

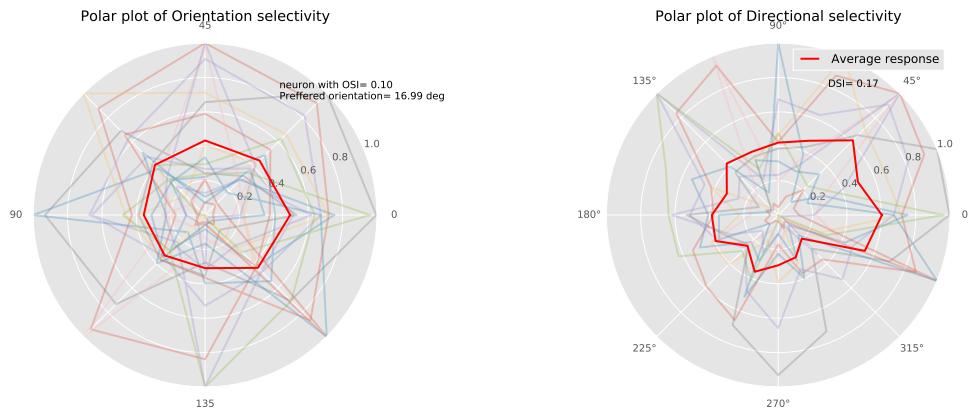
A complex neuron is expected to have high OSI and high DSI. Polar plots of responses in orientation and direction space is shown in Figure 3.2



(a) Responses plotted in orientation space (b) Responses plotted in direction space

Figure 3.2: Responses $R(\theta_k)$ of a complex neuron for each θ_k plotted in both orientation and direction space. Unequal lobes in direction space shows one direction is preferred than its opposite.

An orientation insensitive neuron is expected to have low values for both OSI and DSI. Polar plots of responses in orientation and direction space is shown in Figure 3.3.



(a) Responses plotted in orientation space (b) Responses plotted in direction space

Figure 3.3: Responses $R(\theta_k)$ of an orientation insensitive neuron for each θ_k plotted in both orientation and direction space. Similar responses to all orientations shows absence of selectivity.

3.2 Modeling neuronal response

Modeling the response of neuron to various orientations and visualizing is a great way to see if in fact there is an orientation selectivity. If the cell seems selective, we can also characterize the degree of selectivity and preferred orientation from model parameters.

Orientation tuning curve is modeled using a Gaussian function with constant offset. The empirical form of the orientation tuning curve is,

$$R_o(\theta) = C + R_p \exp \left\{ \frac{-\|\theta - \theta_{pref}\|^2}{2\sigma^2} \right\}$$

Where $R_o(\theta)$ is the time-averaged response of neuron to angle of orientation θ . Parameter θ_{pref} is the preferred orientation of the neuron. Tuning width σ tell us how much the cell is selective. C is a constant offset.

Similarly, we can model direction tuning curve using a mixture of Gaussian functions with a constant offset.

$$R_d(\theta) = C + R_p \exp \left\{ \frac{-\|\theta - \theta_{pref}\|^2}{2\sigma_1^2} \right\} + R_n \exp \left\{ \frac{-\|\theta - \theta_{null}\|^2}{2\sigma_2^2} \right\}$$

Where $R_o(\theta)$ is the time-averaged response of neuron to angle of direction θ . Relative magnitude of tuning widths, σ_1 and σ_2 denote the amount of directional selectivity. C is a constant offset.

Parameters are estimated by minimizing squared sum of error. Sum of squared error is defined as:

$$SSE = \sum_{i=1}^N \|R(\theta_i) - R_o(\theta_i)\|^2$$

A gradient descent algorithm finds the optimum parameters by minimizing SSE.

In Figure 3.4 , fit of tuning curves of a complex cell is given. The distinct peak in the orientation tuning curve shows selectivity to orientation θ_{pref} . In the direction tuning curve, peaks of different magnitude shows one direction is more preferred than other. In Figure 3.5 , fit of tuning curves of a complex cell is given. The

