



**Manifold Structure of High-Dimensional Data in
Artificial and Biological Neural Networks**

ZHANG Liu

**Capstone Final Report for BSc (Honours) in
Mathematical, Computational and Statistical Sciences**

Supervised by: Francesca Spagnuolo

Co-supervised by: Steven W. Zucker

AY 2021/2022

Yale-NUS College Capstone Project

DECLARATION & CONSENT

1. I declare that the product of this Project, the Thesis, is the end result of my own work and that due acknowledgement has been given in the bibliography and references to ALL sources be they printed, electronic, or personal, in accordance with the academic regulations of Yale-NUS College.
2. I acknowledge that the Thesis is subject to the policies relating to Yale-NUS College Intellectual Property ([Yale-NUS HR 039](#)).

ACCESS LEVEL

3. I agree, in consultation with my supervisor(s), that the Thesis be given the access level specified below: [check one only]

☐ **Unrestricted access**

Make the Thesis immediately available for worldwide access.

☐ **Access restricted to Yale-NUS College for a limited period**

Make the Thesis immediately available for Yale-NUS College access only from _____ (mm/yyyy) to _____ (mm/yyyy), up to a maximum of 2 years for the following reason(s): (please specify; attach a separate sheet if necessary):

_____.

After this period, the Thesis will be made available for worldwide access.

☐ **Other restrictions: (please specify if any part of your thesis should be restricted)**

Zhang Liu, Cendana College

Name & Residential College of Student



Signature of Student

October 15 2021

Date

Francesca Spagnuolo

Name & Signature of Supervisor

Date

Acknowledgements

I would like to thank my supervisor Prof. Francesca Spagnuolo and co-supervisor and mentor Prof. Steven W. Zucker and Dr. Luciano Dybala for their generous guidance and advice. They have made possible the many serendipitous moments in this project. There have been ups and downs in this project, but their support has helped me keep a keen spirit for learning and discovery.

I am also grateful for my family and friends who have been patiently bearing with my endless sharing of my work and ideas.

YALE-NUS COLLEGE

Abstract

B.Sc (Hons)

**Manifold Structure of High-Dimensional Data in
Artificial and Biological Neural Networks**

by ZHANG Liu

Key words: Manifold Learning, recurrent neural networks, computational vision, harmonic analysis

This work builds on a long line of research aiming to develop more accurate mathematical and computational models of the visual system. It has recently been shown that feed-forward neural networks turn out to be inaccurate models for the neurobiological networks in the visual cortex. We focused on a related question that has not been investigated: how is the structure of recurrent and transformer neural networks related to that of neurobiological networks (in the mouse visual cortex)?

We first build the biological and artificial neuron tensors using experimental neural spiking data and numerical simulations on recurrent and transformer neural networks. Using tensor component analysis, we discover which groups of neurons respond similarly to which stimuli input. Since these groups are likely not independent, we use the non-linear dimensionality reduction method, diffusion maps, to infer a manifold of neurons. This manifold structure implies a functional network (represented by the discrete data graph underlying the continuous manifold) and thus reflects both the neural circuit connections and the neuron's role in those circuits. Comparing the manifold structures of biological and artificial neural networks allows us to make precise inferences about similarities and differences in their respective functional circuits.

Contents

Acknowledgements	ii
Abstract	iv
1 Introduction	1
1.1 Aim and significance	1
1.2 Historical context	3
1.2.1 Hierarchy Models	3
1.2.2 Alternative Models	6
Parallel Models	6
Recurrent Models	7
1.3 Related works on geometric and topological approach to neural networks	8
1.4 Organization of the report	10
2 Overview of Methodology	11
2.1 Main objective	11
2.2 What is a neural manifold?	11
2.3 How to generate the neural manifold?	13
2.4 What can we analyze from the neural manifold?	14
3 Preliminaries on Tensor CP decomposition	16

3.1	PCA	16
3.2	Higher-order generalization of PCA: tensor CP decomposition	18
3.3	Demonstrate tensor CP decomposition by using the face image dataset	20
3.4	Remarks on linear dimensionality reduction	21
4	Preliminaries on Diffusion Map	22
4.1	Demonstration of the method	26
4.2	Remarks on non-linear dimensionality reduction	28
5	Experiments with Biological Neural Tensors	29
5.1	Data collection	29
5.2	Applying tensor CP decomposition	31
5.3	Applying diffusion map	33
6	Experiments with Artificial Neural Tensors	35
6.1	Building the artificial neural tensor	35
6.1.1	“Visual stimuli” in ANNs	36
6.1.2	“Individual neuron” in ANNs	37
6.1.3	“Receptive field” of an artificial neuron	39
6.1.4	“Neuron population response” in ANNs	42
6.2	Applying tensor CP decomposition	42
7	Topological Methods	44
7.1	Topological data analysis	44
7.2	Step 1: Applying persistent homology	44
7.3	Step 2: Pairwise Wasserstein distance	48

7.4	Analysis of results	51
7.5	Discussions	53
8	Conclusion and Next Steps	54
8.1	Summary of work done	54
8.2	Next steps	56
	Bibliography	58
A	Persistent Homology Theory	66
A.1	Motivation	66
A.2	Homotopy	67
A.3	Simplicial Complexes	69
A.4	Persistent homology	73

Chapter 1

Introduction

1.1 Aim and significance

This capstone project builds on a long line of research aiming to develop more accurate mathematical and computational models of the visual system. There has recently been evidence suggesting that feed-forward neural networks such as Convolutional Neural Networks (CNNs) turn out to be inaccurate models for the primary visual cortex (V1), even though CNN was initially inspired by the earliest computational model of the primary visual cortex (the Neocognitron model) and has performed well on most computer vision tasks.

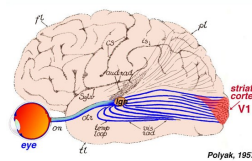


FIGURE 1.1: Visual input goes from the eye to primary visual cortex (V1). Adapted from Polyak (1957).

The central question we will be investigating in this project is the following: how is the structure of artificial neural networks (ANNs) related

to that of neurobiological networks in the primary visual cortex? This remains an open question.

One challenge of comparing the neural network structures is that the neural responses (from both V1 and ANNs) are usually encoded in high-dimensional representation. Our approach to this question is to infer a neural manifold using dimensionality reduction methods on the neural responses. Informally, the neural manifold shows the clusters of neurons grouped by their firing patterns in response to a given visual stimulus. This manifold structure implies a functional network (represented by the discrete data graph underlying the continuous manifold) and thus reflects both the neural circuit connections and the neuron's role in those circuits.

We then compare the neural manifolds in biological and artificial neural networks. For the latter, we will be investigating Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and the latest Transformer and Perceiver networks. By comparing the manifold structures of biological and artificial neural networks we can make precise inferences about similarities and differences in their respective functional circuits.

The significance of achieving the goals outlined above is twofold. First, there is significant interest in comparing the structure of artificial and biological neural networks. Artificial neural networks have proved capable of many computer vision tasks at a level competitive to biological systems. However, whether artificial and biological neural networks have the same functional circuit remains an open question. (Gwilliams and King, 2017)

Second, modeling the visual system is an important task. In the field of neuroscience, despite decades of research, we have not fully understood the science behind visual perception. In the field of AI, more sophisticated models of the visual system could inspire new computational algorithms in solving increasingly demanding computer vision tasks. In fact, research in computer vision has taken many inspirations from neuroscience. In LeCun, Bengio, & Hinton's seminal work on deep learning (LeCun, Bengio, and Hinton, 2015), they stated that "ConvNets have their roots in the Neocognitron," which was one of the earliest computational models of the visual system.

1.2 Historical context

The earliest efforts in modeling the visual system began with Hubel and Wiesel's hierarchy model in the 1960s. Since then, numerous alternative models were proposed, such as the parallel models and recurrent models, all of which aimed to more accurately reflect the actual neural connectivity in the visual system.

1.2.1 Hierarchy Models

Hubel and Wiesel's hierarchy model (Hubel and Wiesel, 1962) was the first seminal work on modeling the visual system, which classified the neurons in the primary visual cortex (V1) into simple cells and complex cells. The essential difference between the simple cells and complex cells is that the responses of simple cells are modulated by the spatial phase of a sine grating, whereas the responses of complex cells are largely phase

invariant. In other words, as we progress from simple cells to complex cells, the neurons become selective for increasingly complex stimuli and at the same time become more tolerant to the exact position within their receptive fields.

Based on this, a natural way to construct complex cells is to group responses from simple cells with the same orientation preference, but with different phase preferences.

This idea directly inspired the Neocognitron model (Fukushima, 1980). In the Neocognitron model, simple cells (termed as “S-cells” in the original paper) are tuned to simple stimuli at the convolution layers. Their outputs are then combined at the pooling layers by taking the maximum or average to form complex cells (termed as “C-cells” in the original paper). As a result, complex cells are tuned to more complex stimuli.

The Neocognitron model was among a myriad of hierarchical models of the visual system. It went on to inspire the Convolutional Neural Networks (CNNs), a class of artificial neural networks that has become dominant in various computer vision tasks. (Yamashita et al., 2018)

The figure below shows a systematic illustration of the general idea behind the hierarchical models:

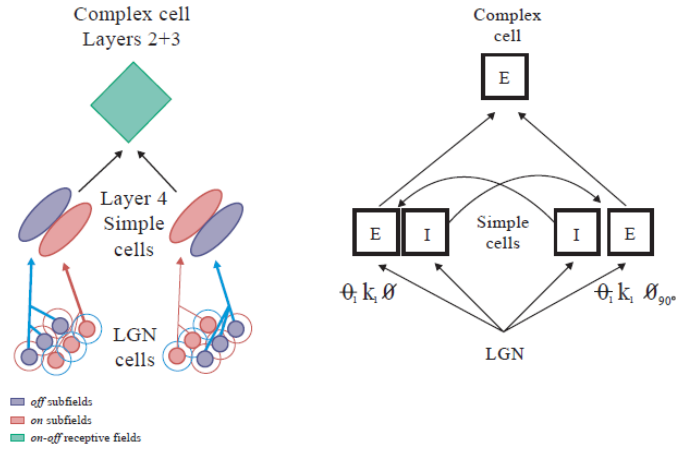


FIGURE 1.2: Illustration of the hierarchical models (Martinez and Alonso, 2003)

The possible advantages of hierarchical models are as follows:

- If a visual recognition task can be decomposed into low-complexity learning tasks for each layer of a hierarchical learning machine, then each layer may require only a small number of training examples (Poggio, Smale, et al., 2003).
- The lowest levels of the hierarchy may represent a dictionary of features that can be shared across multiple classification tasks (Geman, 1999), thus increasing efficiency.

There are also some known limitations of hierarchical models:

- Hierarchical models assume that the computations at each successive stage being largely feed-forward (Riesenhuber and Poggio, 1999, DiCarlo, Zoccolan, and Rust, 2012). This is limited because back-projections are also likely to be a key part of the visual system.
- There remains a very broad distribution of tuning and receptive field sizes in all areas of the visual hierarchy. Hence, the anatomical

hierarchy should be taken as an idealization and cannot be taken as a strict flowchart of visual information (Hegde and Felleman, 2007).

1.2.2 Alternative Models

Since Hubel and Wiesel proposed the classification of simple cells and complex cells, many other hierarchical models have been proposed for a more realistic representation for the cortical circuitry. Furthermore, new experimental and computational evidence provided serious alternatives to this hierarchical model, including parallel models and recurrent models (Martinez and Alonso, 2003). There are still many controversies and debates over which models best capture neural mechanisms of the visual system.

Parallel Models

The first strong evidence against the hierarchical model was the discovery that some complex cells, like simple cells, receive monosynaptic input from the thalamus (Hoffmann, Stone, and Sherman, 1972). Based on this discovery, Hoffman and Stone proposed that both cell types, simple and complex, were generated in parallel by separate thalamocortical pathways, as shown in diagram A in Figure 1.3.

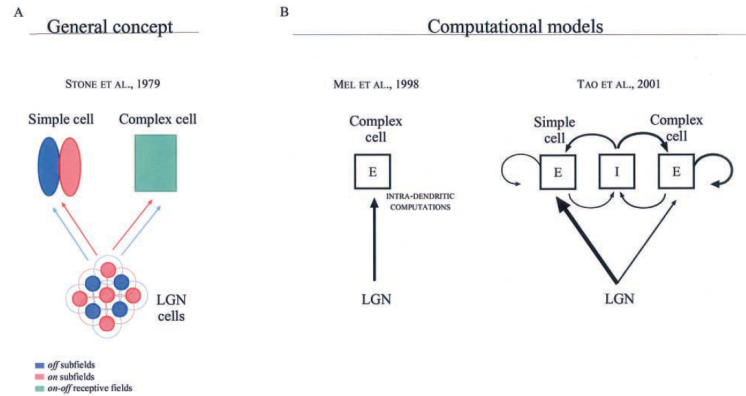


FIGURE 1.3: Illustration of the parallel models (Martinez and Alonso, 2003)

Simple cells and complex cells are far from being two parallel cortical pathways in the same way that X and Y cells are parallel thalamic pathways. However, the idea that some complex receptive fields can be generated at least in part by direct thalamic inputs is likely to be correct. (Martinez and Alonso, 2003)

Recurrent Models

Recurrent models changed the focus of attention from single cells to networks of cortical connections. (Martinez and Alonso, 2003) An illustration for recurrent models are shown in Figure 1.4.

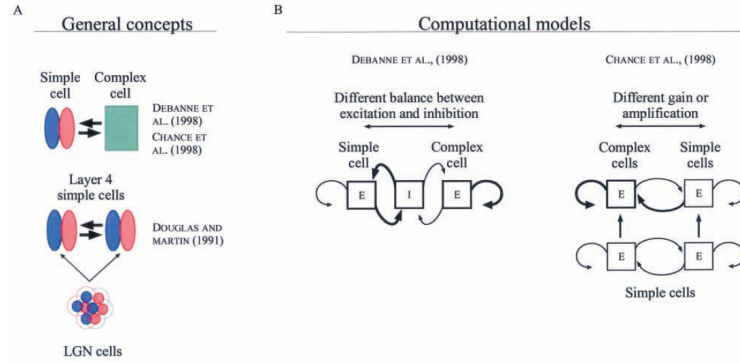


FIGURE 1.4: Illustration of the recurrent models (Martinez and Alonso, 2003)

1.3 Related works on geometric and topological approach to neural networks

Throughout the historical progress in neuroscience, there have been numerous efforts to apply geometric and topological methods from related fields in mathematics and computer science, including computational geometry, differential geometry, spectral graph theory, and algebraic topology, among many others. Recently, studying the structure of neural population response with high-dimensional neural spiking data has become a commonly used technique in studying the structure of brain cortices. The high-dimensional nature of neural spiking data have thus motivated various dimensionality reduction methods, both linear and non-linear.

The paper (Chung and Abbott, 2021) provided a comprehensive review of the recent studies that have applied geometric and topological methods to discover the properties of neural population geometry.

The authors of (Gallego et al., 2017) were the first to propose the concept of “neural modes,” which are specific patterns of correlated neural

activity that span the neural manifolds. They thus proposed a generative model of individual neuronal activity based on the activation of neural modes. The parameters of such model can be obtained using dimensionality reduction methods. Williams et al., 2018 used linear dimensionality reduction method of tensor canonical polyadic (CP) decomposition to discover low-dimensional neural dynamics by extracting three “tensor factors,” which are interconnected, low-dimensional descriptions of neural data.

The first work that applied topological methods in neuroscience is (Singh et al., 2008). The authors applied computational topology to analyze neural population activity in the primary visual cortex and concluded that the topological structure of neural population activity is align with the topological structure of a 2-sphere both when the cortex is spontaneously active and when evoked by natural images.

Following that, there have been many applications of persistent homology in studying the topological structure of neural population activity across different systems. A notable application is (Chaudhuri et al., 2019), where the authors applied persistent homology to study the the neural population activity from the post-subiculum and anterodorsal thalamus. They showed that the head direction of mice can be decoded from a one-dimensional ring structure. A recent addition to this line of work was (Beshkov and Tiesinga, 2021). The authors of this paper showed that when using the geodesic distance instead of the Euclidean distance, persistent homology method was able to successfully identify highly non-linear features. They then provided an application using the Neuropixels dataset recorded in mouse visual cortex after stimulation

with drifting gratings.

