An Experimental Study of Persistent Homology Dimension Regularisation

Charlotte Sweeney

4th Year Project Report
Computer Science and Mathematics
School of Informatics
University of Edinburgh

2023

Abstract

We ask the question of whether the generalisation improvements that are made by regularising the persistent homology dimension of network trajectories are a consequence of better compressibility. It is in fact not the case but through answering this question, we identify properties of these regularised models that contradict the current understanding of the role of gradient noise in stochastic gradient descent.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Charlotte Sweeney*)

Table of Contents

Chapter 1

Introduction

Deep neural networks (DNNs) have grown to be a staple in modern-day machine learning, often containing millions or billions of parameters. Despite their wide-spread usage, many questions regarding the generalisation ability of neural networks remain unanswered. DNNs are often heavily overparameterised and so have the capability to simply memorise the training data, yet it is often the case that they are still able to generalise well to unseen data when trained with stochastic gradient descent (SGD). It is a phenomenon that contradicts classic statistical learning theory where good generalisation performance is achieved at a sweet spot between the simplicity and the complexity of the model, hence the usage of regularisation in machine learning methods.

A thorough, undisputed theory to explain the underlying mechanics of SGD remains lacking in the field. It has been a topic of interest for many years and it is generally believed that the stochasticity of SGD acts as a form of implicit regularisation [1, 2]. This property is thought to be the reason SGD is able to achieve such competitive performance against more sophisticated variants, such as ADAM [3]. This implicit regularisation has been observed in the form of a bias towards low norm solutions [4, 5] or solutions that have low complexity [6, 7].

Knowledge of the underlying biases of SGD allows for the development of new optimisation algorithms with these biases in mind. As the behaviour of SGD becomes more well-known, we can gain insight into how to directly influence the methods used to train DNNs towards better generalising solutions. Such a method was developed after recent observations determined that the gradient noise of SGD is better modelled as being drawn from a heavy-tailed distribution rather than a Gaussian distribution [8]. *Persistent homology dimension* regularisation (Dim_{PH} regularisation) [9] seeks to increase the heavy-tailedness of the noise distribution as heavier tails have been shown to be strongly correlated with the generalisation ability of the resulting model [10]. The computation and differentiation of this regularisation term is achieved via methods in topological data analysis - an emerging field in machine learning that applies concepts from computational algebraic topology in a machine learning setting.

This dissertation aims to explore the behaviour of SGD when implemented with Dim_{PH}

regularisation with a particular focus on the links between compression and generalisation. Compressible models are considered to generalise better on unseen data as they can be well-approximated by models with much fewer parameters, allowing for bounds of the generalisation error of the compressed model to also be applied to the uncompressed model [11, 12]. We want to answer the question of

> Can the improvement in generalisation induced by minimising the persistent homology dimension be attributed to the compressibility of the resultant models?

The goal of asking such a question is to better understand the underlying mechanisms of SGD by knowing *why* such solutions have good properties in generalisation.

1.1 Contributions

While the main objective of this dissertation is to the answer the question as presented above, we also uncover a number of additional observations about current assumptions of SGD generalisation. Our contributions may be summarised twofold.

- Our results demonstrate that, despite the sound motivation for considering such a scenario, it is not the case the compressibility is an underlying factor to explain the improvements in generalisation made by Dim_{PH} regularisation.
- We analyse other properties of models trained by Dim_{PH} regularisation and identify differences between the heavy-tailed noise induced by the ratio of step size to batch size and heavy-tailed noise induced by controlling the persistent homology dimension.

1.2 Outline of Report

Chapter 2 describes the background material necessary to understand the computation of the persistent homology dimension and the methods we use in our experiments. Specifically, it provides details on stochastic gradient descent, network pruning, the use of persistent homology in topological data analysis, and what is meant by a fractal dimension alongside definitions of fractal dimensions either used or mentioned within this dissertation.

Chapter 3 details the recent advancements in understanding the mechanics of SGD and how that has lead to the development of Dim_{PH} regularisation. **Chapter 4** explores the existing literature surrounding the heavy-tailed noise assumption of SGD, compression, and generalisation and ties the results to the question we intend to answer in this dissertation. It also provides a breakdown of the question into subpoints. These subpoints are addressed in the experiments we conduct in **Chapter 5** which details the experimental setup and results. Finally, **Chapter 6** concludes the work in this dissertation by summarising and evaluating the achievements and limitations of this project.

Chapter 2

Preliminaries

This chapter will give an overview of the background material required to understand the motivations and results of this dissertation. **Section 2.1** will provide background on stochastic gradient descent; a common algorithm used to train neural networks. S**ection 2.2** will give a high-level overview of network pruning, primarily the motivation for such a process and the main ideas behind it. **Section 2.3** covers the necessary content in topological data analysis, namely persistent homology, to understand the computation of the persistent homology dimension that is introduced in the following section. **Section 2.4** explains what is meant by a fractal dimension and gives definitions of the Hausdorff, box counting, and persistent homology dimension.

Readers who are familiar with the content should feel free to skip to Chapter 3 where a more detailed description of the basis for Dim_{PH} regularisation is discussed. We do, however, recommend reading about the persistent homology dimension is Section 2.4 to understand its computation given in Chapter 3.

2.1 Stochastic Gradient Descent

Stochastic gradient descent (SGD) is a popular algorithm for finding approximate solutions for an optimisation problem. In the context of machine learning, the algorithm aims to find a parameterisation, Σ, of a model, f, that minimises a continuously differentiable loss function, $\mathcal{L}\left(f\left(\mathcal{X};\theta\right),\mathcal{Y}\right)$, where $\mathcal{X},\mathcal{Y}$ denote the set of observations and labels, respectively.

SGD is a variant of full-batch gradient descent. The parameter update rule for full-batch gradient descent is given by,

$$\theta_{t+1} = \theta_t - \eta\nabla\mathcal{L}\left(f\left(\mathcal{X};\theta_t\right),\mathcal{Y}\right),$$

where η is the step size and θ_t are the model parameters found at iteration t.

The gradient vector of the loss function, $\nabla\mathcal{L}\left(f\left(\mathcal{X};\theta_t\right),\mathcal{Y}\right)$, gives the direction of steepest descent. That is, the direction in which the parameters should be shifted towards in order to maximally reduce the loss. It is computed over the whole training set, meaning that the time to compute the gradient grows with the size of the training set.

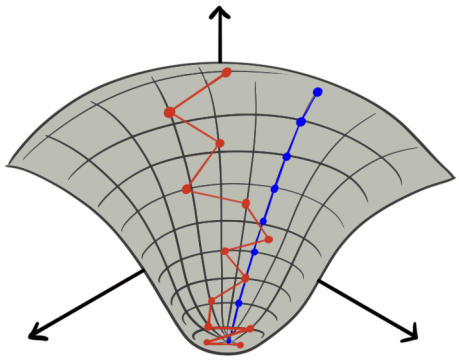


Figure 2.1: Paths of full-batch gradient descent (blue) and stochastic gradient descent (red) converging to a local minimum. Although the path of SGD is much noisier, it does eventually converge near the destination of the full-batch gradient descent path.