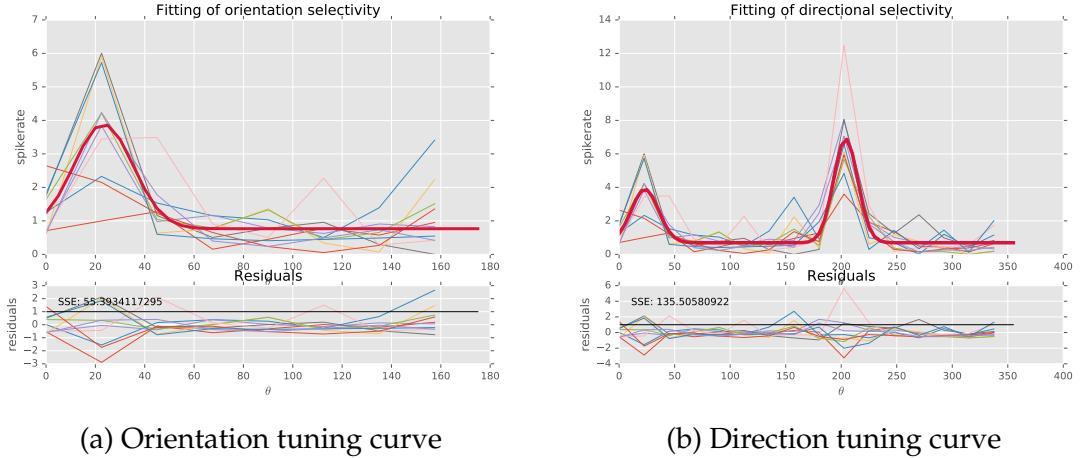
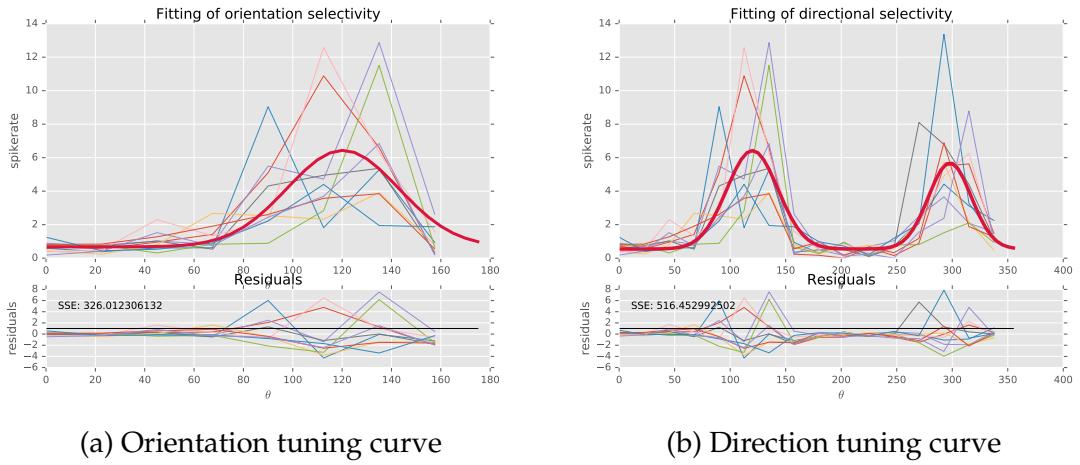


Figure 3.4: Fit of orientation and direction tuning curves of a neuron. The distinct peak in the orientation tuning curve shows selectivity to orientation θ_{pref} . Different σ_1 and σ_2 in direction tuning curve shows direction sensitive cell. - Thus a complex cell

distinct peak in the orientation tuning curve shows selectivity to orientation θ_{pref} . In the direction tuning curve, peaks of same magnitude shows of stimuli direction is irrelevant.

3.3 Finding similarly tuned neurons

Receptive field of a neuron in V1 consists of a subset of neurons in one layer below it. Those neurons in turn have a receptive field. By following layers down, we can find a visual field for each neuron in V1. Orientation selectivity of neurons in V1 detects edges and their orientation in images. By studying similarly tuned neurons, we can get an insight to redundancy of coding and distribution of



(a) Orientation tuning curve

(b) Direction tuning curve

Figure 3.5: Fit of orientation and direction tuning curves of a neuron. The distinct peak in the orientation tuning curve shows selectivity to orientation θ_{pref} . Similar σ_1 and σ_2 in direction tuning curve direction of stimuli is irrelevant. - Thus a simple cell

functionally similar cells in V1.

Similarly tuned cells are expected to have similar response to different stimuli orientations. A high correlation of responses $R(\theta_k)$ to various angle θ_k will represent similarly tuned cells. Even after having same preferred orientations, the neurons could have different degree of selectivity. We would like to find similarly tuned neurons which have an ‘acceptable’ OSI.

Pearson correlation between each pair of neurons in a mice are computed and plotted in Figure 3.6. The neurons in both axes have a decreasing OSI value. The neuron pairs that lie in bottom left of the Figure 3.6 are the ones we have interested. Finally to find neurons that are similarly tuned to a particular neuron, choose the corresponding row and all the neurons in that row having good correlation value and having acceptable OSI are selected. Figure 3.7 shows tuning curves of some neurons retrieved from correlation study.

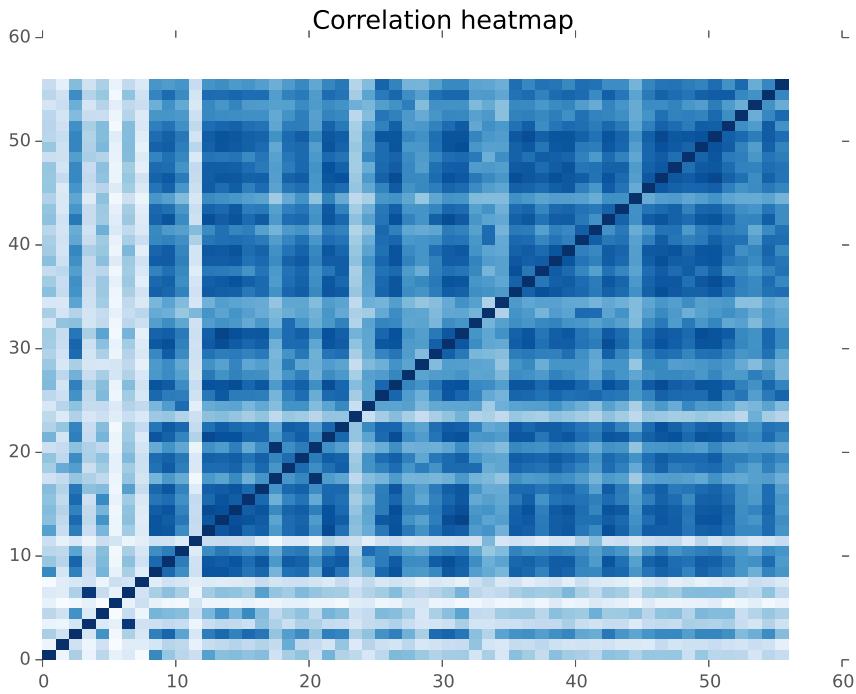


Figure 3.6: Pairwise Pearson correlation of neurons. Both axes are sorted in decreasing order of OSI.