1.4 Organization of the report

The current chapter outlined the goals and significance of this project, historical background of the research question, and related works. Chapter 2 will give an overview of the methodology and Chapters 3 and 4 will address the different parts of the methodology in greater details. Chapter 5 and 6 will present the implementation details on the experiments for biological and artificial neural networks, respectively. Chapter 7 is an extension of the method to study the topological features of the manifold structure. Chapter 8 will summarize the tasks and results completed during this semester and provide a blueprint for the next steps.

Chapter 2

Overview of Methodology

2.1 Main objective

The main objective of our project is to study the manifold structure of biological and artificial neural networks. This objective is equivalent to generating the underlying neural manifolds from data that are obtained from biological and artificial neural networks respectively. In what follows, we will give an overview of the methodology according to the following questions:

- What is a neural manifold?
- How to generate the neural manifold?
- What can we analyze from the neural manifold?

2.2 What is a neural manifold?

Before explaining the idea behind “neural manifold” in the context of neuroscience, we first introduce the general concept of a manifold as a mathematical object.

Definition 1 (Homeomorphism) A continuous map $f : X \rightarrow Y$ be a function between topological spaces X and Y . If f is bijective, then the inverse f^{-1} exists. If both f and f^{-1} are continuous, then f is a homeomorphism. The two topological spaces, X and Y , are said to be homeomorphic if there exists a homeomorphism between X and Y .

Definition 2 (Manifold) In brief, a real n -dimensional manifold is a topological space M for which every point $x \in M$ has a neighborhood homeomorphic to Euclidean space \mathbb{R}^n .

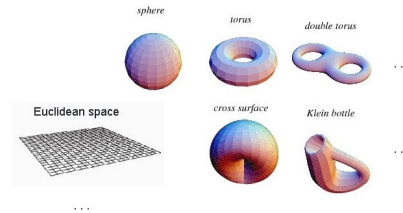


FIGURE 2.1: Examples of 2-dimensional manifolds.

Our approach to study the neural population response via the neural manifold is based on the premise of “the Manifold Hypothesis,” which states that real-world high-dimensional data lie on a low-dimensional manifold embedded within the high-dimensional space. (DeepAI, 2019) The original representation of data has a large degree of freedom, that is, the number of variables that are really necessary to describe the data is much smaller than the number of variables in the original representation.

In the context of neuroscience, neural spiking data are usually high-dimensional due to the kind of data representation (the peristimulus, or PSTH, diagrams) determined by the lab instrument. However, it has been shown that the neural connections constrain the possible patterns of neural population activity (Okun et al., 2015, Sadtler et al., 2014, Tsodyks,

1999) and that the possible patterns are confined to a low-dimensional manifold (Stopfer, Jayaraman, and Laurent, 2003, Yu et al., 2009) spanned by a few independent patterns that are called “neural modes.” (Gallego et al., 2017)

We call this underlying low-dimensional manifold the “**neural manifold**.” The distance between a pair of points on the neural manifold indicate how similar their firing patterns are in response to a given visual stimulus. Intuitively we can understand the neural manifold as a collection of clusters of neurons grouped by their firing patterns in response to a given visual stimulus. Figure 2.2 shows an example of a neural manifold generated from neural spiking data.

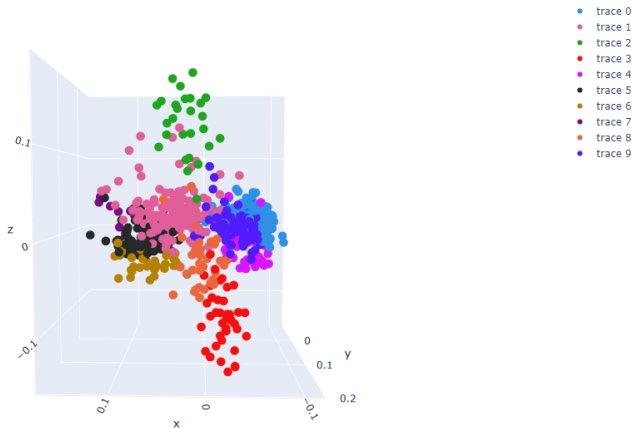


FIGURE 2.2: Visualize an example of a neural manifold generated from neural spiking data.

2.3 How to generate the neural manifold?

The task of generating the neural manifold is equivalent to learning¹ the low-dimensional manifold underlying the original high-dimensional

¹“Learning” here refers to unsupervised learning, which is a class of approaches in machine learning that discover meaningful features without training data.

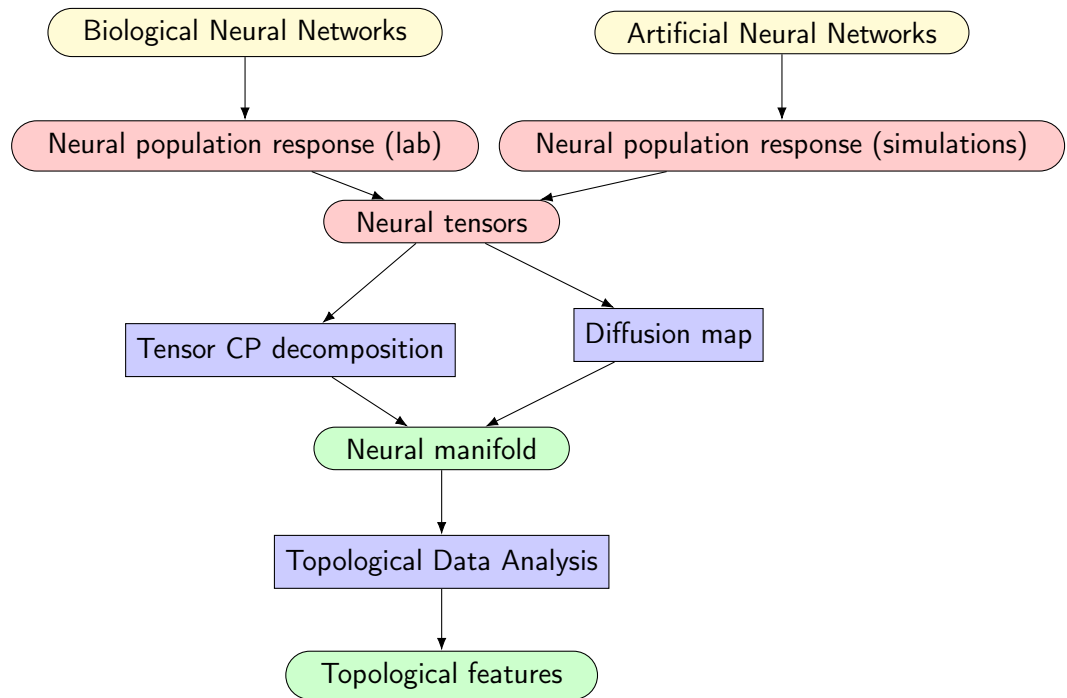
neural spiking data, which amounts to a dimensionality reduction task. In our project, we use both the linear dimensionality reduction method tensor canonical polyadic (CP) decomposition and the non-linear dimensionality reduction method diffusion map. The theoretical preliminaries for these two methods will be explained in full details in the next two chapters.

2.4 What can we analyze from the neural manifold?

We can directly compare the low-dimensional neural manifold for the biological neural networks with that for the artificial neural networks. The neural manifolds implies a functional network. The functional network is represented by the discrete data graph underlying the continuous manifold and thus reflects both the neural circuit connections and the neuron's role in the circuit. We can thus make precise inferences about the similarities and differences between the particular biological and artificial neural networks in terms of their respective functional circuit, which is what we set out to compare.

We can further extend this comparison to the *topological* properties of the neural manifolds by extracting the topological features via persistent homology and quantifying the topological similarities using appropriate distance metric between the topological features. In our study, we have chosen the commonly used p -Wasserstein distance to quantify the pairwise similarities between the topological features of the neural manifolds given different stimuli.

The following flowchart showed a schematic summary of the methodology of the project.



Chapter 3

Preliminaries on Tensor CP decomposition

This chapter explains in detail the linear dimensionality reduction method used in this project, tensor canonical polyadic (CP) decomposition, a higher-order generalization of principal component analysis (PCA).

3.1 PCA

Principal component analysis (PCA) is the simplest matrix decomposition model. The PCA model can be formulated as an optimization problem. Given a N -by- M matrix \mathbf{X} with N variables and M features, we can approximate \mathbf{X} with the product of two orthogonal rank-one matrices:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T = \sum_{r=1}^R u_r \circ v_r, \quad (3.1)$$

where \circ denotes the outer product operator. The equivalent element-wise formulation of the PCA model is

$$x_{ij} \approx \sum_{r=1}^R u_i^r v_j^r. \quad (3.2)$$

With PCA, the dimensionality of the original matrix can be reduced from N -by- M to N -by- R . This can be illustrated with the diagram below

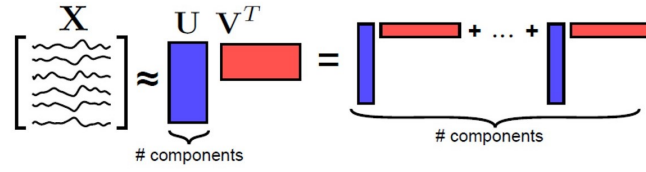


FIGURE 3.1: Illustration for PCA. (Adapted from (Williams et al., 2018))

- maximize variance:

$$\begin{aligned} & \underset{\mathbf{V}}{\text{maximize}} \quad \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 \\ & \text{subject to} \quad \mathbf{V} \text{ orthonormal} \end{aligned} \quad (3.3)$$

- minimize residuals:

$$\begin{aligned} & \underset{\mathbf{U}, \mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \\ & \text{subject to} \quad \mathbf{U}, \mathbf{V} \text{ orthogonal} \end{aligned} \quad (3.4)$$

Remark 3 Note that without the constraint of \mathbf{U}, \mathbf{V} being orthogonal, PCA has infinite number of solutions since $\mathbf{U}\mathbf{V}^T = \mathbf{U}\mathbf{F}^{-1}\mathbf{F}\mathbf{V}^T = \mathbf{U}'\mathbf{V}'^T$.

When the data has a non-negative constraint, non-negative matrix factorization (NMF) is used. In NMF, the second part of the optimization

becomes the following instead:

$$\begin{aligned} & \underset{\mathbf{U}, \mathbf{V}}{\text{minimize}} \quad \|\mathbf{X} - \mathbf{UV}^T\|_F^2 \\ & \text{subject to} \quad \mathbf{U} \geq 0, \mathbf{V} \geq 0. \end{aligned} \tag{3.5}$$

3.2 Higher-order generalization of PCA: tensor CP decomposition

The main references for this section are (Williams et al., 2018), (Kolda and Bader, 2009), and (Hong, Kolda, and Duersch, 2020).

Tensor canonical polyadic (CP) decomposition generalizes PCA from matrices to higher-order tensors.

Definition 4 (Tensor norm) *The norm of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is*

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2}. \tag{3.6}$$

Definition 5 (Rank-one tensors) *An N -way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is rank-one if it can be written as the outer product of N vectors, that is,*

$$\mathcal{X} = \vec{a}^{(1)} \circ \vec{a}^{(2)} \circ \dots \circ \vec{a}^{(N)} \tag{3.7}$$

Analogous to PCA, tensor CP decomposition aims to approximate the original tensor by a sum of rank-one tensors, each of which can then be written as the outer product of N vectors. Given a 3-way tensor $\mathcal{X} \in$

$\mathbb{R}^{I \times J \times K}$, the tensor CP decomposition model is formulated as

$$\mathcal{X} \approx [[\mathbf{A}, \mathbf{B}, \mathbf{C}]] = \sum_{r=1}^R a_r \circ b_r \circ c_r, \quad (3.8)$$

where R is the number of components and $a_r \in \mathbb{R}^I, b_r \in \mathbb{R}^J, c_r \in \mathbb{R}^K$ for $r = 1, \dots, R$. The element-wise formulation analogous to 3.2 is

$$x_{ijk} \approx \sum_{r=1}^R a_{ir} b_{jr} c_{kr} \quad (3.9)$$

for $i = 1, \dots, I, j = 1, \dots, J, k = 1, \dots, K$.

With tensor CP decomposition, we obtain the tensor factors which are analogous to the principle components. This can be illustrated with the diagram below

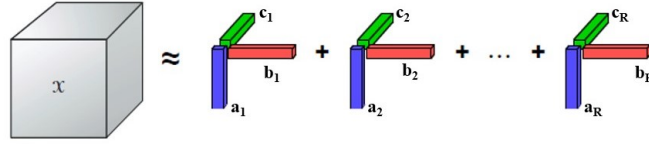


FIGURE 3.2: Illustration for tensor CP decomposition.
(Adapted from (Williams et al., 2018))

The alternating least squares (ALS) method is one of the most common algorithms to compute a tensor CP decomposition with R components. The steps of the ALS are summarized as follows:

1. fix \mathbf{B} and \mathbf{C} to solve for \mathbf{A} :

$$\min_{\mathbf{A}} \sum_{ijk} \left(x_{ijk} - \sum_l a_{il} b_{jl} c_{kl} \right)^2 \quad (3.10)$$

2. fix \mathbf{A} and \mathbf{C} to solve for \mathbf{B} :

$$\min_{\mathbf{B}} \sum_{ijk} \left(x_{ijk} - \sum_l a_{il} b_{jl} c_{kl} \right)^2 \quad (3.11)$$

3. fix \mathbf{A} and \mathbf{B} to solve for \mathbf{C} :

$$\min_{\mathbf{C}} \sum_{ijk} \left(x_{ijk} - \sum_l a_{il} b_{jl} c_{kl} \right)^2 \quad (3.12)$$

and repeat the above steps until some convergence criterion is satisfied.

3.3 Demonstrate tensor CP decomposition by using the face image dataset

As an intuitive demonstration for the tensor CP decomposition method, we apply tensor CP decomposition on face image data with 1000 face images, with each image having dimension 96-by-96. Thus, the face image data are encoded in a 3-way tensor of dimension 1000-by-32-by-32. Using tensor CP decomposition, we obtain the first 5 tensor components, which intuitively represent the prominent facial features in the face images:

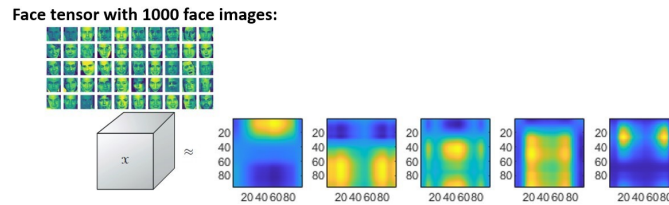


FIGURE 3.3: First 5 tensor factors for face image data.

By projecting the data onto the first few principal components and apply the k -means clustering method, we obtain the following clusters grouped by similar face images:

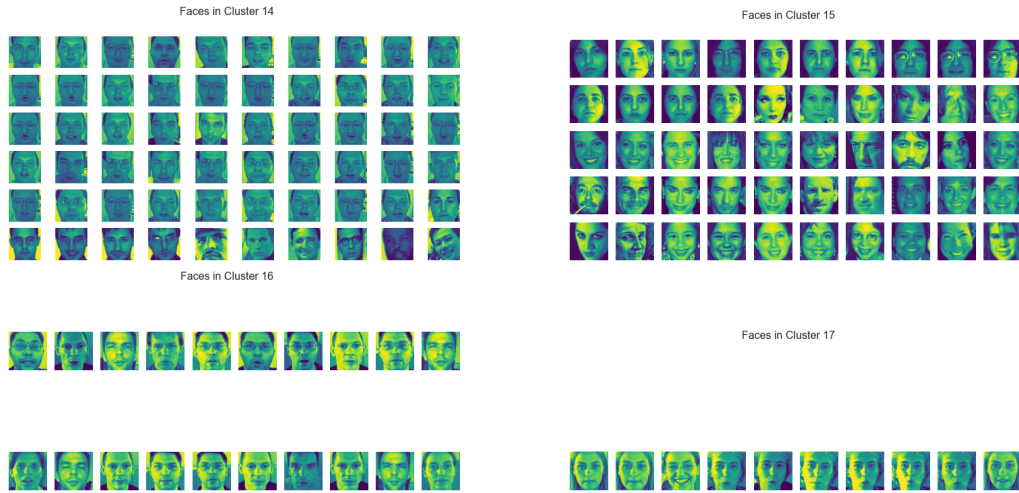


FIGURE 3.4: Some arbitrary clusters showing the results of tensor CP decomposition for face data.