This is not a favourable property as the training of machine learning models typically requires vast amounts of data. SGD circumvents such computational complexity by computing the loss function gradient over a random sample of the training set. The parameter update rule for SGD is therefore given by,

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}\left(f\left(\{x_{i_j}\}_{j=1}^{j=b}; \theta_t\right), \{y_{i_j}\}_{j=1}^{j=b}\right), \tag{2.1}$$

where $\{x_{i_j}\}_{j=1}^{j=b}, \{y_{i_j}\}_{j=1}^{j=b}$ are samples of size b that are randomly drawn from the training set at each iteration. The stochastic gradient, $\nabla \widehat{\mathcal{L}} = \nabla \mathcal{L}\left(f\left(\{x_{i_j}\}_{j=1}^{j=b}; \theta_t\right), \{y_{i_j}\}_{j=1}^{j=b}\right)$, is an unbiased estimate[1] of the true gradient, $\nabla \mathcal{L} = \nabla \mathcal{L}\left(f\left(\mathcal{X}; \theta_t\right), \mathcal{Y}\right)$. While the trajectory of SGD iterates will be stochastic, they will converge to the trajectory computed via full-batch gradient descent as illustrated in Figure 2.1.

2.2 Network Pruning

For very large networks, training is often a long and expensive process due to the required computational power, possibly requiring resources beyond what is available. Network pruning aims to reduce the computational power required to run these networks by compressing the network into a smaller one that achieves comparable performance. It is a subset of of a more general class of compression techniques, containing methods such as quantisation and knowledge distillation [13].

Approaches to pruning can vary in many ways. Some methods vary in how it determines whether to remove a component or not. Some methods will prune a network once it has been trained, others will prune a network before training has even begun. This section will not provide a comprehensive overview of pruning methods but it will aim

[1] The expectation of the estimate is equal to the value it is estimating.

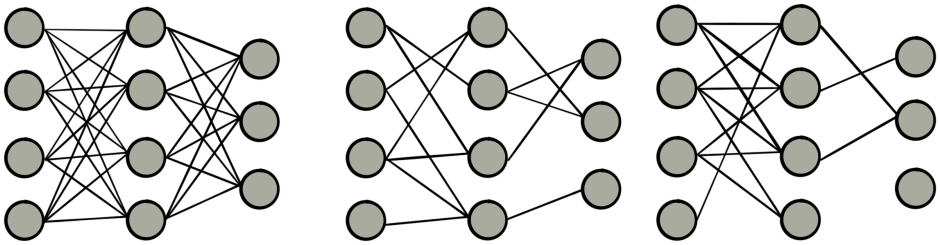


Figure 2.2: Pruning a network (left) via local (middle) and global (right) pruning. Global pruning will result in different amount sparsities in each layer.

to describe what comprises a pruning approach, alongside common techniques in each component. For the interested reader, a large amount of literature surrounds network pruning as it is a technique that has been in use for decades and remains subject to new developments [14, 15, 16].

2.2.1 Pruning Structure

A pruning method may be structured or unstructured. An unstructured approach will remove the most basic building blocks of a network - the neurons themselves. This approach is the most flexible as the rate of sparsity may be achieved via any combination of neuron removal.

Structured pruning treats entire modules of a network as a unit, either keep or removing whole structures at a time. It is easier to benefit from the improved computational efficiency of unstructured pruning as the sparse matrices produced by unstructured pruning require a specific implementation to streamline the computation [17]. This often leads to the structured pruning method producing networks that are much more computationally efficient and light on storage. However, the flexibility of the unstructured method is lost in the process.

2.2.2 Pruning Criteria

The decision of whether to prune the unit in question is down to the chosen criteria. Common examples of criteria involve scoring by the magnitude of the weight as small magnitude weights are thought to contribute very little to the model computation. Other criteria consider the combination of the magnitude of the weight and its sensitivity. More sensitive weights will likely cause large changes in loss whereas the removal of small, more stable weights should alter very little in the network's performance.

The criteria may be considered on a local or global level. A global level will simply score each unit, without comparison to its neighbours, and prune the network to the given sparsity level. A local pruning method will prune certain sections to the given sparsity level e.g. on a per layer basis. This will reduce the chances of *layer collapse* - a result where whole layers are pruned away, causing incompatibilities of dimension within the network and rendering it un-trainable.

2.2.3 Pruning Algorithms

Most pruning algorithms can be decomposed into distinct stages: training, pruning, and fine-tuning. Training is what the name suggests - the model is trained as normal to achieve good performance. In the pruning stage, the components of the model are assigned scores based on the criteria and the components with the lowest scores are removed. The actual pruning itself can take place over a single or perhaps multiple iterations, producing proportion of the desired sparsity at each loop. The final stage of fine-tuning continues to train the model for a short period to allow it to regain some of the performance lost from the removal of components.

This is not a rigid framework as many popular pruning methods will deviate from this structure. For example, in their paper introducing the Lottery Ticket Hypothesis, Frankle and Carbin ([18]) develop iterative magnitude pruning. In this method, the pruning process consists of two stages: the training stage - where the model is trained from a specified initialisation for a certain number of epochs - and the pruning phase - where the lowest $\frac{p}{n}\%$ of weight magnitudes are removed from the network. Here, p is the desired sparsity of the final network and n is the number of times the two stages will be iterated for. When the training stage is reached again, the weights of the network are wound back to the very initial parameters before the training restarts.

Algorithm 1 Iterative Magnitude Pruning

procedure ITERATIVE MAGNITUDE PRUNING(Model f, model parameters θ, pruning rate p)
  $\widehat{\theta} \leftarrow \theta$ ▷ Save the initial model parameters
  $m[i] \leftarrow 1$ ▷ Initialise mask to 1s.
  for i in $1 \cdots$ n **do**
    $m \leftarrow \text{PRUNE}(\theta, m, \frac{p}{n})$ ▷ Update the pruning mask.
    $\theta \leftarrow \text{TRAIN}(f, \widehat{\theta} \odot m, \mathcal{X}, \mathcal{Y})$ ▷ Apply mask to initial parameters and retrain.
  end for
end procedure
procedure PRUNE($\theta, m, \frac{p}{n}$)
  for i in $1 \cdots |\theta|$ **do**
    if $m[i] = 1$ and $|\theta[i]| \in$ smallest $\frac{p}{n}\%$ of unpruned weight magnitudes **then**
      $m[i] \leftarrow 0$
    end if
  end for
  return m
end procedure

2.3 Topological Data Analysis

Topology is the branch of mathematics that studies the properties of spaces that remain invariant under continuous deformation. For example, if a space contains a 'hole', then no amount of stretching or twisting the space will remove the existence of that hole. Algebraic topology is the field that studies algebraic structures, such as groups or modules,

that represent the topological features of the spaces in question. We can compare spaces by their topological properties either by maps between the spaces themselves or maps between the algebraic structures, giving rise to notions of equivalence between spaces. One such notion is homological equivalence, where spaces that have the same structure of holes are considered to be the 'same'. This leads to the quintessential example from topology: the doughnut being equivalent to the coffee cup.