3.4 Study of principal components

Information contained in the visual stimuli reaches V1 after it gets passed through lower layers. Edges and motion of edges in the stimuli are represented by orientation and direction sensitive neurons in V1. Through Principal Component Analysis (PCA) we aim to study the tightness of information representation. A subset of neurons (~ 60) in V1 are sampled at 20Hz for 6 seconds during the experiment. We analyze principal components of the responses to find redundancy in the responses. The number of uncorrelated principal components which capture ‘most’ of the variance in data can determine redundancy.

Principal Component Analysis is an orthogonal transformation of possibly correlated observations to linearly uncorrelated components called principal compo-

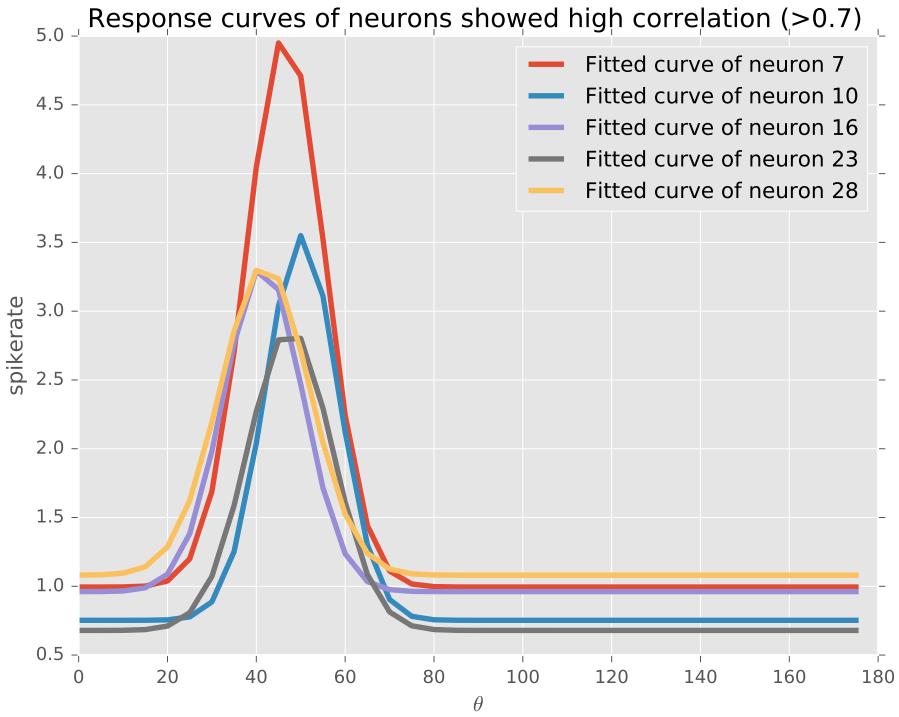


Figure 3.7: Orientation tuning curves of some neurons retrieved from correlation study. All of them have a same preferred orientation.

nents. The principal components are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric. The first principal component captures largest variance and decreases further. By observing number of components that needs to capture a desired variance, we can tell correlations in the data. Transforming original data to new orthogonal basis gives a representation which has dimension less than or equal to the number of original dimension. Reconstructing original data from subset of principal components also indicates amount of correlation in original data.

Here we take number of neurons as feature dimension. The responses are averages across trials. PCA is done on the data to find ratio between number of principal components and variance explained. Figure 3.8a shows the result for a mouse towards a sinusoidal grating stimuli. The original data was transformed

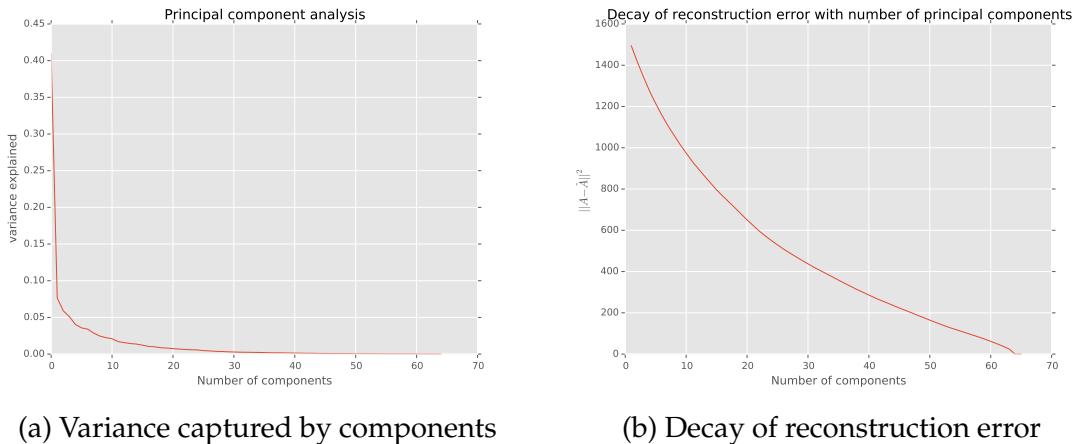


Figure 3.8: Principal Component Analysis of response of neuron to drifting sinusoidal grating stimuli.

to principal components basis with a subset of principal components. Attempting to reconstructing original data from transformed data produces an error. The reconstruction error depends on number of chosen subset of principal components. Figure shows reconstruction error for different number of principal components. As we increase number of principal components, the error decreases. And finally as the number components equals the original feature dimension, reconstruction error vanish.