3.4 Remarks on linear dimensionality reduction

Linear dimensionality reduction techniques work well when the data lie near a linear subspace of high-dimensional space. They do not work well when the data lie near a non-linear manifold embedded in the high-dimensional space. (Fefferman, Mitter, and Narayanan, 2016) Due to this reason, we introduce the non-linear dimensionality reduction technique of diffusion map in the next chapter.

Chapter 4

Preliminaries on Diffusion Map

In this chapter we introduce the non-linear dimensionality reduction method of diffusion map. The main references for this section are (Coifman et al., 2005), (Coifman and Lafon, 2006) and (Stanley, 2020). We will now outline the steps in the diffusion map algorithm.

1. Discretize the underlying manifold as a weighted graph.

Definition 6 (Discretization of a manifold) *Given $X = \{x_1, x_2, \dots, x_n\}$, a set of data sampled from a Riemannian manifold \mathcal{M} , we can construct the discretization of \mathcal{M} as a weighted, undirected graph, $\mathcal{G}_X = (X, g_\varepsilon)$ where X is the set of vertices given by the data and g_ε is a Gaussian kernel function $g_\varepsilon : (x_i, x_j) \rightarrow [0, \infty]$ defined by the following rule*

$$g_\varepsilon(x_i, x_j) = \frac{1}{2} \left(\exp \frac{-\|x_i - x_j\|_2^2}{2\varepsilon(x_i)} + \exp \frac{-\|x_i - x_j\|_2^2}{2\varepsilon(x_j)} \right), \quad \forall x_i, x_j \in X, \quad (4.1)$$

where $\varepsilon : X \rightarrow \mathbb{R}$ is the bandwidth function by which the Gaussian kernel is parameterized.

Gaussian kernel is only one of the possible kernel functions, but it gives a physically intuitive construction for learning the manifold underlying the data sample because it effectively considers a ball of radius ε around each data point.

The choice of ε determines the radius of neighborhood of $x_i \in X$. ε is application-specific and is usually a global bandwidth, that is, $\varepsilon(x_i) = \varepsilon(x_j)$ for all $x_i, x_j \in X$.

2. Compute the similarity matrix associated with the weighted graph.

Given the weighted graph \mathcal{G}_X , we can compute the corresponding similarity matrix. Each entry in the similarity matrix, W_{ij} , is the pairwise similarity value between data/vertices x_i and x_j and is taken as the weight of the edge between nodes x_i and x_j .

$$W_{ij} = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}. \quad (4.2)$$

3. Construct a lazy random walk on \mathcal{G}_X .

First, we define random walk on the weighted, undirected graph \mathcal{G}_X :

Definition 7 (Random walk on graphs) A random walk on $\mathcal{G}_X = (X, g_\varepsilon)$ is a process that begins at some vertex x_i , and at each time step moves to another vertex x_j . Since the graph is weighted, it moves to a neighbor x_j with probability $p(j \mid i)$, proportional to the weight of the corresponding edge, and is defined as follows:

$$p(j \mid i) = \frac{W_{ij}}{\sum_k W_{ik}}. \quad (4.3)$$

The random walk is lazy if we allow $p(i | i) > 0$, i.e., the probability of staying at some point i as the time step moves forward is non-zero.

The probabilities $p(j | i)$ can be represented by a Markov matrix M , which is essentially the similarity matrix normalized to have row sums equal to 1:

$$M = D^{-1}W, \text{ where } D_{ii} = \sum_k W_{ik}. \quad (4.4)$$

The probability of reaching node x_j from x_i after t steps is then

$$p(t, j | i) = e_i^T M^t e_j. \quad (4.5)$$

4. Eigendecomposition of the Markov matrix:

The eigendecomposition of M is derived from the eigendecomposition of $M_s = D^{1/2}MD^{-1/2} = \Omega\Lambda\Omega^T$:

$$M = D^{-1/2}\Omega\Lambda\Omega^T D^{-1/2} := \Psi\Lambda\Phi^T. \quad (4.6)$$

Note that since Ψ and Φ are mutually orthogonal, Ψ contains the right eigenvectors, as shown below:

$$M\Psi = \Psi\Lambda\Phi^T\Psi = \Psi\Lambda = \Lambda\Psi. \quad (4.7)$$

The eigendecomposition of M after t steps is then

$$M^t = \Psi\Lambda^t\Phi^T. \quad (4.8)$$

- Diffusion coordinate functions are the right eigenvectors of Markov matrix scaled by their corresponding eigenvalues:

$$\Upsilon := \Psi \Lambda. \quad (4.9)$$

- Diffusion distance after t steps is the following:

$$\|e_i^T \Upsilon - e_j^T \Upsilon\|^2 = \sum_k (p(t, k | i) - p(t, k | j))^2 (D_{kk}^{-1}). \quad (4.10)$$

- Infer geometric properties from the growth of eigenvalues of M .

For intuition behind the relation between the growth of eigenvalues and the geometric properties of the underlying manifold that we begin with, we consider the following two extreme situations:

- (a) If the discretization of the underlying manifold is a disconnected graph (none of the nodes are connected), then:

$$P = I, \lambda_i = \lambda_j \quad \forall i, j,$$

which implies a flat spectrum with zero decay rate.

- (b) If the discretization of the underlying manifold is a fully connected graph (each of the node is connected to all the rest of the nodes), assuming weights of all edges are 1, then:

$$\lambda_1 = 1, \lambda_i = 0 \quad \forall i \neq 1.$$

Key ideas of diffusion maps:

- The similarity kernel gives us the *local* geometry. As the time steps move forward, we integrate the local geometry and thus reveal the geometric structures at different scales.
- A cluster from a random walk is a region where the probability of escaping this region is low.

4.1 Demonstration of the method

As a demonstration of the diffusion maps method, we apply it on synthetic spiral data and MNIST handwritten digits images data:

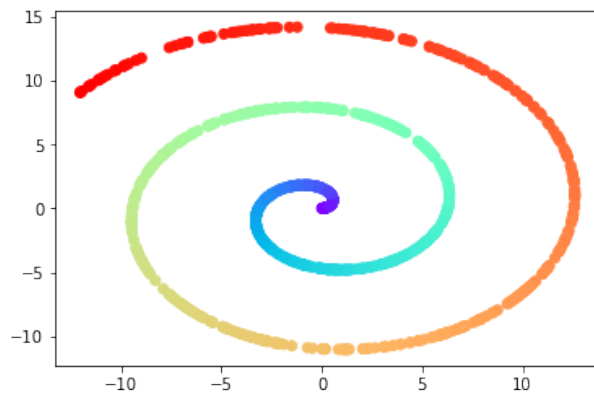


FIGURE 4.1: Visualising the original spiral data.

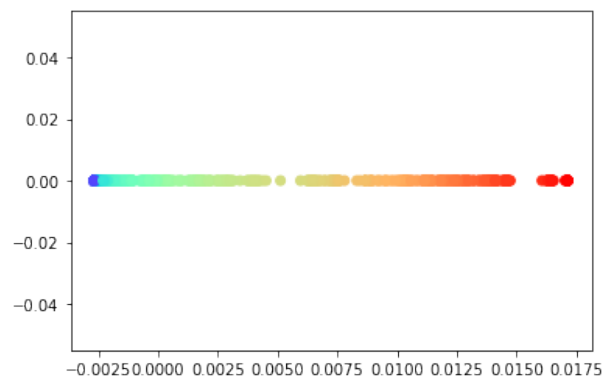


FIGURE 4.2: First non-trivial coordinate function.



FIGURE 4.3: Sample data from the MNIST.

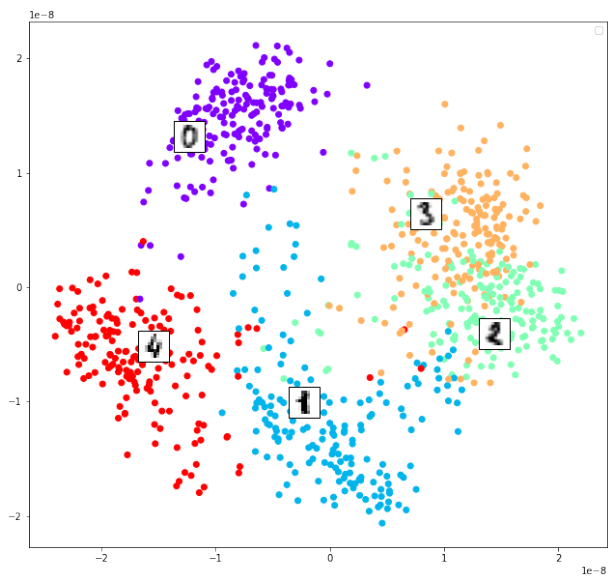


FIGURE 4.4: First two non-trivial coordinate functions.

4.2 Remarks on non-linear dimensionality reduction

Setting the right parameters for non-linear dimensionality reduction is very challenging, especially for lab data where sampling errors are prevalent. For this reason, we will be investigating this method in greater details in the second half of the project instead.

Chapter 5

Experiments with Biological Neural Tensors

5.1 Data collection

The neural spiking data (Dyballa, 2021) were collected from lab experiments conducted on mice. The experimental setup is shown in the figure below. Visual stimuli of moving artificial gratings of six different types were flashed in front of the mouse. Each visual stimuli were moving in eight directions. Response of neurons in the mouse retina was recorded with electrodes and encoded in peristimulus (PSTH) diagrams. Each PSTH diagram shows the firing rate of one neuron over time for each of the eight directions respectively. Brighter pixels indicate higher firing rates.

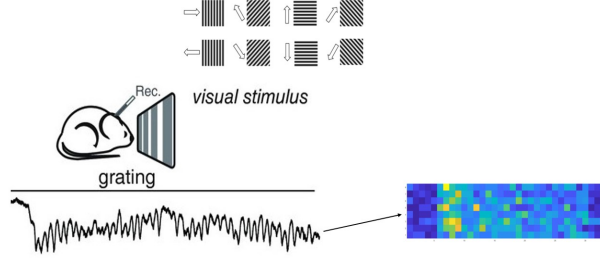


FIGURE 5.1: Visualising neural data from lab experiments.

The neural spiking data have 3 dimensions: the first dimension represents the 698 neurons, the second dimension represents the six different types of stimuli, and the third dimension represents the vectorized PSTH diagrams, each of which has $8 \times 33 = 264$ pixels. Due to the number of dimensions, we need to generalize the familiar notion of matrix (which only has two dimensions) to its higher-order analogue, tensors, which we define as follows:

Definition 8 (Tensors) *An N -way tensor is an element of the tensor product of N vector spaces.*

A 1-way tensor is a vector, $v = [v_1 \ v_2 \ \dots \ v_n]^T$.

A 2-way tensor is a matrix, $A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{pmatrix}$.

Definition 9 (Neural population response) *Suppose \mathcal{S} is a set of S visual stimuli (e.g., images) $\mathcal{S} = \{s_1, s_2, \dots, s_S\}$, each moving over a time interval of T in d directions. The neural population response of a set of N neurons to a stimulus m_i over time is $\mathcal{N} = \{\vec{n}_1, \vec{n}_2, \dots, \vec{n}_N\}$, where $\vec{n}_i \in \mathbb{R}^{dT}$.*

Definition 10 (Neural tensors) *Each neural tensor encodes the neural population response of a set of neurons to a set of moving visual stimulus over the*

time and directions of the movement. It is thus a 3-way tensor of dimension N -by- S -by- dT .

5.2 Applying tensor CP decomposition

To implement the tensor CP decomposition, we used the software, Tensor Toolbox for MATLAB. (Brett W. Bader et al., 2021) Our code is available in the GitHub repository¹ for this project. After obtaining the tensor factors from tensor CP decomposition, we used PCA to project data onto a lower-dimensional linear subspace and k-means to label the clusters by distances in the lower-dimensional linear subspace.

We can visualize the first five tensor factors from biological neural tensor:

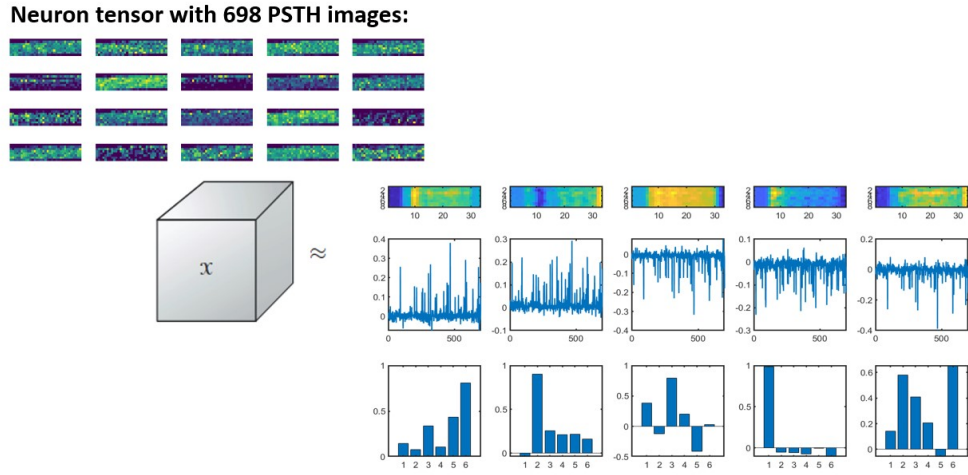


FIGURE 5.2: First 5 tensor factors for neural data.

¹<https://github.com/zhang-liu-official/capstone-liu-2022>

The neural manifold generated from the neural tensor shows the clusters of neurons grouped by their firing patterns encoded in the PSTH diagrams:

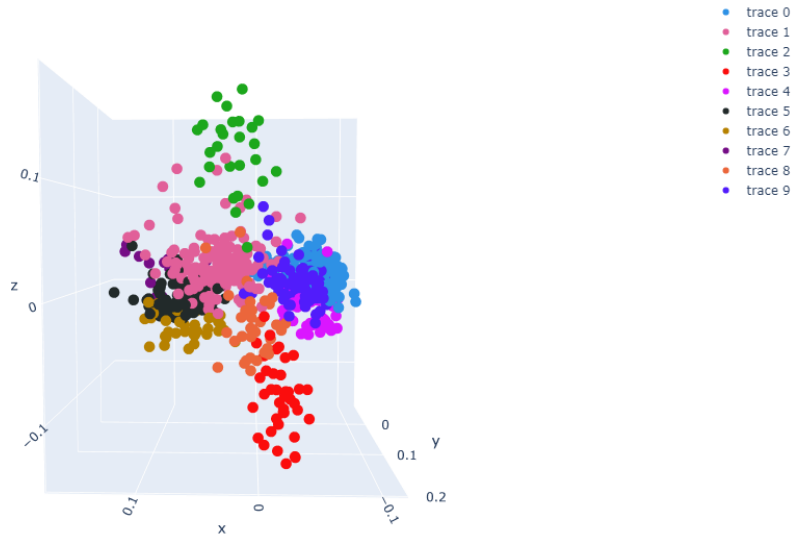
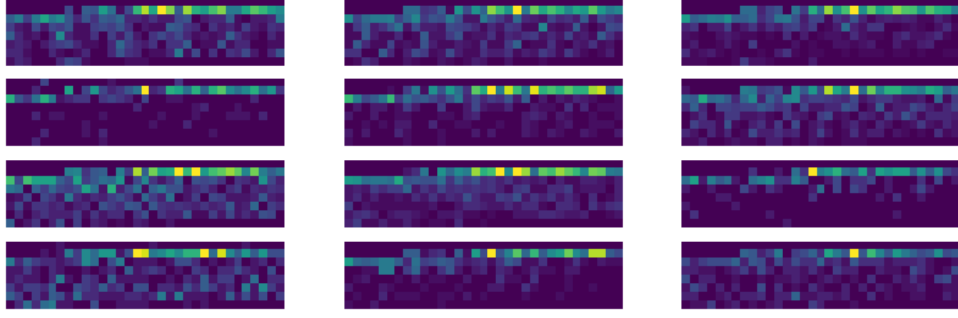
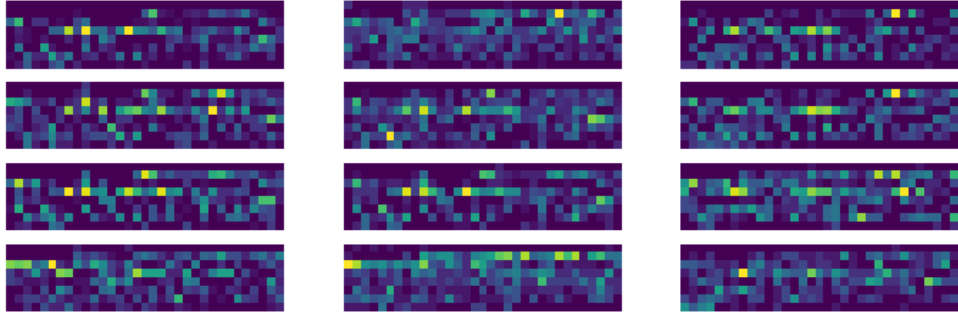


FIGURE 5.3: Neural manifolds: clusters of neurons grouped by firing patterns, each point represent a neuron.