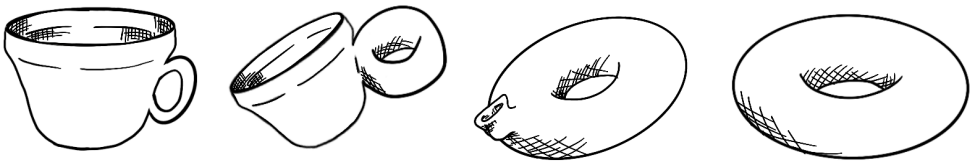


Figure 2.3: Morphing a cup into doughnut.

The application of these topological ideas to datasets is the field of topological data analysis; a field that applies computational topology to problems in statistics and data science. The inclusion of topological ideas to data analysis has proven beneficial in a variety of fields; from shape analysis [19] to medical imaging [20], biology [21] to sensor networks [22]. It has even been used to analyse the complexity of the structure of a neural network itself [23].

One of the main methods used in Topological Data Analysis is that of persistent homology. The general idea is that we construct a nested sequence of objects called simplicial complexes on top of the point cloud of the data. We then compute the n^{th}-dimensional homology group of each simplicial complex at each step of the sequence.

All content from this section has been taken from sources [24] and [25]. Relevant content is included here for completeness and the reader's convenience but for a more comprehensive overview, interested readers should refer to the original sources.

2.3.1 Simplicial Complexes

Simplicial complexes are a key component in simplicial homology - the computationally tractable sibling of singular homology. Singular homology is difficult to compute as the groups that represent the 'holes' of the space are often uncountable. Simplicial homology is much easier to compute and is equivalent to singular homology when the *triangulation* of a space is homeomorphic[2] to a simplicial complex. Therefore, if the homology of the simplicial complex can be computed, the homology of the original space can also be computed. This is why topological data analysis (and many other branches of computational topology) mainly concerns itself with simplicial complexes.

Definition 1 (k-simplex)**.** *For $k \geq 0$, a k-simplex (σ) in a Euclidean space $\mathbb{R}^m$ is the convex hull of a set P of $k+1$ affinely independent points in $\mathbb{R}^m$. A face of a k-simplex is the convex hull of any subset of points from P.*

[2]Two topological spaces X, Y are homeomorphic if there exists continuous invertible maps between them.

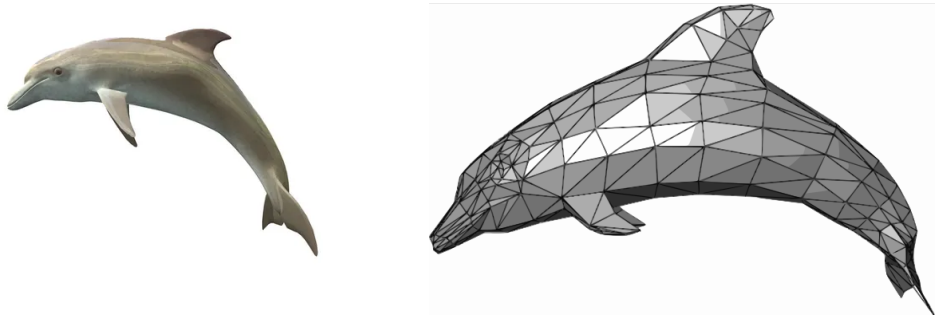


Figure 2.4: Triangulation of a dolphin.[3]

Remark 1. *Affine independence of an affine space is the analogue to linear independence of a vector space. Affinely independent points are linearly independent but in a way that is agnostic to the origin. If we take a set of linearly independent points in a vector space and perform an affine transformation on them (translation), the resulting vectors would be affinely independent but possibly not linearly independent.*

Definition 2 (Simplicial Complex)**.** *A simplicial complex, K, is a set containing finitely many simplices that satisfy the following restrictions.*

- *K contains every face of each simplex in K.*
- *For any two simplices $\sigma, \tau \in K$, their intersection is either empty or a face of both σ and τ.*

We may also define a simplicial complex independent of geometry. A collection, K, of non-empty subsets of a given set $V(K)$ is an abstract simplicial complex if every element $\sigma \in K$ has all of its non-empty subsets $\sigma' \subseteq \sigma$ also in K.

Remark 2. *The above definition contain the definitions for both a geometric and an abstract simplicial complex. The two notions are equivalent as an example of one may rendered as an example of the other. Therefore, the two definitions have here been compressed into one.*

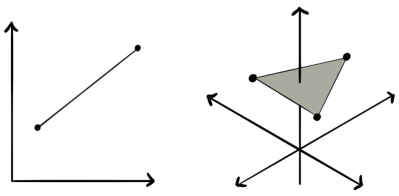


Figure 2.5: Example of a 1-simplex (left) and a 2-simplex (right). Both contain 0-simplices (verticies).

Example 1. *Let $V(K) = \{1,2,3,4,5\}$ and let $K = \mathcal{P}(\{1,2,3\}) \cup \mathcal{P}(\{3,4,5\}) \cup \{2,4\}$, where $\mathcal{P}(X)$ denotes the power set of X. This abstract simplicial complex may be realised geometrically, as shown in Figure 2.6.*

Example 2. *Let $V(K) = \{1,2,3\}$ and consider $K = \{\{1\},\{2\},\{1,2\},\{2,3\}\}$. $\sigma = \{2,3\} \in K$ but $\sigma' = \{3\} \subset \sigma$ is not in the set, therefore K is not a simplicial complex.*

2.3.2 Homology

Homology is a tool in topology that connects algebraic structures to features in a topological space - these features being the p^{th}-dimensional holes of the space. Cycles and boundaries are important notions to define what a hole is: a cycle that is not the boundary of a lower dimensional simplex. These notions can be extended into algebraic structures to give us the p-dimensional homology groups.

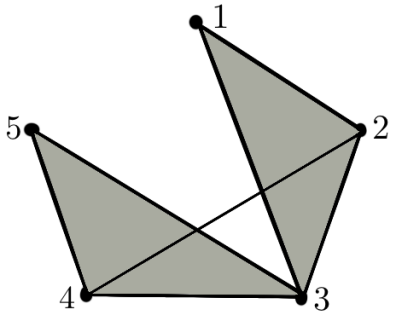


Figure 2.6: Realisation of an abstract simplicial complex.