3.5 Study of Reliability

Information encoding in primary visual cortex is a complex process due to intrinsic neuronal variability. Intertrial reliability of information encoding in neurons is the first step into analyzing coding mechanism of V1. The degree of trial-to-trial variability in a response is commonly measured in terms of reliability. A neuron is said to be reliable if it fires the same number of precisely timed spikes on every repetition of a stimulus [Tiesinga *et al.* [2008]].

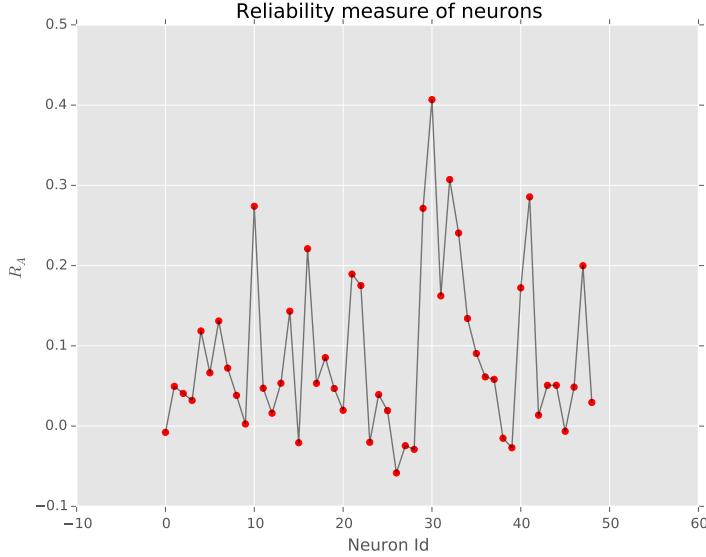


Figure 3.9: Reliability measure R_A for different neurons

The experiment performed measures calcium concentration in neurons rather than spike rate. Trial to trial reliability can be computed using correlation between responses of each trial. When a visual stimuli is presented T times, we can compute response reliability R_A of a neuron to the stimuli A .

$$R_A = \frac{2}{T^2 - T} \sum_{i=1}^T \sum_{j=i+1}^T \rho(f_{i,A}, f_{j,A})$$

where $f_{i,A}$ is the response of neuron to i^{th} trial of movie A and ρ is the Pearson correlation.

The study was done for 48 neurons in a mice and the response reliability is plotted in Figure 3.9 for each neuron. It is evident that response reliability for most of neurons are small. The study shows responses of neurons in V1 are not robust because of intrinsic noise. If the whole sequences is not reliable, could there be subsequences that are reliable? We will explore it in coming chapters by detecting common subsequences.

CHAPTER 4

Searching for Motifs

In Music, motif is a perceivable recurring fragment. They are elementary signatures that repeats to form bigger units. Detecting such motifs in a song tells us more about the song like its melody. In Genetics, motif is a sequence pattern of nucleotides in a longer DNA sequence. From a signal processing perceptive, motifs is a signal segment that recurs in a longer signal. A valid motif should have an ‘acceptable’ length such that it has a significance. Motif analysis is import because in every domain, occurrence of motifs has a meaning to it.

In the experiment, each neuron responses are captured as a time series sampled at 20Hz for 10 seconds. In neuron responses we define a Motif as subsequence of response time series which has the following properties:

- Recurs in the same response signal but in a different part **or**
- Recurs in the response of the same neuron to a different trial **or**
- Recurs in the response of another neuron in the same mouse **or**
- Recurs in the response of neuron in a different mouse **and**
- Has a significant length in time.

Analyzing motifs in neuronal signals will help us understand reliable information representation. Even if two responses of same trial has a long subsequence but not time synchronized, Study of reliability in Section 3.5 will fail as the Pearson correlation coefficient will output a small value. Motivic analysis can detect such subsequences even if they are time shifted.

Study of motifs across two different neurons within a mouse can explain correlation between two neurons. Correlated neurons can represent neuronal interconnections. Correlated and synchronous activity in populations of neurons has been observed in many brain regions and has been shown to play a crucial role in cortical coding, attention, and network dynamics [Rosenbaum *et al.* [2014]]. Studying correlations between neurons is again not effective if their responses are time separated as Pearson correlation will fail. Analyzing motifs across neurons will be a better way to study neuronal correlations.

Motifs found in neurons from two *different* mice will represent similar biological process both the neurons undergoes. As the motifs found cannot be due to interconnections, they are due to same activation mechanism of neurons. In this case, we would expect a small motif compared to motifs found within mouse.

In this chapter, we will analyze the presence of motifs in neuronal signals. The presence of motifs will motivate us to extract the motifs and their significance.