We can plot the PSTH diagrams associated with some points inside the neural manifold, where each point represents a neuron.



(A) PSTH diagrams showing responses of neurons in an arbitrary cluster to stimulus type 1.



(B) PSTH diagrams showing responses of neurons in another cluster to stimulus type 1.

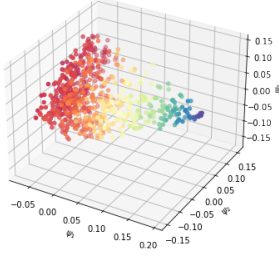
5.3 Applying diffusion map

Recall that the neural spiking data set used in this project is represented by a 3-way tensor of dimension 698-by-6-by-264. The three dimensions represent neurons, stimuli, and firing patterns (PSTH diagram) respectively. With this neural spiking data set, we can create six point clouds, each corresponds to the neural population response towards one type of stimuli, which we denote as X_1, X_2, \dots, X_6 . Each of the point cloud X_i is thus represented by a matrix of dimension 698-by-264, giving us a point cloud of 698 points in \mathbb{R}^{264} .

We used the pydiffmap package (Eastman et al., 2017). In our implementation, we reduced the dimensionality of each point cloud from the space of \mathbb{R}^{264} to \mathbb{R}^3 using diffusion map. The resulting three-dimensional

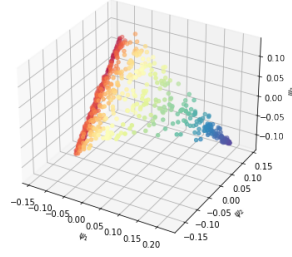
embedding of the original point clouds colored by the first three diffusion coordinates are shown below:

Embedding given by first three DCs.



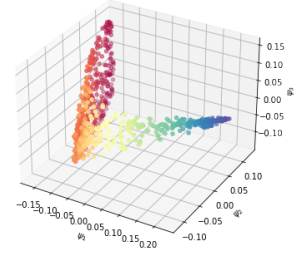
(A) Three-dimensional embedding of the original point cloud X_1 .

Embedding given by first three DCs.



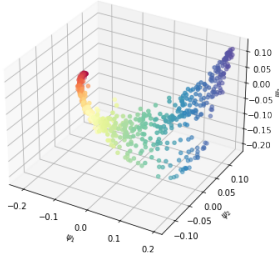
(B) Three-dimensional embedding of the original point cloud X_2 .

Embedding given by first three DCs.



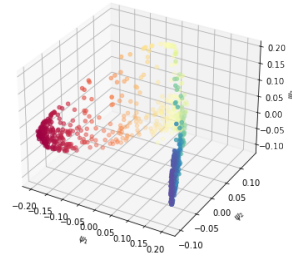
(C) Three-dimensional embedding of the original point cloud X_3 .

Embedding given by first three DCs.



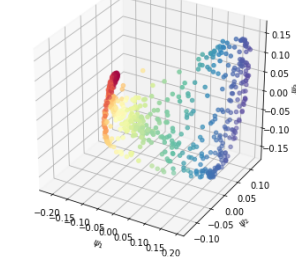
(D) Three-dimensional embedding of the original point cloud X_4 .

Embedding given by first three DCs.



(E) Three-dimensional embedding of the original point cloud X_5 .

Embedding given by first three DCs.



(F) Three-dimensional embedding of the original point cloud X_6 .

Chapter 6

Experiments with Artificial Neural Tensors

We generated the neural manifold for artificial neural networks (ANNs) using the same approach as we did for biological neural networks in the previous chapter. But unlike the latter where the neural tensors were already available from the lab experiments, we need to first build the artificial neural tensor by taking the outputs of neuron units in pre-trained ANNs.

6.1 Building the artificial neural tensor

In the first semester, we have been focusing on investigating the manifold structure of CNN, and specifically the VGG-16 model, but the same method applies to any ANNs. VGG-16 is one of the most successful CNN models in computer vision tasks such as image classification. Its architecture is shown in the figure below. Since previous works have already trained the VGG-16 model to obtain optimal weights, we will directly use the pre-trained weights to generate the artificial neural tensor.

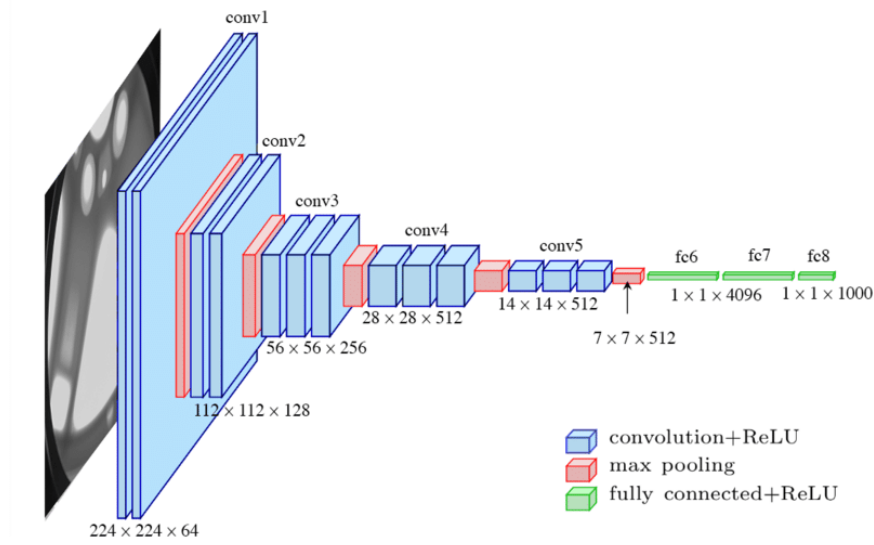


FIGURE 6.1: Visualizing the structure of the VGG-16 model.

6.1.1 “Visual stimuli” in ANNs

Analogous to the visual flows used in lab experiments, the “visual stimuli” for ANNs are natural images of different objects (e.g. cars, cats, and dogs) selected from the commonly used data set, CIFAR-10 (Krizhevsky, 2009). CIFAR-10 consists of 60000 32×32 colour images in 10 classes, with 6000 images per class. To keep the size of the artificial neural tensor manageable, we randomly selected 10 images from each class as the visual stimuli. In order to simulate the movement of visual flows used in lab experiments, we implemented the shifts in image, that is, for every image we shifted the image vertically and horizontally (by one pixel per step).

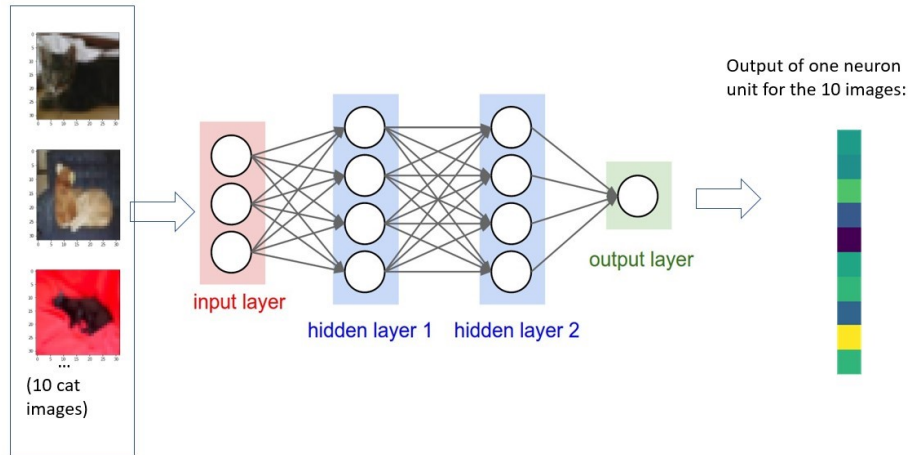


FIGURE 6.2: Computing output of one neuron unit in the ANNs given 10 input cat images.

Remark 11 *At first it might seem that the use of different “visual stimuli” for biological and artificial neural networks will lead to an unfair comparison. However, this approach is appropriate for the following reason. The mouse visual system was trained on naturalistic visual information and the visual flows used in the lab experiments are also naturalistic. The VGG-16 model was trained on ImageNet, which are very similar to the CIFAR-10 data set that we used for “visual stimuli” for ANNs. Thus, in both cases, the visual stimuli are similar to what the respective network was trained to recognize.*

That being said, we could also adapt the experiments for ANNs to take the same visual flows as the input. In fact, it is on the agenda for the next steps in our project.

6.1.2 “Individual neuron” in ANNs

In the lab experiments for biological neural networks, moving visual stimuli trigger spikes in the activation potential in the neurons in the retina. The individual neuron response is the firing rate of the neuron which is

dependent on the electrostatics processes that take place in the synapse. The biological neuron is coarsely modeled by the neuron unit in the ANNs. At the “dendrite,” the input is taken from the “axons” of the connected neurons. The weight of each neuron models the synaptic process in biological neuron. At the “cell body,” all the inputs are multiplied with the weights and summed. We then add the bias term to the sum and apply the non-linear activation function to obtain the final output from this individual neuron unit at the “axon.” The parallel between an individual biological and artificial neuron is shown in the figures below.

Note that in our experiments, instead of taking the input from previously connected neuron unit, we make a modification by making each neuron unit always take the image vector as the input.

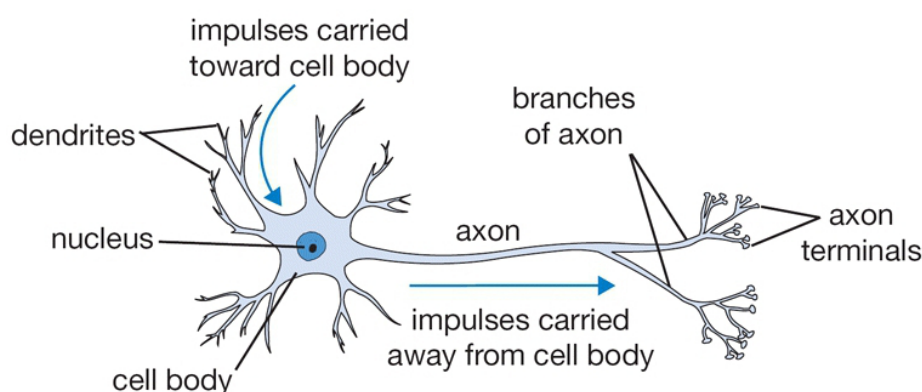


FIGURE 6.3: An individual neuron in biological neural networks. Adapted from *Notes for CS231n: Convolutional Neural Networks for Visual Recognition*.

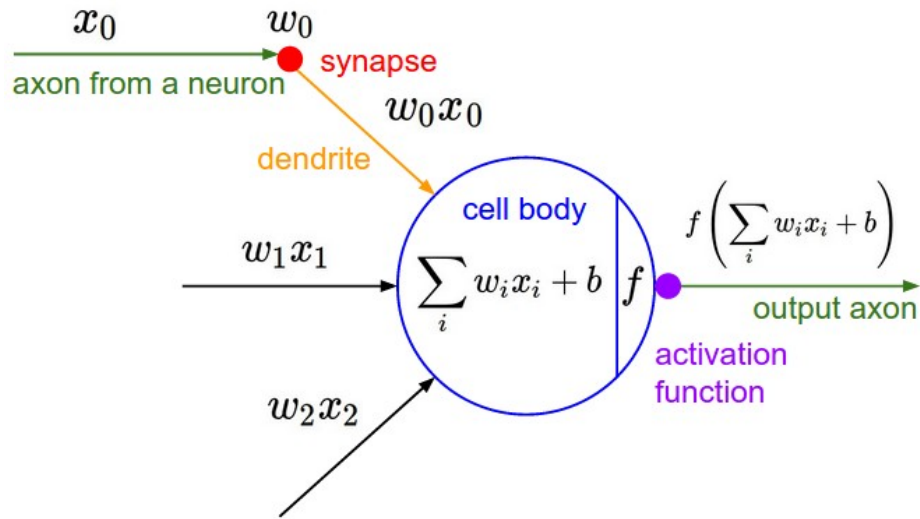


FIGURE 6.4: An "individual neuron" in ANNs. Adapted from *Notes for CS231n: Convolutional Neural Networks for Visual Recognition*.

6.1.3 "Receptive field" of an artificial neuron

Each layer in the CNN has several different feature maps. Each feature map corresponds to a filter of a specific size and specific weights. The filter is essentially a matrix that, when taking inner product with the input image, give an output that highlights some specific features in the input image. To illustrate this, we can visualize the 64 feature maps in the first convolutional layer of the VGG-16 model, given the input bird image.



FIGURE 6.5: Input image. Adapted from *How to Visualize Filters and Feature Maps in Convolutional Neural Networks*.



FIGURE 6.6: Visualizing the 64 feature maps in the first convolutional layer of the VGG-16 model. Adapted from *How to Visualize Filters and Feature Maps in Convolutional Neural Networks*.

Each feature map will have a number of neurons. Each neuron attends to a specific subregion of the image, which is called their “receptive field.” This term in fact originated from the study of the biological visual system, which is defined as a restricted region of visual space where a luminous stimulus could drive electrical responses in a retinal ganglion cell.

The size of the receptive field of is the same as the filter size corresponding to the feature map in which the neuron is located. Each neuron will have the same set of weights as the other neurons in the same feature map (this is sometimes called “weights sharing”).

6.1.4 “Neuron population response” in ANNs

Analogous to the neuron population response in biological neural networks, we computed the all the neuron output given the input image and analyze them collectively, instead of looking at each neuron output individually.

However, one difference from the biological neural networks is that given an image input, most of the neurons will have output of small values (since except for the specific neurons that target at recognizing a specific features, a large number of pre-trained weights are close to zero). In other words, the “firing” of neurons in ANNs will be sparse. Thus, we selected only ten feature maps with highest average firing rate (the average is taken over all neurons in the respective feature map).

We now illustrate the dimensions of the resulting artificial neural tensor built from the first convolutional layer of VGG-6. Since there were 1024 neurons in each feature map and we selected 10 feature maps, there were 10240 neurons in total. The visual stimuli consists of 10 images from each of the 10 classes, which give us 100 images. Each of the images have 1024 shifts. In the end, the size of the artificial tensor for the first convolutional layer is thus 10240-by-100-by-1024.