Definition 3 (p-Chain)**.** *Let K be a simplicial complex with m_p p-simplices. And let R be a ring. A p-chain is a linear combination of p-simplicies with coefficients in R,*

$$c = \sum_{i=1}^{m_p} \alpha_i \sigma_i \quad \text{for } \alpha_i \in R \text{ and } p\text{-simplex } \sigma_i.$$

Under addition, p-chains form an R-module. The addition operator is given by

$$c + c' = \sum_{i=1}^{m_p} (\alpha_i + \alpha_i')\sigma_i.$$

Definition 4 (p^{th} Chain Group)**.** *The set of p-chains form a group under the addition of p-chains, $C_p(K)$, called the p^{th} chain group.*

Remark 3. *In particular,* ***we will consider the case where*** *$R = \mathbb{Z}_2$. The identity is the zero chain, 0, and inverse element of a chain, c, is c itself.*

Definition 5 (Boundary Operator)**.** *Let $\partial_p : C_p(K) \to C_{p-1}(K)$ denote a homomorphism between chain groups, defined by*

$$\partial_p c = \sum_{j=0} (-1)^j a_j (\partial_p \sigma_j) \quad \text{for } c = \sum_{j=0} a_j \sigma_j \in C_p(K),$$

where $\partial_p \sigma = \sum_i \sigma \setminus \{v_i\}$ and $\partial_0 \sigma = \varnothing$. ∂_p is the boundary operator.

Example 3. *Let K be the simplicial complex as given by Figure 2.7 and let $c = \{\{1,2,3\}\}$ be a 2-simplex. A As we are only considering the case where $R = \mathbb{Z}_2$, application of the boundary operator on c gives,*

$$\begin{aligned} \partial_2 c &= -\{2,3\} + \{1,3\} - \{1,2\} \\ &= \{2,3\} + \{1,3\} + \{1,2\} \end{aligned} \tag{2.2}$$

i.e. the collection of 1-simplices that form the boundary of c. Further application of ∂ gives,

$$\begin{aligned} \partial_1 \circ \partial_2 (c) &= \partial_1(\{2,3\} + \{1,3\} + \{1,2\}) \\ &= \partial_1\{2,3\} + \partial_1\{1,3\} + \partial_1\{1,2\} \\ &= -\{2\} + \{3\} - \{1\} + \{3\} - \{1\} + \{2\} = 0 \end{aligned} \tag{2.3}$$

Definition 6 (Chain Complex)**.** *A chain complex, $(C(K),\partial)$, is a sequence formed by a sequence of spaces, C_p, together with boundary maps, ∂_p.*

$$0 = C_{k+1}(K) \xrightarrow{\partial_{k+1}} C_k(K) \xrightarrow{\partial_k} \cdots \xrightarrow{\partial_1} C_0(K) \xrightarrow{\partial_0} C_{-1}(K) = 0.$$

Note the important property of the boundary operator that $\partial_{p-1} \circ \partial_p(c) = 0$.

Definition 7 (p-Cycle)**.** *A p-chain, c, is a p-cycle if $\partial_p c = 0$.*

Definition 8 (Cycle Group)**.** *The collection of all p-cycles forms a group under the addition of p-chains called the p^{th} cycle group, Z_p. The p^{th} cycle group is the subgroup of C_p that forms the kernel of the boundary operator:* $\ker \partial_p = Z_p$.

Definition 9 (Boundary Group)**.** *The p^{th} boundary group is the subgroup of C_{p-1} that is obtained from the image of the boundary operator on C_p: $B_{p-1} = im\ \partial_p$.*

Definition 10 (Homology Group)**.** *For $p \geq 0$, the p^{th} homology group is the quotient group $H_p = Z_p/B_p$. The dimension of H_p is called the p^{th} Betti number,*

$$\beta_p = \dim H_p.$$

The p^{th} Betti number refers to the number of p-dimensional holes in the simplicial complex. Note that we are able to take the quotient Z_p/B_p as B_p is a normal subgroup of Z_p via the property of $\partial_{p-1} \circ \partial_p(c) = 0$.

Remark 4. *The elements of the p^{th} homology group are the p-cycles that do not form the boundary of a $p+1$-simplex.*

2.3.3 Persistent Homology

It is not immediately obvious how notions of homology would be applicable in a data analysis context. After all, it is incredibly unlikely that the data will be given to us in the form of a simplicial complex. Therefore, we need a way to interpret our data as such. Here, we assume that our data is a point cloud, and not an object such as a graph, which is naturally a simplicial complex.

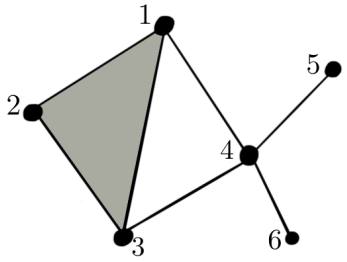


Figure 2.7: The boundary of $\{1,2,3\}$ gives the cycle $\{1,2\}+\{2,3\}+\{1,3\}$.

Definition 11 (Nerve)**.** *Given a finite collection of sets $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$, we define the nerve of the set $\mathcal{U}$ to be the simplicial complex $N(\mathcal{U})$ whose vertex set is the index set, A, and where a subset $\{\alpha_0, \alpha_1, ..., \alpha_k\} \subseteq A$ spans a k-simplex in $N(\mathcal{U})$ if and only if*

$$\bigcap_{i=0}^{k} U_{\alpha_i} \neq \varnothing.$$

Definition 12 (Čech Complex)**.** *Let (M,d) be a metric space and let P be a finite subset of M. Given a real $r \geq 0$, the Čech complex is defined to be the nerve of the set $\{B(p_i,r)\}$, where $B(p_i,r)$ is the geodesic closed ball of radius r, centered at p_i.*

Definition 13 (Vietoris-Rips Complex)**.** *Let (P,d) be a finite metric space. Given real $r \geq 0$, the Vietoris-Rips complex is the abstract simplicial complex $\mathbb{VR}^r(P)$ where a simplex $\sigma \in \mathbb{VR}^r(P)$ if and only if $d(p,q) \leq 2r$ for all vertex pairs p,q in σ.*

It should be noted that there are many other ways to convert a point cloud (finite metric space) to a simplicial complex. The Čech and Vietoris-Rips complex are among those most commonly used in topological data analysis but, depending on the usage, other complexes may be more suitable.

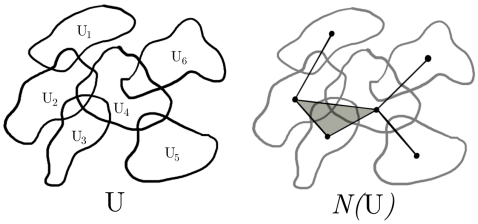


Figure 2.8: A set, U, and its corresponding nerve, $N(U)$.