4.1 Cross-Correlation Function

Cross correlation is a measure of similarity between two signals as a function of lag of one signal relative to the other. It is commonly used for searching a short query sequence in a large reference sequence. The function calculates sliding dot product at various lags. For discrete signals, cross correlation is defined as:

$$(f \star g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m] g[m+n].$$

Searching for a motif using correlation function is possible if we already have a subsequence that we suspect as a motif. To visualize correlation function in query search, we will manually select a subsequence and search for its occurrences in a neuronal response signal. For the scope of this chapter, we take a subsequence from the response of a ‘template neuron’ and search for its occurrences in response of a ‘target neuron’.

Trial averaged responses of template and target neurons are taken, then a subset of former is extracted using parameters frame ending and frame width. This subsequence is compared with target sequence as a function of lag. Figure 4.1 shows Cross-Correlation function between a target neuron response and a manually chosen subsequence from template neuron. This study is not effective as we

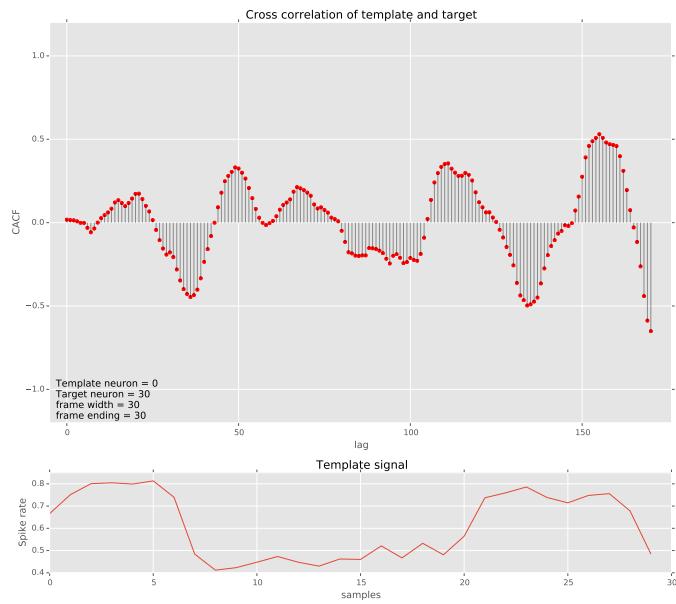


Figure 4.1: Cross-Correlation function between a target neuron response and a manually chosen subsequence from template neuron.

do not know what to search for. Manually selecting a subsequence is inefficient and selected subsequence does not guarantee to be valid motif.

4.2 ACFGram

A spectrum describes a signal in terms of energy spread over its frequency components. A spectrogram does exactly the same but also takes another time into consideration. Computing spectrogram is done by first making chunks/frames of time-domain signal which usually overlap. Magnitude of Fourier transform of each frame is computed to form magnitude spectrum of that frame. These frame spectra are stacked horizontally in increasing order of frame ending time to form a spectrogram. This enables us to study the change of spectra with time.

We formulate an analogous visualization of Cross-Correlation function where the template signal changes in each frame. A frame is a subsequence of original signal with a width and frame ending parameter. Changing the frame ending successively will return overlapping frames having same width. Figure 4.2 shows how a frame from template signal is selected. Cross-Correlation of extracted

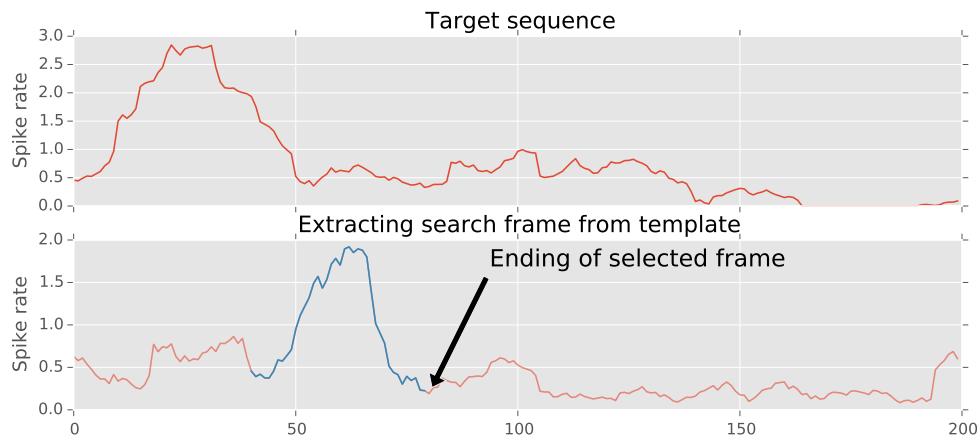


Figure 4.2: Extracting frame from template based on frame width and ending.

frame and target signal at various lags are computed. Next frame is extracted by incrementing frame ending parameter. The process is then repeated for the next frames. Cross-Correlation function of each frame is then stacked horizontally to

form a temporal dimension. We call the resulting Time Vs Lag Vs Cross-Correlation function as ACFGram. In the previous study In Section 4.1 we varied frame width and frame ending manually to extract a signal subsequence and then queried it in target signal. ACFGram removes frame ending parameter. Existence of motifs will be clear if we iterate frame width. The study was conducted across trials of same neuron and across different neurons.