6.2 Applying tensor CP decomposition

Having built the artificial neural tensor, we then applied the tensor CP decomposition to obtain the first five tensor factors:

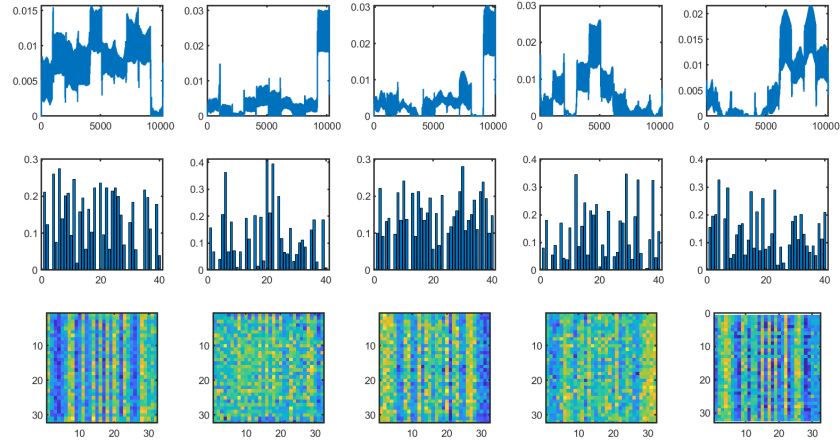


FIGURE 6.7: First 5 tensor factors for artificial neural data.

We then generated the neural manifolds from artificial neural tensor, which can be visualized in the following plot.

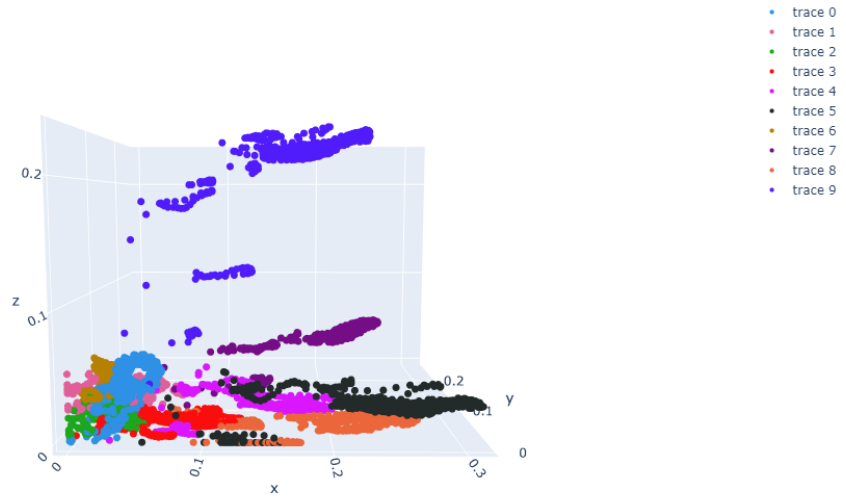


FIGURE 6.8: Neural manifold generated from artificial neural tensor.

Chapter 7

Topological Methods

7.1 Topological data analysis

This chapter provides the details for an explorative study that extends the method by further extracting topological features from the neural manifolds in response to different visual stimuli. We present here an overview of the steps of our implementation and the results. Full details on the theoretical background of persistent homology can be found in [Appendix A](#).

Recall that in Chapter 6, we showed the results of using diffusion map to generate the three-dimensional neural manifolds corresponding to six stimuli types, respectively. In this chapter we describe further experiments that investigated the topological structure of the neural manifolds.

7.2 Step 1: Applying persistent homology

We first applied persistent homology to extract the topological features from the six neural manifolds respectively. In this step, these topological features are represented by persistence barcodes and persistence diagrams.

Our implementation used the package `ripser` (Tralie, Saul, and Bar-On, 2018) to obtain the respective persistence diagrams from the embeddings. Using the (birth, death)-intervals from each persistence diagram, we drew the equivalent persistence barcode representation. The complete code can be found in Appendix A. The following figures show the results from applying persistent homology.

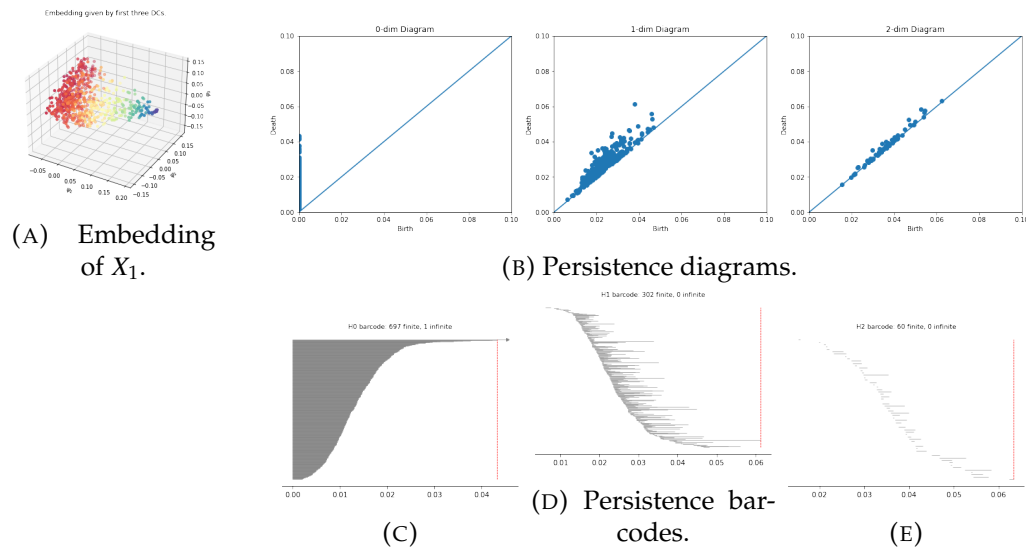


FIGURE 7.1: Results for applying persistent homology on the three-dimensional embedding of X_1 .

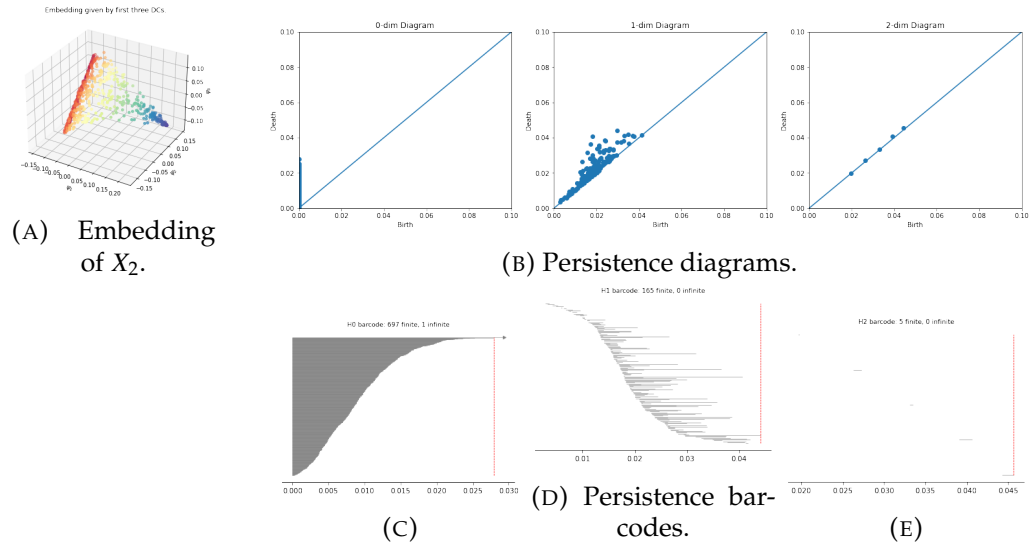


FIGURE 7.2: Results for applying persistent homology on the three-dimensional embedding of X_2 .

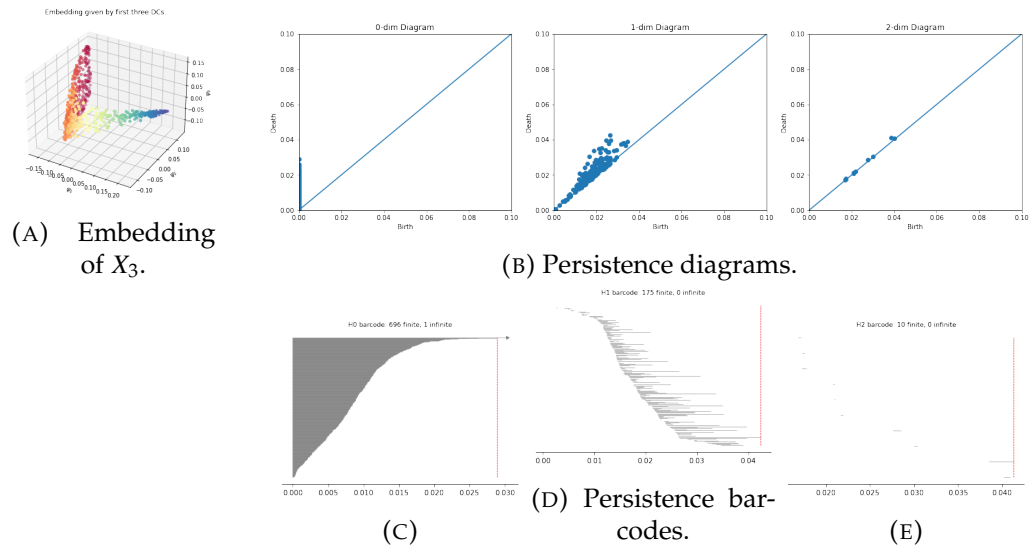


FIGURE 7.3: Results for applying persistent homology on the three-dimensional embedding of X_3 .

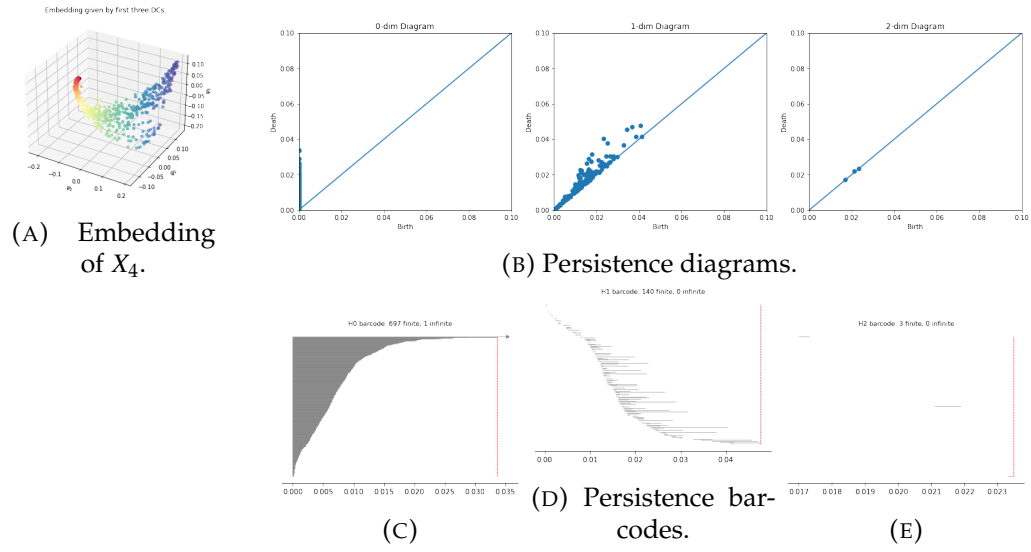


FIGURE 7.4: Results for applying persistent homology on the three-dimensional embedding of X_4 .

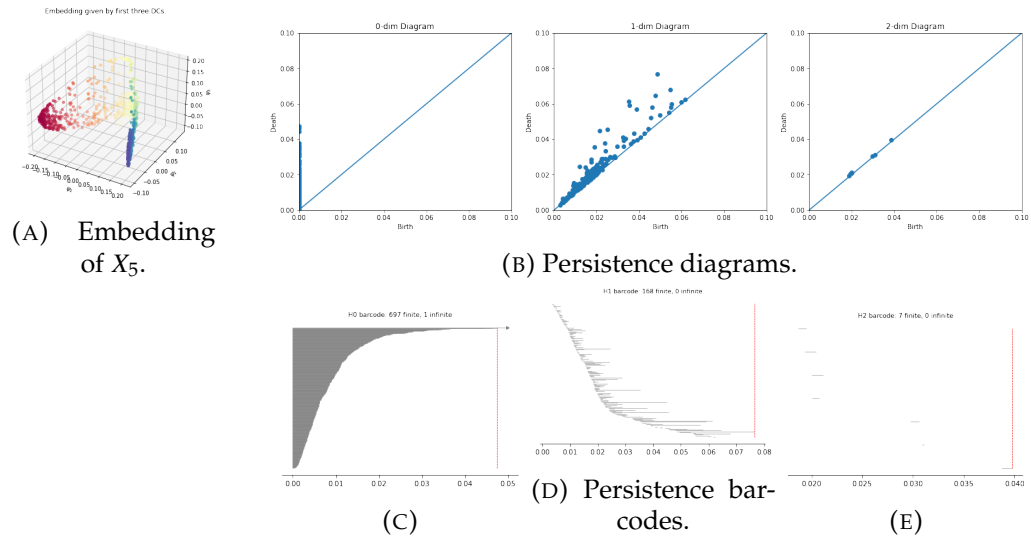


FIGURE 7.5: Results for applying persistent homology on the three-dimensional embedding of X_5 .

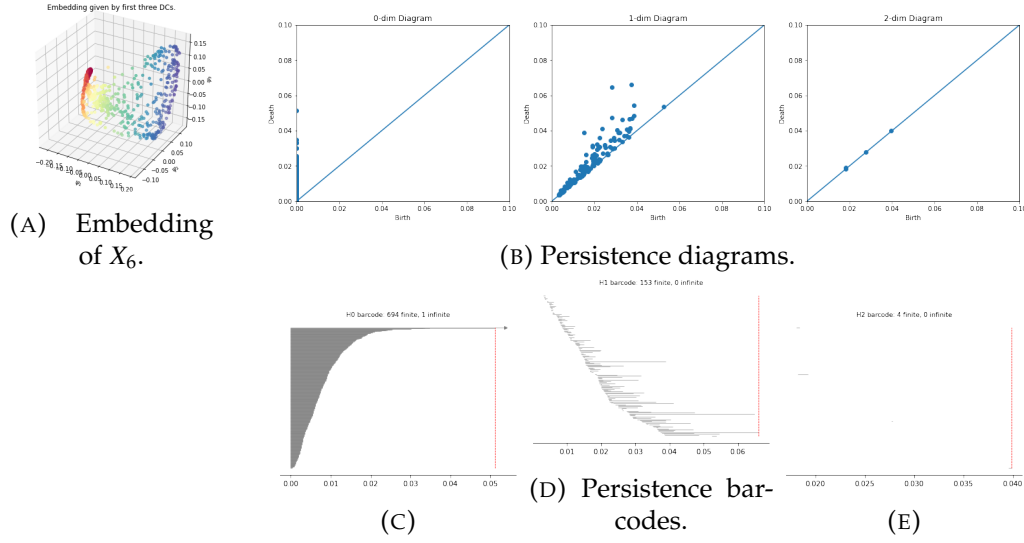


FIGURE 7.6: Results for applying persistent homology on the three-dimensional embedding of X_6 .

7.3 Step 2: Pairwise Wasserstein distance

In this step, we used the gudhi package (The GUDHI Project, 2021) to compute the pairwise Wasserstein distance between the persistence diagrams. The method used is based on (Kerber, Morozov, and Nigmatov, 2016).

We first computed the first pairwise distances between the persistence diagrams for each of the six point clouds.

H_0	X_1	X_2	X_3	X_4	X_5	X_6
X_1	0.0	2.77	3.05	3.95	3.08	3.42
X_2	2.77	0.0	0.43	1.35	1.0	0.84
X_3	3.05	0.43	0.0	1.03	0.94	0.66
X_4	3.95	1.35	1.03	0.0	1.11	0.72
X_5	3.08	1.0	0.94	1.11	0.0	0.51
X_6	3.42	0.84	0.66	0.72	0.51	0.0

TABLE 7.1: Pairwise Wasserstein distance between persistent diagrams for homology group H_0 .

H_1	X_1	X_2	X_3	X_4	X_5	X_6
X_1	0.0	0.46	0.5	0.64	0.64	0.55
X_2	0.46	0.0	0.17	0.29	0.36	0.3
X_3	0.5	0.17	0.0	0.25	0.33	0.3
X_4	0.64	0.29	0.25	0.0	0.27	0.29
X_5	0.64	0.36	0.33	0.27	0.0	0.26
X_6	0.55	0.3	0.3	0.29	0.26	0.0

TABLE 7.2: Pairwise Wasserstein distance between persistent diagrams for homology group H_1 .