The conversion of data to a simplicial complex will be dependent on some parameter. For the Čech and Vietoris-Rips complexes, this parameter is r - the radius that determines whether vertices are considered to be apart of the same complex or not. It will likely be an impossible task to know which r is the correct one. That is, which value of r gives the correct structure of the underlying space from which the data has been sampled. The solution to this predicament comes from the *persistence* portion of persistent homology. Instead of concerning ourselves with which parameter is the correct one, we consider *all* of them and consider the homological features that are prominent throughout all the choices of parameter to be the ones inherited from the underlying space.

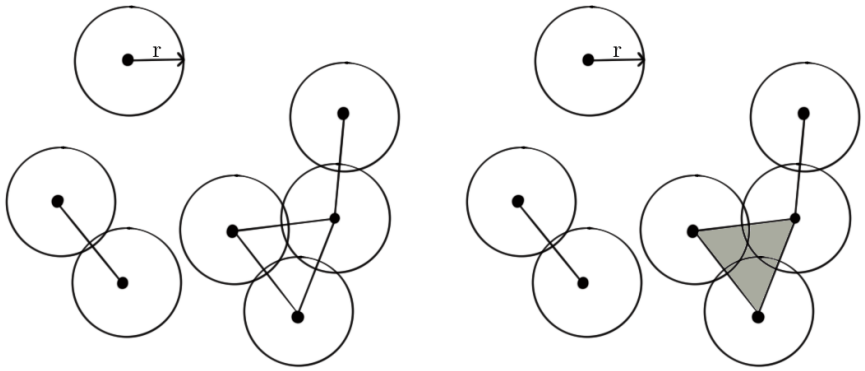


Figure 2.9: Čech Complex (left) and Vietoris-Rips Complex (right) of a point set, P.

Definition 14 (Simplicial Filtration)**.** *A filtration, $\mathcal{F}$, of a simplicial complex, K, is a nested sequence of its subcomplexes,*

$$\mathcal{F} : \varnothing \subseteq K_0 \subseteq K_1 \subseteq \ldots \subseteq K_n = K.$$

Definition 15 (Filtration Function)**.** *If a simplicial filtration, $\mathcal{F}$, is obtained from a function, $f : K \to \mathbb{R}$ then $\mathcal{F}$ is induced by f. If the simplicial filtration is given without*

an explicit input function, then $\mathcal{F}$ is induced by the simplex-wise monotone function $f : K \to \mathbb{R}$, $\sigma \mapsto i$ for $\sigma \in K_i \setminus K_{i-1}$ such that for every $\sigma' \subseteq \sigma$, $f(\sigma') \leq f(\sigma)$.

Definition 16 (Persistence Module)**.** *A simplicial filtration, $\mathcal{F}$, of a simplicial complex, K, induces a homomorphism induced from the inclusion map between subcomplexes.*

$$h_p^{i,j} = \imath_* : H_p(K_i) \to H_p(K_j),$$

for $p \geq 0$ and $0 \leq i < j \leq n$. The sequence of such induced homomorphisms form a persistence module,

$$0 = H_p(K_0) \to H_p(K_1) \to \ldots \to H_p(K_n).$$

Definition 17 (Persistent Homology Groups)**.** *The p^{th} persistent homology groups are given by the images of the homomorphisms, $H_p^{i,j} = \text{im } h_p^{i,j}$.*

Definition 18 (Persistent Betti Numbers)**.** *The p^{th} persistent Betti numbers are the dimensions of the corresponding persistent homology groups, $\beta_p^{i,j} = \dim H_p^{i,j}$.*

Remark 5. *The elements of the p^{th} persistent homology groups consist consist of the homology classes that persist from K_i to K_j.*

Definition 19 (Birth and Death)**.** *A p^{th} homology class, $\xi \in H_p(K_a)$, is born at K_i for $i \leq a$ if $\xi \in H_p^{i,a}$ but $\xi \notin H_p^{i-1,a}$. The class dies at K_j for $a < j$ if $h_p^{a,j-1}(\xi)$ is non-trivial but $h_p^{a,j} = 0$.*

Definition 20 (Persistence Pairing Function)**.** *For $0 < i < j \leq n+1$, let*

$$\mu_p^{i,j} = (\beta_p^{i,j-1} - \beta_p^{i,j}) - (\beta_p^{i-1,j-1} - \beta_p^{i-1,j}).$$

$\mu_p^{i,j}$ therefore counts the number of classes that are born at K_i and die at K_j. Should $j = n+1$, we take this to mean that the classes are alive until the end of the filtration and so set $n+1$ to ∞.

Definition 21 (Class Persistence)**.** *For $\mu_p^{i,j} \neq 0$, the persistence $Pers([c])$ of a class $[c]$ that is born at K_i and dies at K_j if given as $Pers([c]) = a_j - a_i$, where $f(K_i) = a_i$.*

Definition 22 (Persistence Diagram)**.** *A persistence diagram, $Dgm_p(\mathcal{F}_f)$ of a filtration, $\mathcal{F}$, induced by f is given by the set of points on the extended plane,*

$$\text{Dgm}_p(\mathcal{F}_f) = \{(a_i, a_j) | \mu_p^{i,j} \neq 0 \text{ and } i < j\} \subseteq \bar{\mathbb{R}}^2.$$

The points on the diagonal, $\Delta = \{(a,a)\}$, are added with infinite multiplicity, $\mu_p^{i,i} = \infty$.

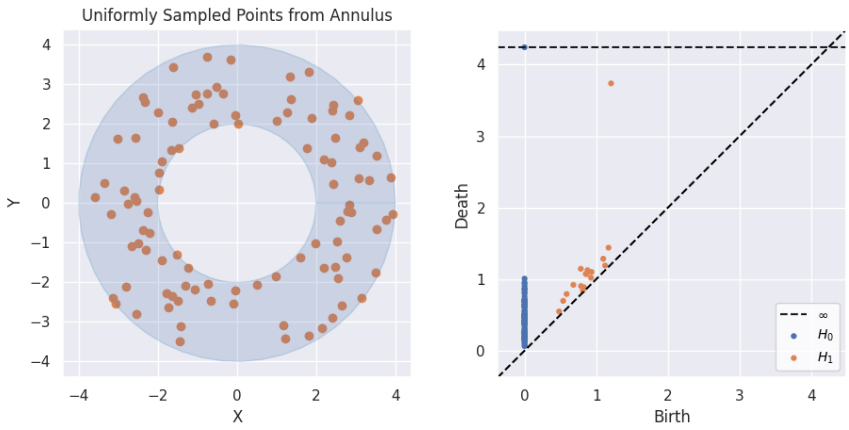


Figure 2.10: Points sampled from an annulus and the corresponding persistence diagram. The point cloud has a connected component that lasts for the whole filtration and a hole that has a large persistence.