The Figure 4.3b shows ACFGram plots for template and target responses from same neuron. The study aim to find presence of repeating subsequence within a neuron's response. The length of suspected motif is varied as a parameter. For a small frame width (~ 15) the estimation of Cross-Correlation function is poor. Also such a small subsequence is not significant enough to be considered as a motif. As we Increase the frame width, repeating patterns are found which indicates the presence of repeating subsequence. The Figure 4.4b shows ACFGram plots for

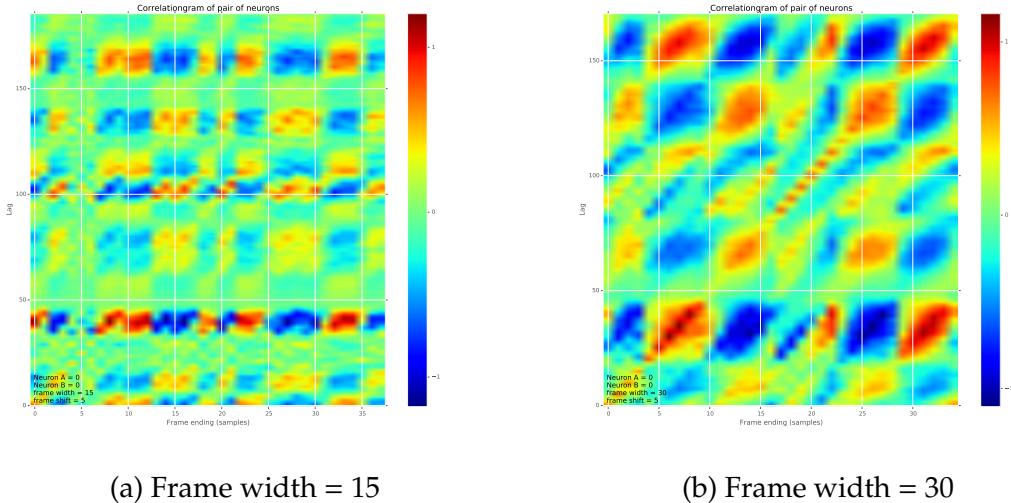


Figure 4.3: ACFGram plots for template and target responses from same neuron.

two different neuron responses. Motifs across neurons are studied for finding a better correlation metric. For different frame widths, responses indicate presence of motifs across neurons. The approximate length of motifs is more than 2 seconds,

which is significant. Importantly, more or less any pair of neurons taken from the sampled set of 65 neurons from each mouse exhibited the presence of motifs. This indicates the inter neuronal correlation. This supports the claim that information in V1 is coded in population by inter-neuronal correlations. The search for the

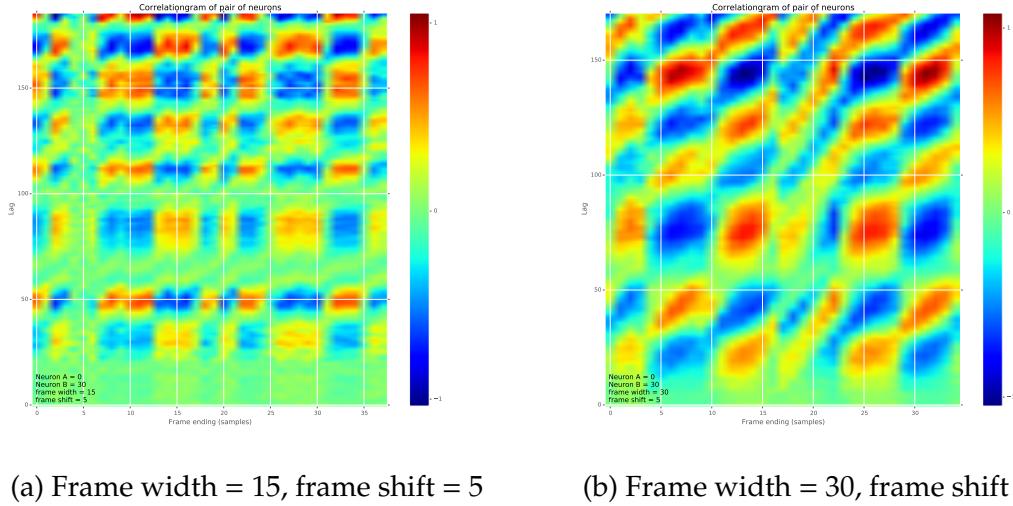


Figure 4.4: ACFGram plots for template and target responses from different neurons.

presence of motifs in primary visual cortex succeeded in with experimentally verifying rough common subsequences of significance (> 2 seconds) exists in neuronal responses. This motivates us to find the longest motifs you can find between two neurons. Also till now we used Pearson correlation as a measure of similarity. We look into rough measures of similarity and longest common subsequence in the next chapter.

CHAPTER 5

Rough Longest Common Subsequence (RLCS)

Having established existence of motifs in neuronal responses, A robust algorithm to extract best possible motifs from two signals is required. As a longer motif has more significance in the context, an algorithm which produce longest common subsequence is desirable. Longest Common Subsequence (LCS) is a classical problem in query string matching, data comparison and Bio-informatics. We will eventually look into an adapted LCS problem which can be used to match signals called Rough Longest Common Subsequence (RLCS).

5.1 Longest Common Subsequence (LCS)

A subsequence of a string S is a set of characters that appear from left to right in original string S but not necessarily consecutive. There will be more than one subsequence for any string with length greater than unity. For two strings S_1 and S_2 , the common entries in sets of substrings of each strings are called common subsequences of S_1 and S_2 . Among the common subsequences, the subsequence with maximal length is called Longest Common Subsequence.

Some examples of LCS are given in Table 5.1. Note that there can be more than one LCS for two strings.