H_2	X_1	X_2	X_3	X_4	X_5	X_6
X_1	0.0	0.06	0.06	0.06	0.06	0.06
X_2	0.06	0.0	0.01	0.0	0.01	0.0
X_3	0.06	0.01	0.0	0.0	0.01	0.0
X_4	0.06	0.0	0.0	0.0	0.0	0.0
X_5	0.06	0.01	0.01	0.0	0.0	0.0
X_6	0.06	0.0	0.0	0.0	0.0	0.0

TABLE 7.3: Pairwise Wasserstein distance between persistent diagrams for homology group H_2 .

Building upon the same method, we further compared the six point clouds with known three-dimensional shapes (2-sphere and torus) based on the Wasserstein distance.

First, we extracted the topological features of the 2-sphere and three-dimensional torus, as we did for the point clouds in Step 2.

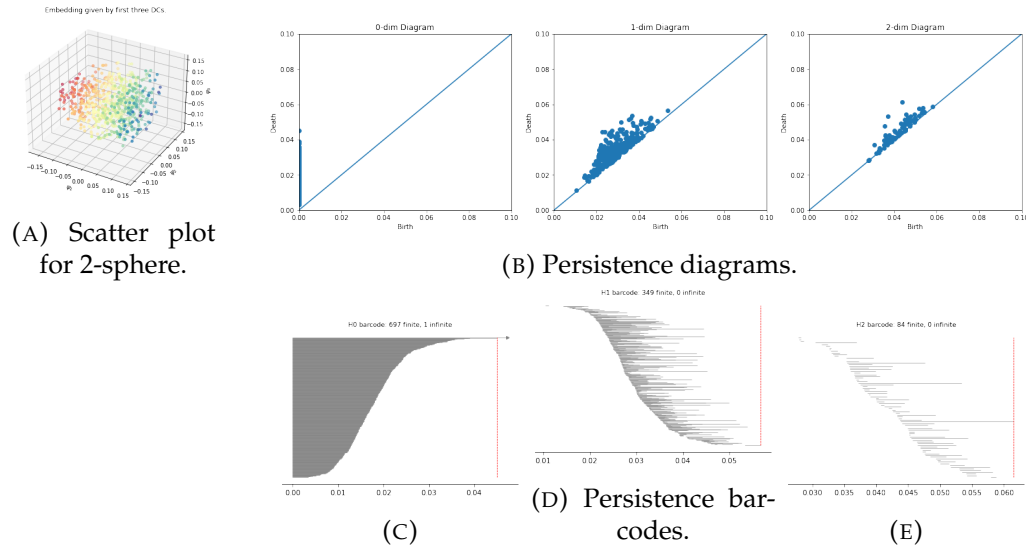


FIGURE 7.7: Results for applying persistent homology on the 2-sphere.

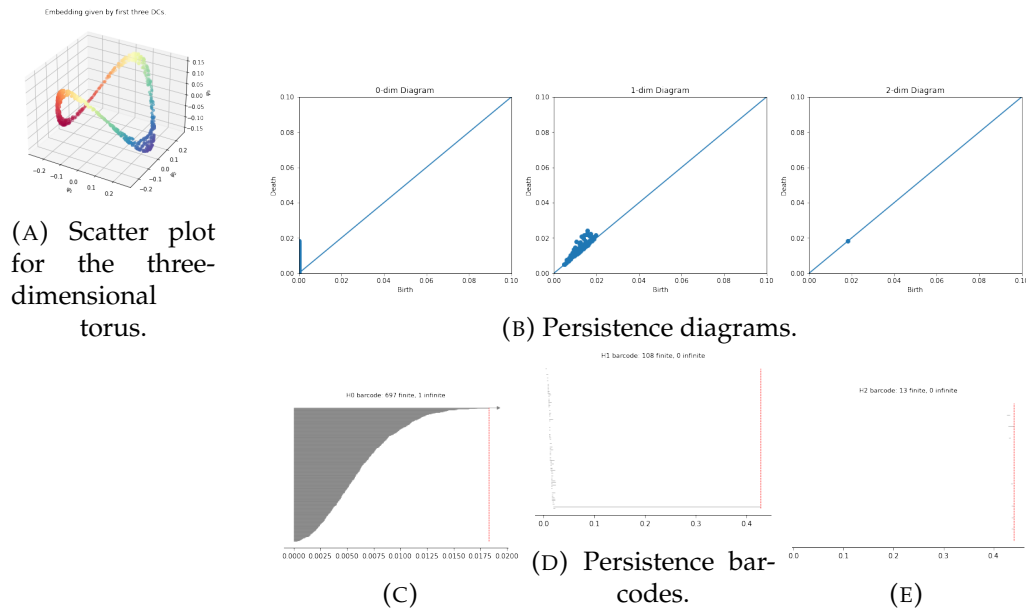


FIGURE 7.8: Results for applying persistent homology on the three-dimensional torus.

Then, we computed the Wasserstein distance between the six point clouds and the known shapes:

H_0	X_1	X_2	X_3	X_4	X_5	X_6
2-sphere	2.48	4.46	4.64	5.18	4.53	4.83
torus	3.63	1.46	1.31	0.73	1.38	1.06

TABLE 7.4: Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_0).

H_1	X_1	X_2	X_3	X_4	X_5	X_6
2-sphere	0.62	0.77	0.80	0.87	0.85	0.76
torus	0.74	0.49	0.44	0.34	0.45	0.47

TABLE 7.5: Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_1).

H_2	X_1	X_2	X_3	X_4	X_5	X_6
2-sphere	0.11	0.12	0.12	0.12	0.13	0.12
torus	0.057	0.017	0.018	0.016	0.018	0.016

TABLE 7.6: Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_2).

7.4 Analysis of results

Based on the results from the above tables of Wasserstein distances, our observations and inferences are as follows.

1. The topological structure for X_1 is distinct from the rest.

Based on the pairwise Wasserstein distances in Tables 7.1 and 7.2, X_1 is notably different from the rest of the point clouds in terms of topological structure. In the context of our application, this implies that the neural population response evoked by stimulus type 1 is significantly different from the other stimulus types. This observation leads us to hypothesize that there is some neuroscientific reason behind this distinction in the topological structure of neural population response. It might be interesting to conduct further lab

experiments to investigate what special properties this stimulus has and why it causes such different neural population response.

2. Wasserstein distances between persistence diagrams are nearly negligible for H_2 .

Based on Table 7.3, pairwise Wasserstein distances between persistence diagrams for H_2 are small enough to be nearly negligible. This implies that the intrinsic dimensionality of this neural data might be even lower than three-dimensional since there is no significant differences in homology groups H_2 for the point clouds.

3. Shape comparison with 2-sphere and torus.

Based on Tables 7.4 and 7.5, the Wasserstein distances between the point clouds and the 2-sphere are smaller than the Wasserstein distances between the point clouds and the torus for all X_i except for X_1 . We can thus infer that except for X_1 , all other point clouds are more similar to the shape of a torus than the 2-sphere. This implies that the topological structure of the neural population response evoked by stimuli type 1 is more similar to a sphere while the neural population response evoked by the rest of the stimuli types used in the experiments are more similar to a torus.

It would be interesting to compare this result with the hypothesis in (Ben-Yishai, Bar-Or, and Sompolsky, 1995, Blumenfeld, Rogat, and Krajcik, 2006, Goldberg et al., 2004, Singh et al., 2008): If we are given an oriented stimulus, and if the orientation is a circular variable, then the hypothesis is that the neural population response

evoked by such stimulus must have a topological structure equivalent to that of a circle. However, to fully test this hypothesis, further experiments with different types of stimuli need to be conducted.

7.5 Discussions

In this chapter, we demonstrated our proposed approach to compare the point clouds that represent the neural population response to different stimuli. Our proposed approach is contingent on the topological structures of the point clouds.

One significant limitation in this specific application is that lab data usually involve a lot of sampling errors such as missing data and inconsistent densities, causing noise to the true underlying geometry of the neural population response. Persistent homology works well on synthetic data, but might not work when the sampling errors are too significant.

The advantages of topological approach is that we can provide a succinct and useful summary of the global geometric structure of the neural spiking data, thus obtaining a useful account of how similarly the neurons collectively respond to visual stimuli. This is especially important in applications where the notion of connectedness and clusters are salient. As an emerging field, TDA will certainly see more applications in solving problems that involve understanding the geometric and topological structure of high-dimensional data.

Chapter 8

Conclusion and Next Steps

8.1 Summary of work done

In the first semester, the main objective is to develop proficiency in the necessary theoretical foundation and implementation skills. These included tensor CP decomposition, diffusion map, and persistent homology. To gain better intuition, I demonstrated each of the methods on toy data sets such as face image data and MNIST Lecun et al., 1998.

Due to the interdisciplinary nature of the project, much time is also spent studying the theoretical foundations of computational neuroscience. This includes the history of computational models of the visual system and the connections with mathematical and computational ideas.

I have also conducted many exploratory experiments. A complete list of the experiments are included below. For the biological neural tensor, I conducted the following experiments:

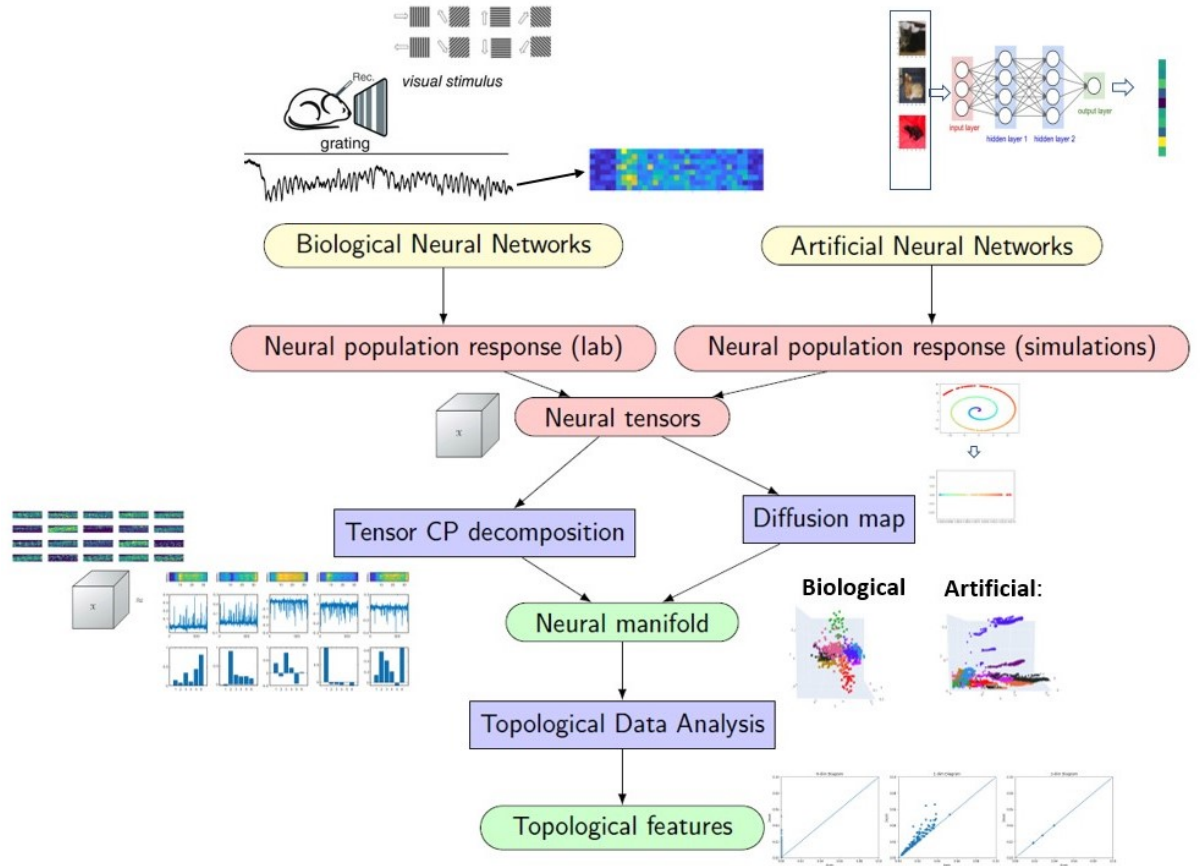
1. applying tensor CP decomposition to generate the neural manifold
2. applying diffusion map to generate the neural manifold

3. applying persistent homology and Wasserstein distance to compare the topological features of the neural manifold

For the artificial neural tensor, I conducted the following experiments:

1. building the artificial neural tensor without considering shifts in images
2. building the artificial neural tensor with shifts in images
3. applying tensor CP decomposition on the artificial neural tensor (with shifts in images)
4. studying how the neural manifold changes during the training of an ANN (instead of using the pre-trained ANN)
5. applying persistent homology (though with limited explainability)

In all, I am on track with the timeline and have completed more than the plans we laid out at the beginning of the semester. A bird-eye view of the work done during the first semester is shown in the figure below.



8.2 Next steps

In the past few months, I have finished learning the foundations behind the methods and applying these methods to conduct relevant experiments using the neural tensors from the lab neural spiking data and the pre-trained VGG-16 model.

Given the progress made, I propose the following next steps:

- Provide more thorough and rigorous interpretations of the results for CNN. This will likely involve more sophisticated understanding of theoretical neuroscience, which could be addressed by reading

the book on neuromathematics and talking to the co-supervisor who is working directly in this area.

- Apply the same methods to RNN to investigate whether there would be any interesting differences.
- Come up with a way to quantify the differences between neural manifolds. This could potentially be done by using persistent homology and the Wasserstein distance.
- Implement diffusion map on the artificial neural tensors with the appropriate parameters.
- Identify a potential novelty that this project could offer, which would be a contribution that is not in the research literature before. This could be some interesting results from the experiments, or a proposal of a new improvement to the existing methodology.

Bibliography

- Ben-Yishai, R, R L Bar-Or, and H Sompolinsky (Apr. 1995). "Theory of orientation tuning in visual cortex." In: *Proceedings of the National Academy of Sciences* 92.9, p. 3844. DOI: [10.1073/pnas.92.9.3844](https://doi.org/10.1073/pnas.92.9.3844). URL: <http://www.pnas.org/content/92/9/3844.abstract>.
- Beshkov, Kosio and Paul Tiesinga (May 2021). *Geodesic-based distance reveals non-linear topological features in neural activity from mouse visual cortex*. en. preprint. Neuroscience. DOI: [10.1101/2021.05.21.444993](https://doi.org/10.1101/2021.05.21.444993). URL: <http://biorxiv.org/lookup/doi/10.1101/2021.05.21.444993> (visited on 11/05/2021).
- Blumenfeld, Phyllis, Toni Rogat, and Joseph Krajcik (Jan. 2006). "Motivation and Cognitive Engagement in Learning Environments." In: *The Cambridge Handbook of the Learning Sciences*.
- Brett W. Bader, Tamara G. Kolda et al. (Apr. 5, 2021). *Tensor Toolbox for MATLAB*. Version 3.2.1. URL: <https://www.tensortoolbox.org/>.
- Carlsson, Gunnar (Jan. 2009). "Topology and data". en. In: *Bulletin of the American Mathematical Society* 46.2, pp. 255–308. ISSN: 0273-0979. DOI: [10.1090/S0273-0979-09-01249-X](https://doi.org/10.1090/S0273-0979-09-01249-X). URL: <http://www.ams.org/journal-getitem?pii=S0273-0979-09-01249-X> (visited on 11/02/2021).