2.4 Fractal Dimension

Typical notions of dimension are usually insufficient to describe fractal structures in a meaningful way. For example, a space-filling curve (such as a Hilbert Curve) would be considered a one-dimensional object under the usual definition of dimensions - the same dimension as that of a straight line or a sine wave. Using a fractal dimension, such as the Hausdorff dimension, a Hilbert Curve would have a dimension of $\frac{3}{2}$ and the straight line a dimension of 1. This aligns more closely with the intuition that a Hilbert Curve would take up more space and hence be a higher dimensional object than a line.

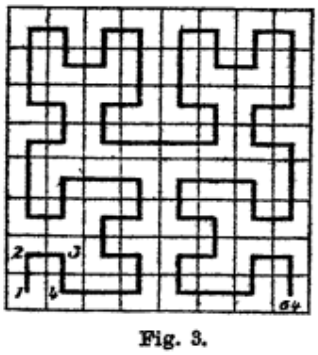


Figure 2.11: An example of a Hilbert Curve, given in Hilbert's original paper [26].

The following explanations of Hausdorff and Box-Counting dimensions are given from [27].

2.4.1 Hausdorff Dimension

Definition 23 (δ-Cover)**.** *Let F be a subset of $\mathbb{R}^n$, $\delta > 0$, and $\{U_i\}$ be a countable collection of of subsets of $\mathbb{R}^n$ where the diameter of U_i is at most δ and F is covered by*

the collection - that is, $F \subset \cup U_i$. Then $\{U_i\}$ is a δ-cover of F.

Definition 24 (s-dimensional Hausdorff Measure)**.** *Let F be a subset of $\mathbb{R}^n$ and $s \geq 0$. We define the s-dimensional Hausdorff measure as*

$$\mathcal{H}^s(F) = \lim_{\delta \to 0} \mathcal{H}^s_\delta(F) = \lim_{\delta \to 0} \left(\inf\{ \sum |U_i|^s \, | \{U_i\} \text{is a } \delta\text{-cover of } F \} \right).$$

Definition 25 (Hausdorff Dimension)**.** *Left F be a subset of $\mathbb{R}^n$. The Hausdorff dimension of F is given by*

$$\dim_H F = \inf\{s \geq 0 | \mathcal{H}^s(F) = 0\} = \sup\{s | \mathcal{H}^s(F) = \infty\}.$$

Typically, $\mathcal{H}^s(F)$ tends to be either 0 or ∞, therefore $\dim_H F$ is the point at which $\mathcal{H}^s(F)$ drops from ∞ to 0.

2.4.2 Box-Counting Dimension

Definition 26 (Box-Counting Dimension)**.** *Assuming the following limit exists, the box-counting dimension of $F \subset \mathbb{R}^n$ is given by*

$$\dim_B F = \lim_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta},$$

where N_δ is the smallest number of closed δ balls that cover F.

Where F is a bounded set of $\mathbb{R}^n$, the Hausdorff and the Box-Counting dimension agree. That is,

$$\dim_B F = \dim_H F.$$

2.4.3 Persistent Homology Dimension

Definition 27 (Persistence Lifetime)**.** *The lifetime of a p^{th} homology class, ξ, is its class persistence as given in Definition 21.*

$$|I(\xi)| = a_j - a_i,$$

where ξ is born at K_i and dies at K_j and the filtration, $\mathcal{F}$ is induced by the function, f.

Definition 28 (α-Weighted Lifetime Sum)**.** *For a finite set $W \subset \mathcal{W} \subset \mathbb{R}^d$, the weighted i^{th} homology lifetime sum is defined as*

$$E^i_\alpha(W) = \sum_{\xi \in \mathrm{PH}_i(\mathrm{VR}(W))} |I(\xi)|^\alpha,$$

where $\mathrm{PH}_i(\mathrm{VR}(W))$ is the i^{th} dimensional persistent homology of the Vietoris Rips complex on a finite point set, W.

Definition 29 (Persistent Homology Dimension)**.** *The PH_i dimension of a bounded metric space, $\mathcal{W}$, is given as*

$$\dim^i_{PH} \mathcal{W} = \inf\{\alpha | \exists C > 0, \forall finite\ W \subset \mathcal{W}\ s.t\ E^i_\alpha(W) < C\}.$$

If W is taken to be a bounded set of $\mathbb{R}^d$, the persistent homology dimension coincides with the box-counting dimension and therefore the Hausdorff dimension. That is,

$$\dim^i_{PH} W = \dim_B W = \dim_H W. \tag{2.4}$$

Chapter 3

Persistent Homology Dimension Regularisation

Persistent homology dimension regularisation (Dim_{PH} regularisation) is a method that aims to reduce the intrinsic dimension of network trajectories during the training of a model via gradient-descent methods. The motivation for such a method stems from recent developments in understanding how the gradient noise of SGD can be linked to a model's generalisation properties. Before giving the definition of Dim_{PH} regularisation, we will first introduce these developments.

3.1 Stochastic Differential Equation Approximation of Stochastic Gradient Descent

Recall the update rule for SGD from Section 2.1.

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}\left(f\left(\{x_{i_j}\}_{j=1}^{j=b}; \theta_t\right), \{y_{i_j}\}_{j=1}^{j=b}\right), \tag{3.1}$$

for continuously differentiable loss function $\mathcal{L}$, randomly sampled batches, $\{x_{i_j}\}_{j=1}^{j=b}$, $\{y_{i_j}\}_{j=1}^{j=b}$, of observation-label pairs and learning rate η. The gradient noise is defined to be the difference $\varepsilon = \nabla(\mathcal{L} - \widehat{\mathcal{L}})$ which is the difference between the stochastic gradient and the true gradient of full-batch gradient descent.

To facilitate analysis of SGD, a common approach is to view it as the discretisation of a stochastic differential equation [28, 29]. In much of the literature, the assumption is made that the gradient noise has finite mean and variance and so allows for the application of the central limit theorem.

Theorem 1 (Central Limit Theorem)**.** *Let $X_1, X_2, \cdots, X_n$ be independent identically distributed (i.i.d) random variables with finite mean, μ and variance, σ^2. Then, as $n \to \infty$,*

$$\frac{(X_1 - \mu) + (X_2 - \mu) + \cdots + (X_n - \mu)}{\sigma\sqrt{n}},$$

converges in distribution to $\mathcal{N}(0,1)$.