LCS problem can be defined formally for two subsequences $X = (x_1, x_2 \dots x_m)$

| String 1 - S_1 | String 2 - S_2 | $LCS(S_1, S_2)$ |
|------------------|------------------|-----------------|
| ABCDGH | AEDFHR | ADH |
| AGGTAB | GXTXAYB | GTAB |
| AGCAT | GAC | AC, GC and GA |
| BACDB | BDCB | BCB |

Table 5.1: Some examples of LCS in string comparison

and $Y = (y_1, y_2 \dots y_n)$ as

$$LCS(X_i, Y_j) = \begin{cases} & \text{if } i = 0 \text{ or } j = 0 \\ LCS(X_{i-1}, Y_{j-1}) \cup x_i & \text{if } x_i = y_j \\ \text{longest}(LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j)) & \text{if } x_i \neq y_j \end{cases}$$

Where $X_i = (x_1, x_2 \dots x_i)$ and $Y_j = (y_1, y_2 \dots y_j)$.

LCS problem has an optimal substructure- the problem can be broken down into subproblems of similar kind. As the problem also has overlapping subproblems, a Dynamic Programming approach is used to find the LCS. Running time of the dynamic programming approach is $O(nm)$ where n and m are the lengths of strings.

5.2 RLCS

LCS problem can be adapted for signal matching. Instead of equality, a rough distance measure is used. If the distance between samples is less than a threshold, they are considered a match. It is not desirable just to optimize the length of subsequence in signals as the gaps between selected samples can cause false alarms. Dynamic Programming optimization has to be penalizing gaps and motivating matches. The rest of the problem and the solution remains similar to LCS problem and similar Dynamic Programming approach can solve the problem efficiently.

Modified LCS problem with rough comparison is stated below. The RLCS of two input sequences $X = \langle x_1, x_2, \dots, x_m \rangle$ and $Y = \langle y_1, y_2, \dots, y_n \rangle$ is obtained by maximizing the score value S .

$$S(X_i, Y_j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ S(X_{i-1}, Y_{j-1}) + x_i & \text{if } dist(x_i, y_j) < \tau_{dist} \\ \max(S(X_i, Y_{j-1}), S(X_{i-1}, Y_j)) & \text{if } dist(x_i, y_j) > \tau_{dist} \end{cases}$$

Where $X_i = \langle x_1, x_2, \dots, x_i \rangle$ and $Y_j = \langle y_1, y_2, \dots, y_j \rangle$.

Various context uses different distance measures for comparing samples. Manhattan distance (city block distance) of notes is used in context of music matching [Lin *et al.* [2011]]. In a work on verification of Raga in Carnatic music [Dutta *et al.*], a domain specific similarity measure is used. RLCS on neuronal signals in this work uses normalized Euclidean distance for checking similarity.

$$dist(x_i, y_i) = (x_i - y_i)^2$$

Distances are normalized from 0 to 1 by performing:

$$dist(x_i, y_i) = \frac{dist(x_i, y_i) - minDist}{maxDist - minDist}$$

Where $minDist$ is the minimum pairwise distance and $maxDist$ is the maximum pairwise distance.

Most important part of RLCS is the adapted score update rules. In the classical LCS problem, the gaps between the selected subsequence was not considered; Instead only the length of the subsequence was of consideration. But in signals,

having a gap in between will give poor signal match. So it is important to penalize the gaps and reward matches. The following score update rules penalizes gaps in subsequence and awards positive score for matches.

1. Score update for a match. $dist(x_i, y_j) < \tau_{dist}$

$$score(i, j) = score(i - 1, j - 1) + 1 - \frac{dist(x_i, y_j)}{\tau_{dist}}$$

This case happens when a match is found. The score should be rewarded with a positive value. The maximum awarded score is 1 for an exact sample match and minimum score given is 0 for when the distance equals threshold. The closer the samples, more the score.

2. Score update for a mismatch. $dist(x_i, y_j) > \tau_{dist}$

$$score(i, j) = score(i - 1, j - 1) - \delta$$

We allowed any number of gaps in a string LCS problem. But in the case of signals, we cannot allow large gaps as it would produce false alarms. We penalize each mismatching samples with a constant score δ .

The problem is then recursively solved using Dynamic Programming algorithm which will provide us with a 2-D score matrix. The Dynamic programming algorithm for input sequences $X = < x_0, x_1, \dots, x_{m-1} >$ and $Y = < y_0, y_1, \dots, y_{n-1} >$ is explained in

Backtracking through score matrix will gives us the Longest Common Subsequence. As we have been penalizing gaps, there will be points in the track where score is zero. For each sample match we had increased score and decreased for gaps. So the zero will mean that there are enough mismatches to balance the positive score created by matches. This lets us identify the boundary of a subsequence. A number of subsequences could be extracted by cutting at points where score goes zero. This set of subsequences are called **Longest Common Subsequence Set** (LCSS) [Dutta *et al.*]. Algorithm ** explains backtracking to extract the LCSS.