- Chaudhuri, Rishidev et al. (Sept. 2019). "The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep". In: *Nature Neuroscience* 22.9, pp. 1512–1520. ISSN: 1546-1726. DOI: [10.1038/s41593-019-0460-x](https://doi.org/10.1038/s41593-019-0460-x). URL: <https://doi.org/10.1038/s41593-019-0460-x>.
- Chung, SueYeon and L F Abbott (2021). "Neural population geometry: An approach for understanding biological and artificial neural networks". In: p. 12.
- Coifman, R. R. et al. (2005). "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps". In: *Proceedings of the National Academy of Sciences* 102.21, pp. 7426–7431. ISSN: 0027-8424, 1091-6490. DOI: [10.1073/pnas.0500334102](https://doi.org/10.1073/pnas.0500334102). URL: <http://www.pnas.org/cgi/doi/10.1073/pnas.0500334102> (visited on 08/05/2021).
- Coifman, Ronald R. and Stéphane Lafon (2006). "Diffusion maps". In: *Applied and Computational Harmonic Analysis* 21.1, pp. 5–30. ISSN: 10635203. DOI: [10.1016/j.acha.2006.04.006](https://doi.org/10.1016/j.acha.2006.04.006). URL: <https://linkinghub.elsevier.com/retrieve/pii/S1063520306000546> (visited on 08/05/2021).
- DeepAI (2019). *Manifold hypothesis*. URL: <https://deepai.org/machine-learning-glossary-and-terms/manifold-hypothesis>.
- DiCarlo, James J, Davide Zoccolan, and Nicole C Rust (2012). "How does the brain solve visual object recognition?" In: *Neuron* 73.3, pp. 415–434.
- Dyballa, Luciano (2021). "The Manifold of Neural Responses Informs Physiological Circuits in the Visual System". English. ISBN: 9798534651942
Publication Title: ProQuest Dissertations and Theses 28321876. Ph.D.

- Ann Arbor: Yale University. URL: <https://www.proquest.com/dissertations-theses/manifold-neural-responses-informs-physiological/docview/2557257234/se-2?accountid=10267>.
- Eastman, Peter et al. (2017). "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics". In: *PLOS Computational Biology* 13.7. DOI: [10.1371/journal.pcbi.1005659](https://doi.org/10.1371/journal.pcbi.1005659).
- Fefferman, Charles, Sanjoy Mitter, and Hariharan Narayanan (2016). "Testing the manifold hypothesis". In: *J. Amer. Math. Soc.* 29.4, pp. 983–1049. ISSN: 0894-0347, 1088-6834. DOI: [10.1090/jams/852](https://doi.org/10.1090/jams/852). URL: <https://www.ams.org/jams/2016-29-04/S0894-0347-2016-00852-4/> (visited on 10/25/2021).
- Fukushima, Kunihiko (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biol. Cybernetics* 36.4, pp. 193–202. ISSN: 0340-1200, 1432-0770. DOI: [10.1007/BF00344251](https://doi.org/10.1007/BF00344251). URL: <http://link.springer.com/10.1007/BF00344251> (visited on 10/25/2021).
- Gallego, Juan A. et al. (2017). "Neural Manifolds for the Control of Movement". In: *Neuron* 94.5, pp. 978–984. ISSN: 08966273. DOI: [10.1016/j.neuron.2017.05.025](https://doi.org/10.1016/j.neuron.2017.05.025). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0896627317304634> (visited on 10/23/2021).
- Geman, Stuart (1999). "Hierarchy in machine and natural vision". In: *Proceedings of the Scandinavian Conference on Image Analysis*. Vol. 1, pp. 179–184.
- Goldberg, Richard M. et al. (Jan. 2004). "A Randomized Controlled Trial of Fluorouracil Plus Leucovorin, Irinotecan, and Oxaliplatin Combinations in Patients With Previously Untreated Metastatic Colorectal

- Cancer". en. In: *Journal of Clinical Oncology* 22.1, pp. 23–30. ISSN: 0732-183X, 1527-7755. DOI: [10 . 1200 / JCO . 2004 . 09 . 046](https://doi.org/10.1200/JCO.2004.09.046). URL: [http : // ascopubs . org / doi / 10 . 1200 / JCO . 2004 . 09 . 046](http://ascopubs.org/doi/10.1200/JCO.2004.09.046) (visited on 11/05/2021).
- Gwilliams, Laura and Jean-Rémi King (2017). "Performance-optimized hierarchical models only partially predict neural responses during perceptual decision making". In: *bioRxiv*. DOI: [10 . 1101 / 221630](https://doi.org/10.1101/221630).
- Hegde, Jay and Daniel J Felleman (2007). "Reappraising the functional implications of the primate visual anatomical hierarchy". In: *The Neuroscientist* 13.5, pp. 416–421.
- Hoffmann, KP, J Stone, and S Murray Sherman (1972). "Relay of receptive-field properties in dorsal lateral geniculate nucleus of the cat." In: *Journal of Neurophysiology* 35.4, pp. 518–531.
- Hong, David, Tamara G. Kolda, and Jed A. Duersch (2020). "Generalized Canonical Polyadic Tensor Decomposition". In: *SIAM Review* 62.1, pp. 133–163. ISSN: 0036-1445, 1095-7200. DOI: [10 . 1137 / 18M1203626](https://doi.org/10.1137/18M1203626). arXiv: [1808 . 07452](https://arxiv.org/abs/1808.07452). URL: [http : // arxiv . org / abs / 1808 . 07452](http://arxiv.org/abs/1808.07452) (visited on 06/10/2021).
- How to Visualize Filters and Feature Maps in Convolutional Neural Networks.* <https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/>. Accessed: 2021-11-08.
- Hubel, D H and T N Wiesel (Jan. 1962). "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". eng. In: *The Journal of physiology* 160.1, pp. 106–154. ISSN: 0022-3751. DOI: [10 . 1113 / jphysiol . 1962 . sp006837](https://doi.org/10.1113/jphysiol.1962.sp006837). URL: <https://pubmed.ncbi.nlm.nih.gov/14449617>.

Kerber, Michael, Dmitriy Morozov, and Arnur Nigmatov (June 2016).

“Geometry Helps to Compare Persistence Diagrams”. en. In: *arXiv:1606.03357 [cs]*. arXiv: 1606.03357. URL: <http://arxiv.org/abs/1606.03357> (visited on 10/01/2021).

Kolda, Tamara G. and Brett W. Bader (2009). “Tensor Decompositions and Applications”. In: *SIAM Review* 51.3, pp. 455–500. ISSN: 0036-1445, 1095-7200. DOI: [10.1137/07070111X](https://doi.org/10.1137/07070111X). URL: <http://epubs.siam.org/doi/10.1137/07070111X> (visited on 06/10/2021).

Krizhevsky, Alex (2009). *Learning multiple layers of features from tiny images*. Tech. rep.

Lecun, Y. et al. (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *Nature* 521.7553, pp. 436–444. ISSN: 1476-4687. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539). URL: <https://doi.org/10.1038/nature14539>.

Martinez, Luis M. and Jose-Manuel Alonso (Oct. 2003). “Complex Receptive Fields in Primary Visual Cortex”. In: *The Neuroscientist* 9.5. Publisher: SAGE Publications Inc STM, pp. 317–331. ISSN: 1073-8584. DOI: [10.1177/1073858403252732](https://doi.org/10.1177/1073858403252732). URL: <https://doi.org/10.1177/1073858403252732> (visited on 11/05/2021).

Notes for CS231n: Convolutional Neural Networks for Visual Recognition. <https://cs231n.github.io/>. Accessed: 2021-11-08.

- Okun, Michael et al. (May 2015). "Diverse coupling of neurons to populations in sensory cortex". In: *Nature* 521.7553, pp. 511–515. ISSN: 1476-4687. DOI: [10.1038/nature14273](https://doi.org/10.1038/nature14273). URL: <https://doi.org/10.1038/nature14273>.
- Poggio, Tomaso, Steve Smale, et al. (2003). "The mathematics of learning: Dealing with data". In: *Notices of the AMS* 50.5, pp. 537–544.
- Riesenhuber, Maximilian and Tomaso Poggio (1999). "Hierarchical models of object recognition in cortex". In: *Nature neuroscience* 2.11, pp. 1019–1025.
- Sadtler, Patrick T. et al. (Aug. 2014). "Neural constraints on learning". In: *Nature* 512.7515, pp. 423–426. ISSN: 1476-4687. DOI: [10.1038/nature13665](https://doi.org/10.1038/nature13665). URL: <https://doi.org/10.1038/nature13665>.
- Singh, Gurjeet et al. (June 2008). "Topological analysis of population activity in visual cortex". In: *Journal of Vision* 8.8, pp. 11–11. ISSN: 1534-7362. DOI: [10.1167/8.8.11](https://doi.org/10.1167/8.8.11). eprint: https://arvojournals.org/arvo/content/_public/journal/jov/933530/jov-8-8-11.pdf. URL: <https://doi.org/10.1167/8.8.11>.
- Stanley III, Jay S. (2020). "Geometric Data Organization: Algorithms and Applications". English. ISBN: 9798516931055 Publication Title: ProQuest Dissertations and Theses 27736931. Ph.D. Ann Arbor: Yale University. URL: <https://www.proquest.com/dissertations-theses/geometric-data-organization-algorithms/docview/2544526068/se-2?accountid=10267>.
- Stopfer, Mark, Vivek Jayaraman, and Gilles Laurent (Sept. 2003). "Intensity versus Identity Coding in an Olfactory System". In: *Neuron* 39.6, pp. 991–1004. ISSN: 0896-6273. DOI: [10.1016/j.neuron.2003.08.011](https://doi.org/10.1016/j.neuron.2003.08.011).

URL: <https://www.sciencedirect.com/science/article/pii/S089662730300535X>.

The GUDHI Project (2021). *GUDHI User and Reference Manual*. 3.4.1. GUDHI Editorial Board. URL: https://gudhi.inria.fr/python/latest/wasserstein_distance_user.html.

Tralie, Christopher, Nathaniel Saul, and Rann Bar-On (2018). “Ripser.py: A Lean Persistent Homology Library for Python”. In: *The Journal of Open Source Software* 3.29, p. 925. DOI: [10.21105/joss.00925](https://doi.org/10.21105/joss.00925). URL: <https://doi.org/10.21105/joss.00925>.

Tsodyks, Misha (Jan. 1999). “Attractor neural network models of spatial maps in hippocampus”. In: *Hippocampus* 9.4. Publisher: John Wiley & Sons, Ltd, pp. 481–489. ISSN: 1050-9631. DOI: [10.1002/\(SICI\)1098-1063\(1999\)9:4<481::AID-HIP014>3.0.CO;2-S](https://doi.org/10.1002/(SICI)1098-1063(1999)9:4<481::AID-HIP014>3.0.CO;2-S). URL: [https://doi.org/10.1002/\(SICI\)1098-1063\(1999\)9:4<481::AID-HIP014>3.0.CO;2-S](https://doi.org/10.1002/(SICI)1098-1063(1999)9:4<481::AID-HIP014>3.0.CO;2-S) (visited on 11/02/2021).

Williams, Alex H. et al. (2018). “Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis”. In: *Neuron* 98.6, 1099–1115.e8. ISSN: 08966273. DOI: [10.1016/j.neuron.2018.05.015](https://doi.org/10.1016/j.neuron.2018.05.015). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0896627318303878> (visited on 06/10/2021).

Yamashita, Rikiya et al. (Aug. 2018). “Convolutional neural networks: an overview and application in radiology”. In: *Insights into Imaging* 9.4, pp. 611–629. ISSN: 1869-4101. DOI: [10.1007/s13244-018-0639-9](https://doi.org/10.1007/s13244-018-0639-9). URL: <https://doi.org/10.1007/s13244-018-0639-9>.

Yu, Byron M. et al. (July 2009). "Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity". In: *Journal of Neurophysiology* 102.1. Publisher: American Physiological Society, pp. 614–635. ISSN: 0022-3077. DOI: [10.1152/jn.90941.2008](https://doi.org/10.1152/jn.90941.2008). URL: <https://doi.org/10.1152/jn.90941.2008> (visited on 11/02/2021).

Zomorodian, Afra (2005). "Computing Persistent Homology". en. In: p. 15.

Appendix A

Persistent Homology Theory

In this appendix, we provide the full details on the theoretical background of persistent homology. The main references for the following exposition are (Carlsson, 2009) and (Zomorodian, 2005).

A.1 Motivation

First of all, there are four major advantages for using topological methods to analyze high-dimensional point clouds.

1. Topology provides qualitative information which is required for data analysis.
2. Metrics are not theoretically justified. Compared to straightforward geometric methods, Topology is less sensitive to the actual choice of metrics.
3. Studying geometric objects using Topology does not depend on the coordinates.
4. Functoriality. This is the most important advantage.

Definition 12 (Functoriality) For any topological space X , abelian group A , and integer $k \geq 0$, there is assigned a group $H_k(X, A)$. For any A and k , and any continuous map $f : X \rightarrow Y$, there is an induced homomorphism $H_k(f, A) : H_k(X, A) \rightarrow H_k(Y, A)$. Then functoriality refers to the following conditions:

- $H_k(f \circ g, A) : H_k(f, A) \circ H_k(g, A)$
- $H_k(\text{Id}_X; A) = \text{Id}_{H_k(X, A)}$.

Functoriality addresses the ambiguities in statistical clustering methods – in particular the arbitrariness of various threshold choices. We now illustrate how exactly functoriality could be used in questions related to clustering.

Let X be the full data set and X_1, X_2 are the subsamples from the data set. If the set of clusterings $C(X_1), C(X_2), C(X_1 \cup X_2)$ correspond well, then we can conclude that the subsample clusterings correspond to clusterings in the full data set X .

A.2 Homotopy

Definition 13 (Homotopy) Homotopy is a family of maps $f_t : X \rightarrow Y$ where $t \in I$ such that $F : X \times I \rightarrow Y$ defined by $F(x, t) \mapsto f_t(x)$ is continuous.

Two maps f_0, f_1 are homotopic if there exists a homotopy f_t between f_0 and f_1 .

Definition 14 (Homotopy equivalence) A map $f : X \rightarrow Y$ is a homotopy equivalence if there is a map $g : Y \rightarrow X$ such that

- $f \circ g$ is homotopic to the identity map on Y , and
- $g \circ f$ is homotopic to f .

Two spaces X, Y are homotopy equivalent if there exists a homotopy equivalence $f : X \rightarrow Y$.

Theorem 15 If f and g are homotopic, then $H_k(f, A) = H_k(g, A)$.

If X and Y are homotopy equivalent, then $H_k(X, A) \cong H_k(Y, A)$.

Note that if two spaces are homotopy equivalent, then all their Betti numbers (defined in subsequent section) are equal.

Definition 16 (Topological space) Associated to a simplicial complex (V, Δ) is a topological space $|(\Delta)|$. $|(\Delta)|$ may be defined using a bijection $\phi : V \rightarrow \{1, 2, \dots, N\}$ as the subspace of \mathbb{R}^N given by the union

$$\bigcup_{\sigma \in \Delta} c(\sigma),$$

where $c(\sigma)$ is the convex hull of the set $\{e_{\phi(s)}\}_{s \in \sigma}$, where e_i denotes the i -th standard basis vector.

We often use abstract simplicial complexes to approximate topological spaces. For simplicial complexes the homology can be computed using only the linear algebra of finitely generated \mathbb{Z} -modules. In particular, for simplicial complexes, homology is algorithmically computable.

Definition 17 (Nerve) Let X be a topological space, and let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ be any covering of X .

The nerve of \mathcal{U} , denoted by $N(\mathcal{U})$, will be the abstract simplicial complex with vertex set A , and where a family $\{\alpha_0, \dots, \alpha_k\}$ spans a k -simplex if and only if $U_{\alpha_0} \cap U_{\alpha_1} \cap \dots \cap U_{\alpha_k} \neq \emptyset$.

One reason that this construction is very useful in the following “Nerve Theorem.” This theorem gives the criteria for $N(\mathcal{U})$ to be homotopy equivalent to the underlying topological space X .

Theorem 18 (Nerve Theorem) *Suppose that X and U are as above, and suppose that the covering consists of open sets and is numerable. Suppose further that for all $\emptyset \subseteq A$, we have that $\bigcap_{s \in S} U_s$ is either contractible or empty. Then $N(\mathcal{U})$ is homotopy equivalent to X .*

The Nerve Theorem is very important in TDA since it provides a way to encode the topology of continuous spaces into abstract combinatorial structures that are more useful for designing computationally efficient data structures and algorithms.