If the assumption of finite variance in the gradient noise is made, then the central limit theorem can be applied to justify modelling the gradient noise as being drawn from a Gaussian distribution. The mean is 0 and the covariance is $\sigma^2\mathbb{I}$, where σ^2 denotes the noise variance in each component of $\nabla(\mathcal{L}-\widehat{\mathcal{L}})$. Equation 3.1 may then be written as

$$\begin{aligned}\theta_{t+1}-\theta_t &= -\eta\nabla\mathcal{L}\left(f\left(\mathcal{X};\theta\right)\mathcal{Y}\right)-\eta\nabla\left(\mathcal{L}\left(f\left(x;\theta\right)y\right)-\mathcal{L}\left(f\left(\mathcal{X};\theta\right)\mathcal{Y}\right)\right)\\ &= -\eta\nabla\mathcal{L}\left(f\left(\mathcal{X};\theta\right)\mathcal{Y}\right)-\eta\sigma\varepsilon, \qquad \varepsilon\sim\mathcal{N}(0,\mathbb{I}).\end{aligned} \tag{3.2}$$

The form of this equation takes that of the approximation found by the Euler-Maruyama method [1] for the stochastic differential equation,

$$d\theta_t = -\nabla\mathcal{L}\left(f\left(\mathcal{X};\theta\right)\mathcal{Y}\right)dt+\sigma\sqrt{\eta}dW_t,$$

where W_t denotes standard Brownian motion. Equation 3.2 is a solution to the stochastic differential equation above, found by a particular numerical optimisation method.

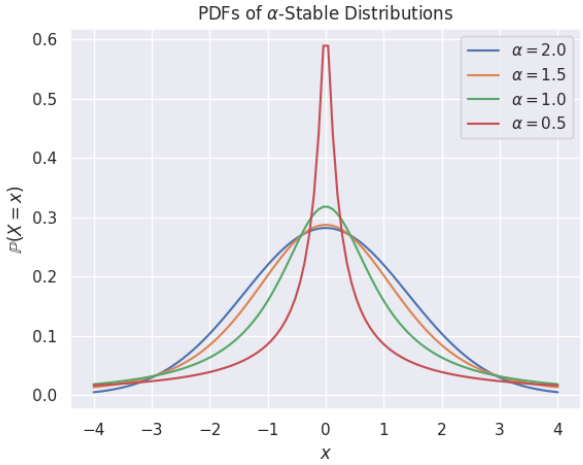


Figure 3.1: A comparison of the densities of α-stable distributions. As α decreases, the distribution is heavier tailed, where $\alpha = 2$ is a Gaussian distribution.

This approximation of SGD has seen much success in the literature [4, 28] however, it does have its shortcomings. There are some behaviours of SGD that this representation is not able to model, such as its tendency to converge within flat minima [29]. The reason for this problem is due to the assumption that the gradient noise can be modelled as being Gaussian distributed. Şimsekli et al ([8]) show that the gradient noise is more accurately modelled as being drawn from a heavy-tailed distribution. They derive the following SDE approximation of SGD that instead drives the process by an α-stable Lévy process,

[1] This is the stochastic extension of the Euler method - a numerical method to find solutions of ordinary differential equations. For more information, see [30].

$$d\theta_t = -\nabla \mathcal{L} dt + \eta^{\frac{\alpha-1}{\alpha}} \sigma dL_t^{\alpha}, \tag{3.3}$$

where L_t^{α} denotes the $|\theta|$-dimensional α-stable Lévy process with independent components, parameterised by the *tail index*, α.

For $\alpha = 2$, the α-stable Lévy process corresponds to Brownian motion. That is, $L_t^2 = W_t$. As α decreases, more of the probability mass is distributed to the tails and so the variance becomes infinite. This shifting of the mass in an α-stable distribution as *alpha* decreases can be seen in Figure 3.1. With this unbounded variance of the gradient noise, the differences in subsequent parameters exhibit discontinuous 'jumps'.

3.2 Persistent Homology Dimension Regularisation

By treating stochastic gradient descent as a continuous-time stochastic process, the recursion of 3.3 produces a fractal-like structure in the network trajectories [31]. Şimsekli et al [10] measure the complexity of this fractal structure via the Hausdorff dimension and bound the generalisation error of the resultant model by a term dependent on this measure. The reduction of the Hausdorff dimension coincides with the reduction of the tail index underlying SGD as the Hausdorff dimension of network trajectories almost surely is equal to the tail index [32].

Birdal et al [9] replace the Hausdorff dimension in the generalisation bound with the persistent homology dimension by the equality as given in 2.4 - $\dim_H = \dim_{PH}$. The computation of the persistent homology dimension is not done exactly; the estimation of $\dim_{PH}$ is computed based upon the following proposition.

Proposition 1. *Let $\mathcal{W} \subset \mathbb{R}^n$ be a bounded set and $\dim_{PH} \mathcal{W} = d_*$. For all $\varepsilon > 0$ and $\alpha \in (0, d^* + \varepsilon)$ there exists $D_{\alpha,\varepsilon}$ such that*

$$\forall n \in \mathbb{N}_{>0}, W_n = \{w_1, \cdots, w_n\} \subset \mathcal{W}, \quad E_{\alpha}^0(W_n) \leq D_{\alpha,\varepsilon} \cdot n^{1 - \frac{\alpha}{d^* + \varepsilon}}. \tag{3.4}$$

The above inequality then leads to the approximation of the persistent homology dimension as used in [9].

Corollary 1. *Let $\mathcal{W} \subset \mathbb{R}^n$ be a bounded set. Then,*

$$\dim_{PH} + \varepsilon = d^* + \varepsilon \leq \frac{\alpha}{1 - m},$$

for α, ε as defined in Proposition 1 and m such that m is the slope of the linear regression line of $\log E_{\alpha}^0(W_n)$ on $\log n$ for samples of n and W_n.

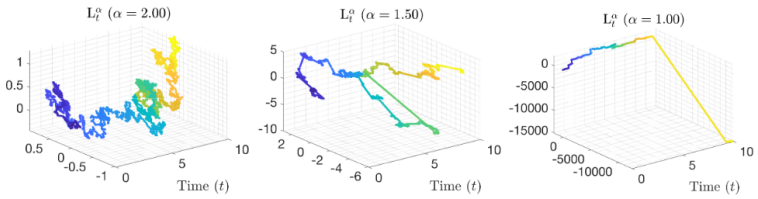


Figure 3.2: Sample paths for the α-stable process, L_t^α. As α decreases, the path taken becomes smoother.Figure taken from [10]

Algorithm 2 Estimation of Dim_{PH}

procedure $\text{DIM}_{PH}\text{ESTIMATION}(\mathcal{W} = \{w_i\}_{i=1}^K, n_{min}, \Delta, \alpha)$
  $n \leftarrow n_{min}, E \leftarrow [], X \leftarrow []$
  while $n \leq K$ **do**
    $W_n \subset \mathcal{W}$ ▷ Randomly sample n points from $\mathcal{W}$.
    $\text{dgm}(W_n) \leftarrow \text{VIETORISRIPS}(W_n)$ ▷ Persistence diagram computation.
    $E[i] \leftarrow \sum_{\gamma \in \text{dgm}(W_n)} |I(\gamma)|^\alpha$ ▷ α-weighted lifetime sums.
    $X[i] \leftarrow n$
    $n \leftarrow n + \Delta$
  end while
  $m, b = \text{LINEARREGRESSION}(\log X, \log E)$ ▷ Get slope and bias of regression line.
  return $\frac{\alpha}{1-m}$
end procedure

The actual estimation of Dim_{PH} as used in the paper is $\frac{\alpha}{1-m}$ as their empirical results show a small margin of error between d^* and $\frac{\alpha}{1-m}$. The algorithm for their computation is given in Algorithm 2. The computation of the estimate is composed of differentiable functions and so is able to be inserted into the loss function for an optimisation problem as a regularisation term. Experimental results from the paper confirm the success of using the persistent homology dimension as a regularisation term.