Algorithm 1 Dynamic Programming algorithm for RLCS ***

```
1: function RLCS( $X, Y, \tau_{dist}, \delta$ )
2:    $m \leftarrow$  length of  $X$ 
3:    $n \leftarrow$  length of  $Y$ 
4:    $p \leftarrow \min(m, n)$ 
5:   for  $i = 0; i < m; i++$  do
6:     for  $j = 0; j < n; j++$  do
7:        $dist(i, j) \leftarrow euclideanDistance(x_i, y_j)$ 
8:        $\text{minDist} \leftarrow \min(dist)$                                  $\triangleright$  For normalization
9:        $\text{maxDist} \leftarrow \max(dist)$                                  $\triangleright$  For normalization
10:
11:       $\text{cost} \leftarrow zeros(m + 2, n + 2)$                                  $\triangleright$  For storing running score
12:       $\text{score} \leftarrow zeros(m + 2, n + 2)$                                  $\triangleright$  For storing score
13:       $\text{diag} \leftarrow zeros(m + 2, n + 2)$                                  $\triangleright$  For backtracking
14:       $\text{partial} \leftarrow zeros(m + 2, n + 2)$                                  $\triangleright$  For storing partial scores.
15:
16:      for  $i = 0; i < m; i++$  do
17:        for  $j = 0; j < n; j++$  do
18:           $d \leftarrow (dist(i, j) - \text{minDist}) / (\text{maxDist} - \text{minDist})$        $\triangleright$  Normalize
19:          if ( $d < \tau_{dist}$ ) then
20:             $\text{diag}(i, j) \leftarrow ' \nearrow '$                                  $\triangleright$  Path to travel while backtracking
21:             $\text{cost}(i, j) \leftarrow \text{cost}(i - 1, j - 1) + (1 - d/\tau_{dist})$ 
22:             $\text{score}(i, j) = \text{score}(i - 1, j - 1)$ 
23:          else if  $\text{cost}(i - 1, j) > \text{cost}(i, j - 1)$  then
24:             $\text{diag}(i, j) \leftarrow ' \uparrow '$ 
25:             $\text{cost}(i, j) = \text{cost}(i - 1, j) - \delta$                                  $\triangleright$  Penalize
26:          if  $i > stringlen$  then return false
27:           $j \leftarrow patlen$ 
28:        loop:
29:          if  $string(i) = path(j)$  then
30:             $j \leftarrow j - 1.$ 
31:             $i \leftarrow i - 1.$ 
32:            goto loop.
33:            close;
34:           $i \leftarrow i + \max(delta_1(string(i)), delta_2(j)).$ 
35:        goto top.
```

Algorithm 2 Dynamic Programming algorithm for RLCS ***

```
1: function RLCS( $X, Y, \tau_{dist}, \delta$ )
2:    $m \leftarrow$  length of  $X$ 
3:    $n \leftarrow$  length of  $Y$ 
4:    $p \leftarrow \min(m, n)$ 
5:   for  $i = 0; i < m; i++$  do
6:     for  $j = 0; j < n; j++$  do
7:        $dist(i, j) \leftarrow euclideanDistance(x_i, y_j)$ 
8:        $\text{minDist} \leftarrow \min(dist)$                                  $\triangleright$  For normalization
9:        $\text{maxDist} \leftarrow \max(dist)$                                  $\triangleright$  For normalization
10:
11:       $\text{cost} \leftarrow zeros(m + 2, n + 2)$                            $\triangleright$  For storing running score
12:       $\text{score} \leftarrow zeros(m + 2, n + 2)$                            $\triangleright$  For storing score
13:       $\text{diag} \leftarrow zeros(m + 2, n + 2)$                            $\triangleright$  For backtracking
14:       $\text{partial} \leftarrow zeros(m + 2, n + 2)$                           $\triangleright$  For storing partial scores.
15:
16:      for  $i = 0; i < m; i++$  do
17:        for  $j = 0; j < n; j++$  do
18:           $d \leftarrow (dist(i, j) - \text{minDist}) / (\text{maxDist} - \text{minDist})$      $\triangleright$  Normalize
```

5.2.1 Motivic analysis of Neuronal signals with RLCS

Having established presence of motifs in neuronal responses, we use Rough Longest Common Subsequence (RLCS) algorithm to find the ‘best’ motifs from a signal processing point of view. Penalizing the gaps makes RLCS a good signal matching algorithm.

The experiment was conducted on responses of neurons from V1 of 11 mice to different visual stimuli. Each signal is sampled at 20 Hz and last for 10 seconds. The study is split into different sections based on which neurons the signals come from.

Motifs in responses of a neuron to different trials

The signals from a neuron’s response to different trials of same stimuli are passed to RLCS algorithm to detect motifs.

Motifs in responses of two neurons in a mice

Motifs in responses of neurons across different mice

5.2.2 Normalized RLCS score as a correlation metric

CHAPTER 6

Inferences and Future work

APPENDIX A

Neural visual pathway

REFERENCES

- Dutta, S., K. S. PV, and H. A. Murthy ()**. Raga verification in carnatic music using longest common segment set.
- Lin, H.-J., H.-H. Wu, and C.-W. Wang** (2011). Music matching based on rough longest common subsequence. *J. Inf. Sci. Eng.*, **27**(1), 95–110.
- Rosenbaum, R., T. Tchumatchenko, and R. Moreno-Bote**, *Correlated neuronal activity and its relationship to coding, dynamics and network architecture*:. Frontiers Research Topics. Frontiers E-books, 2014. ISBN 9782889193578. URL <https://books.google.co.in/books?id=AQNdBgAAQBAJ>.
- Tiesinga, P., J.-M. Fellous, and T. J. Sejnowski** (2008). Regulation of spike timing in visual cortical circuits. *Nature reviews neuroscience*, **9**(2), 97–107.