A.3 Simplicial Complexes

Definition 19 (Simplicial complex) *An abstract simplicial complex is a pair (V, Δ) , where V is a finite set, and Δ is a family of non-empty subsets of V such that*

$$\tau \in \Delta \text{ and } \sigma \subseteq \tau \implies \sigma \in \Delta.$$

$\tau \in \Delta$ is face of Δ . The dimension of a face τ is $|\tau| - 1$.

(Intuition) A simplicial complex Δ in \mathbb{R}^n is a collection of simplices in \mathbb{R}^n such that

1. *Every face of a simplex of Δ is in Δ .*
2. *The intersection of any two simplices of Δ is a face of each.*

Definition 20 (f-vector) For each $n \geq -1$, let $f_n = f_n(\Delta)$ be the number of faces of Δ of dimension n .

The f -vector of Δ is the vector $(f_{-1}, f_0, f_1, \dots, f_d)$ where d is the dimension of Δ .

Example 21 Suppose the family of sets

$$\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, c\}$$

form a simplicial complex E_1 . Then the f -vector of E_1 is $(1, 4, 5, 1)$.

Definition 22 (Oriented complex) (Intuition) An oriented simplex is a simplex σ together with an orientation of σ . If $\{a_0, a_1, \dots, a_p\}$ spans a p -complex σ , then we denote the oriented simplex with $[a_0, a_1, \dots, a_p]$.

(Formal) Let \mathbb{F} be a commutative ring, Δ be a simplicial complex. For each $n \geq -1$, we form a free \mathbb{F} -module $\tilde{C}_n(\Delta; \mathbb{F})$ with a basis indexed by the n -dimensional faces of Δ . For each n -dimensional face $a_0 a_1 \cdots a_n$, we have a basis element $\vec{e}_{a_0, a_1, \dots, a_n}$.

We refer to the basis element $\vec{e}_{a_0, a_1, \dots, a_n}$ as an oriented simplex. The concept of an oriented simplex is analogous to the idea of a unit vector (which gives the direction and has unit length).

Definition 23 (Chain group of degree n) We refer to the free \mathbb{F} -module $\tilde{C}_n(\Delta; \mathbb{F})$ in the previous definition as the chain group of degree n . The rank of $\tilde{C}_n(\Delta; \mathbb{F})$ is the n -th value $f_n(\Delta)$ in the f -vector of Δ .

Example 24 For $E_1 = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, c\}\}$, we have the following chain groups of degrees $-1, 0, 1, 2$, respectively:

- $\tilde{C}_{-1}(E_1) = \{\lambda e_\emptyset : \lambda \in \mathbb{F}\} \cong \mathbb{F}$.

- $\tilde{C}_0(E_1) = \{\lambda_a e_a + \lambda_b e_b + \lambda_c e_c + \lambda_d e_d : \lambda_a, \lambda_b, \lambda_c, \lambda_d \in \mathbb{F}\} \cong \mathbb{F}^4$.
- $\tilde{C}_1(E_1) = \{\lambda_{ab} e_{a,b} + \dots + \lambda_{cd} e_{c,d} : \lambda_{ab}, \dots, \lambda_{cd} \in \mathbb{F}\} \cong \mathbb{F}^5$.
- $\tilde{C}_2(E_1) = \{\lambda e_{a,b,c} : \lambda \in \mathbb{F}\} \cong \mathbb{F}$.

Definition 25 (Simplicial chain complex) *The simplicial chain complex of a simplicial complex Δ , denoted with $C(\Delta)$, is defined as:*

$$\dots \xrightarrow{\partial_{n+2}} \tilde{C}_{n+1}(\Delta) \xrightarrow{\partial_{n+1}} \tilde{C}_n(\Delta) \xrightarrow{\partial_n} \tilde{C}_{n-1}(\Delta) \xrightarrow{\partial_{n-1}} \dots$$

Example 26 *The simplicial chain complex of E_1 , $C(E_1)$ is*

$$\begin{array}{ccccccccc} 0 & \longrightarrow & \tilde{C}_2(E_1) & \xrightarrow{\partial_2} & \tilde{C}_1(E_1) & \xrightarrow{\partial_1} & \tilde{C}_0(E_1) & \xrightarrow{\partial_0} & \tilde{C}_{-1}(E_1) & \longrightarrow & 0 \\ & & \downarrow \cong & & \downarrow \cong & & \downarrow \cong & & \downarrow \cong & & \\ 0 & \longrightarrow & \mathbb{F} & \xrightarrow{\partial_2} & \mathbb{F}^5 & \xrightarrow{\partial_1} & \mathbb{F}^4 & \xrightarrow{\partial_0} & \mathbb{F} & \longrightarrow & 0 \end{array}$$

Recall that E_1 has f -vector $(1, 4, 5, 1)$.

Proposition 27 (The double boundary condition) *Boundary maps satisfy the double boundary condition, that is, $\partial_n \circ \partial_{n+1} = 0 \quad \forall n$.*

Definition 28 (Simplicial homology of degree k) *(Intuition) The simplicial homology gives an algebraic measure on the amount of cycles that are not the boundaries.*

(Formal) Define the \mathbb{F} -module $Z_k(\Delta; \mathbb{F})$ of cycles and the \mathbb{F} -module of the boundary group $B_k(\Delta; \mathbb{F})$ by the following formulas:

- $Z_k(\Delta; \mathbb{F}) = \ker(\partial_k) = \{Z \in \tilde{C}_k(\Delta; \mathbb{F}) : \partial_k(Z) = 0\}$
- $B_k(\Delta; \mathbb{F}) = \text{im}(\partial_{k+1}) = \{Z \in \tilde{C}_k(\Delta; \mathbb{F}) : \partial_{k+1}(x), \quad x \in \tilde{C}_{k+1}(\Delta; \mathbb{F})\}$

By 27, $B_k(\Delta; \mathbb{F})$ is a submodule of $Z_k(\Delta; \mathbb{F})$, i.e., $B_k(\Delta; \mathbb{F}) \subseteq Z_k(\Delta; \mathbb{F}) \subseteq C_k(\Delta; \mathbb{F})$.

We define the simplicial homology in degree k of Δ to be the quotient group

$$\tilde{H}_k(\Delta; \mathbb{F}) = \ker(\partial_k) / \text{im}(\partial_{k+1}) \quad (\text{A.1})$$

$$= Z_k(\Delta; \mathbb{F}) / B_k(\Delta; \mathbb{F}) \quad (\text{A.2})$$

Definition 29 (k -th Betti number) The k -th Betti number of the simplicial complex Δ is

$$\beta_k(\Delta; \mathbb{F}) := \dim \tilde{H}_k(\Delta; \mathbb{F}) \quad (\text{A.3})$$

$$= \dim \ker(\partial_k) - \dim \text{im}(\partial_{k+1}). \quad (\text{A.4})$$

- elements in $\ker(\partial_k)$ are called k -cycles.
- elements in $\text{im}(\partial_k)$ are called k -boundaries.
- k -cycles that are not boundaries represent k -holes, which means that the k -th Betti number $\beta_k(\Delta; \mathbb{F})$ is the number of k -holes.

Definition 30 (Homology as “holes”) In certain well-behaved cases, homology can be interpreted as an algebraic measure on the amount of “holes” in a simplicial complex Δ .

Formally, given $X \subseteq \mathbb{R}^n$, a “hole” of X is a bounded connected component of $\mathbb{R}^n \setminus X$. For a given $k \geq 0$, suppose the $(k+1)$ -skeleton $\Delta^{(k+1)}$ of Δ has a geometric realization $X \subseteq \mathbb{R}^n$. Then $\tilde{H}_k(\Delta; \mathbb{F})$ is a free \mathbb{F} -module of rank equal to the number of holes of X .

Homology associates a vector space $H_k(X)$ to a topological space X for each $k \in \mathbb{N}$.

- $H_0(X)$ is the number of path components in X .

- $H_1(X)$ is the number of holes in X .
- $H_2(X)$ is the number of voids in X .

Remark 31 Note, however, that in the general metric spaces, the interpretation that homology measures the amount of “holes” does not hold.

A.4 Persistent homology

The main idea of *persistence* is that instead of selecting a fixed value of the threshold ε , we would like to obtain a useful summary of the homological information for all the different values of ε at once. As ε increases, we add simplices to the complexes and detect which features “persist.”

The main advantages of persistent homology are threefold:

1. It is based on algebraic topology.
2. It is computable via linear algebra.
3. It is robust with regards to perturbations in the input data.

Definition 32 (Filtered simplicial complex) A subcomplex of Δ is a subset $\Delta^i \subseteq \Delta$ that is also a simplicial complex. Let Δ be a finite simplicial complex and let $\Delta^1 \subset \Delta^2 \subset \dots \subset \Delta^m = \Delta$ be a finite sequence of nested subcomplexes of Δ . The simplicial complex Δ with such a sequence of subcomplexes, $\emptyset \subseteq \Delta^1 \subseteq \Delta^2 \subseteq \dots \subseteq \Delta^m = \Delta$, is called filtered simplicial complex.

Definition 33 (p -persistent k -th homology group) Given a filtered complex, for the i -th subcomplex Δ^i we can compute the associated

- the boundary maps ∂_k^i for all dimensions k .

- boundary matrices M_k^i for all dimensions k .
- groups C_k^i, Z_k^i (cycle group), B_k^i (boundary group), and H_k^i (homology group).

Then the p -persistent k -th homology group $H_k^{i,p}$ of Δ^i is

- $H_k^{i,p} = Z_k^i / (B_k^{i+p} \cap Z_k^i)$
- (equivalently) $H_k^{i,p} \cong \text{Im}(\eta_k^{i,p})$, where $\eta_k^{i,p}$ is a bijection $\eta_k^{i,p} : H_k^i \rightarrow H_k^{i+p}$ that maps a homology class into another homology class containing it.

Note that this is simply the definition of homology group of degree k in Definition 28 with the additional notion of persistence.

Definition 34 (p -persistent k -th homology group) We now have the persistent version of 29:

The p -persistent k -th Betti number of Δ^i is $\beta_k^i p =$ the rank of the free subgroup of $H_k^{i,p}$.

Definition 35 (Persistence complex) A persistence complex \mathcal{C} is a family of chain complexes $\{C_*^i\}_{i \geq 0}$ over R , together with chain map's $f^i : C_*^i \rightarrow C_*^{i+1}$, so that we have the following diagram:

$$C_*^0 \xrightarrow{f^0} C_*^1 \xrightarrow{f^1} C_*^2 \xrightarrow{f^2} \dots$$

The filtered simplicial complex defined in 32 with inclusion maps for the simplices then becomes a persistence complex. To illustrate, below is a portion of a persistence complex with the the chain complexes expanded.

$$\begin{array}{ccccccc}
 \downarrow \partial_3 & & \downarrow \partial_3 & & \downarrow \partial_3 & & \\
 C_2^0 & \xrightarrow{f^0} & C_2^1 & \xrightarrow{f^1} & C_2^2 & \xrightarrow{f^2} & \dots \\
 \downarrow \partial_2 & & \downarrow \partial_2 & & \downarrow \partial_2 & & \\
 C_1^0 & \xrightarrow{f^0} & C_1^1 & \xrightarrow{f^1} & C_1^2 & \xrightarrow{f^2} & \dots \\
 \downarrow \partial_1 & & \downarrow \partial_1 & & \downarrow \partial_1 & & \\
 C_0^0 & \xrightarrow{f^0} & C_0^1 & \xrightarrow{f^1} & C_0^2 & \xrightarrow{f^2} & \dots
 \end{array}$$

Definition 36 (Persistence modules) A persistence module \mathcal{M} is a family of R -modules M^i , together with homomorphisms $\phi^i : M^i \rightarrow M^{i+1}$.

Theorem 37 (Classification Theorem) For a finite persistence module \mathcal{M} with coefficients in the field F ,

$$H_*(\mathcal{M}; F) \cong \underbrace{\bigoplus_i x^{t_i} F(x)}_{\text{free module}} \oplus \underbrace{\left(\bigoplus_j x^{r_j} (F(x)/(x^{s_j} F[x])) \right)}_{\text{torsion module}} \quad (\text{A.5})$$

There exists a bijection between the set of free elements and the set of homology generators with birth at t_i and persist for all future parameter values.

There exists a bijection between the set of torsion elements and the set of homology generators with birth at r_j and death at $r_j + s_j$.

The classification theorem gives the fundamental characterization of persistence barcode.

Theorem 38 (Barcode as the persistence analogue of Betti number) The rank of $H_k^{i \rightarrow j}(\mathcal{C}; F)$ gives the number of intervals in the barcode of $H_k^{i \rightarrow j}(\mathcal{C}; F)$ spanning the parameter interval $[i, j]$. In particular, $H_*^{i \rightarrow j}(\mathcal{C}_*^i; F)$ gives the number of intervals that contain i .

Remark 39 *As with Betti number, the barcode for H_k does not give the actual structure of the homology group, but just a continuously parameterized rank. The barcode is useful in that it can qualitatively filter out topological noise (since they are “short-lived” features) and capture significant topological features (features that persist over increasing values of ε).*

List of Tables

7.1	Pairwise Wasserstein distance between persistent diagrams for homology group H_0	48
7.2	Pairwise Wasserstein distance between persistent diagrams for homology group H_1	49
7.3	Pairwise Wasserstein distance between persistent diagrams for homology group H_2	49
7.4	Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_0).	51
7.5	Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_1).	51
7.6	Wasserstein distance between persistent diagrams of the point clouds and 2-sphere and torus (homology group H_2).	51

List of Figures

1.1	Visual input goes from the eye to primary visual cortex (V1). Adapted from Polyak (1957).	1
1.2	Illustration of the hierarchical models (Martinez and Alonso, 2003)	5
1.3	Illustration of the parallel models (Martinez and Alonso, 2003)	7
1.4	Illustration of the recurrent models (Martinez and Alonso, 2003)	8
2.1	Examples of 2-dimensional manifolds.	12
2.2	Visualize an example of a neural manifold generated from neural spiking data.	13
3.1	Illustration for PCA. (Adapted from (Williams et al., 2018))	17
3.2	Illustration for tensor CP decomposition. (Adapted from (Williams et al., 2018)	19
3.3	First 5 tensor factors for face image data.	20
3.4	Some arbitrary clusters showing the results of tensor CP decomposition for face data.	21
4.1	Visualising the original spiral data.	26
4.2	First non-trivial coordinate function.	27

4.3	Sample data from the MNIST.	27
4.4	First two non-trivial coordinate functions.	27
5.1	Visualising neural data from lab experiments.	30
5.2	First 5 tensor factors for neural data.	31
5.3	Neural manifolds: clusters of neurons grouped by firing patterns, each point represent a neuron.	32
6.1	Visualizing the structure of the VGG-16 model.	36
6.2	Computing output of one neuron unit in the ANNs given 10 input cat images.	37
6.3	An individual neuron in biological neural networks. Adapted from <i>Notes for CS231n: Convolutional Neural Networks for Visual Recognition</i>	38
6.4	An “individual neuron” in ANNs. Adapted from <i>Notes for CS231n: Convolutional Neural Networks for Visual Recognition</i>	39
6.5	Input image. Adapted from <i>How to Visualize Filters and Feature Maps in Convolutional Neural Networks</i>	40
6.6	Visualizing the 64 feature maps in the first convolutional layer of the VGG-16 model. Adapted from <i>How to Visualize Filters and Feature Maps in Convolutional Neural Networks</i>	41
6.7	First 5 tensor factors for artificial neural data.	43
6.8	Neural manifold generated from artificial neural tensor.	43
7.1	Results for applying persistent homology on the three-dimensional embedding of X_1	45
7.2	Results for applying persistent homology on the three-dimensional embedding of X_2	46

7.3	Results for applying persistent homology on the three-dimensional embedding of X_3	46
7.4	Results for applying persistent homology on the three-dimensional embedding of X_4	47
7.5	Results for applying persistent homology on the three-dimensional embedding of X_5	47
7.6	Results for applying persistent homology on the three-dimensional embedding of X_6	48
7.7	Results for applying persistent homology on the 2-sphere.	50
7.8	Results for applying persistent homology on the three-dimensional torus.	50