Informally, the regularisation of Dim_{PH} can be thought of as *smoothing* out the paths taken in SGD. As the persistent homology dimension is a fractal measure, it measures the intrinsic dimension of objects. In the update rule for SGD with Dim_{PH} regularisation, parameters would be updated so that they minimise the loss function with the step that is taken. The loss function also prioritises updates that do not deviate significantly from the current path of the process. This results in the *smoothness* of the path that is seen in Figure 3.2.

Chapter 4

Problem Formulation

The aim of this chapter is to provide the reader with an overview of the current literature that has influenced the formation of the question this dissertation intends to answer. The connection of the persistent homology dimension to problems in machine learning has so far only been considered by Birdal et al in their paper introducing Dim_{PH} regularisation. Other instances of the persistent homology dimension in literature are primarily focused on it mathematical properties [33, 34]. As the work surrounding Dim_{PH} regularisation is rather limited, we will often discuss our motivations for the topic of this dissertation in terms of the effect of heavy-tailed gradient noise in SGD. Our justification for which relies on the fact that the Hausdorff dimension of network trajectories almost surely equates to the tail index, α, of the process [32]. Recalling that a smaller tail index results in a heavier-tailed process and the equivalence of the Hausdorff and persistent homology dimension, theories that involve the reduction of α should also be applicable to the reduction of the persistent homology dimension.

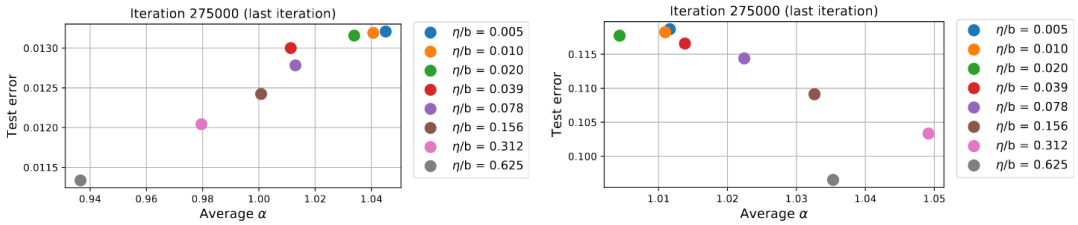
4.1 Generalisation and Dim_{PH} Regularisation

Here, we discuss what is currently known about the connection of the persistent homology dimension to generalisation and where lies the gaps that we seek to address.

In their paper introducing the persistent homology dimension to problems in machine learning, Birdal et al demonstrate the strong correlation of the dimension with generalisation. Subsequent experiments in the work regarding the implementation of Dim_{PH} regularisation show that the inclusion of this term resulted in both increased test accuracy and lower persistent homology dimension. The results of Birdal are promising for us with regards to our question surrounding the link of generalisation to commpression however, contradictory results in other papers reduces the certainty of these correlations being consistent.

On exploring the distribution properties of gradient noise in SGD, Simsekli et al [35] note that the correlation between the tail index and generalisation differs depending on the loss function. However, observations from other papers, some of which being subsequent works of the same authors, consistently demonstrate a positive correlation

between the tail index and generalisation. Although the experiment in question of Simsekli et al is rather limited in contrast to the many other works in this topic, it raises an interesting point that it may not always be the case that this correlation is observed. Therefore, there may be instances were improved generalisation does not result in lower persistent homology dimension, prompting us to investigate this relationship in our experiments.



(a) Linear hinge (b) NLL

Figure 4.1: The plot of results from [35] that show the opposing correlation of the tail index against both generalisation and increasing ratio of step size to batch size.

Noisy SGD and Flat Minima: There are other ways to induce heavy tails in SGD than controlling the Dim_{PH} of network trajectories. One such way is controlling the ratio of the learning rate to the batch size, $\frac{\eta}{b}$. Similarly to the persistent homology dimension, this ratio is strongly correlated with the generalisation error [36, 37]. Large ratios correspond with better performing models - an intriguing discovery as it contrasts with the notion that smaller step sizes and larger batches will cause the trajectory of SGD to more closely follow that of full-batch gradient descent.

It is widely believed that the improvements in generalisation induced by large ratios are due to the shape of minima that SGD converges to. Flatter, wider minima are known to have better generalisation properties than sharp minima [6]. For a long time, it has been known that SGD is biased towards such minima although the reasons for which were unknown. It is this bias that Simsekli use as a part of their justification for the heavy-tailed SDE approximation as described in Section 3.2. Heavy tails mean that the ability to escape from minima is not dependent on height but on width, meaning SGD is more likely to find itself trapped in a wider minimum[8].

As such benefits are able to be gained via one method of inducing heavy tails, it seems reasonable to ask whether controlling the noise via the persistent homology dimension will also see such benefits.

4.2 Compression and Generalisation

In this section we temporarily deviate from the discussion around the persistent homology dimension to explore how compression can be related to generalisation, providing context as to why compression forms a particular part of this dissertation.

Occam's Razor is a commonly used idea in machine learning when describing the generalisation performance of models. The idea is that simpler models are more likely

to capture the true underlying relationships of the data. If a model is compressible, then the model may be accurately described with a reduced amount of information therefore it can be considered 'simpler'. This then leads to the notion that the more compressible, 'simpler' models are likely to generalise better.

This idea of drawing a connection between the intrinsic dimension of model parameters and Occam's Razor is what motivates works that theoretically demonstrate a relationship between compressibility and generalisation. Suzuki et al [11] derive a method that allows for the compression-based bound of a compressed network to be applicable to the uncompressed one. Hsu et al [12] produce generalisation bounds for a model based on the bounds of a distilled network. These works both show that the generalisation of a model is strongly connected to its compression properties as improvements in the generalisation bounds of the compressed model may propagate to the bounds of the original model. This prompts us to consider when the generalisation ability of a model can be attributed to its compressibility.

4.3 Compression and Dim$_{PH}$ Regularisation

From this point onwards, we specifically consider compression in terms of network pruning as opposed to alternative compression methods like distillation. The potential connection of low persistent homology dimension and susceptibility to pruning methods occurs due to the induced statistical properties of heavy-tailed network trajectories. Heavy-tailed gradient noise has been shown to lead the resulting model to posses a number of properties commonly thought to be indicators of good compressibility.

Flat Minima: The pruning of network components will cause perturbations in the model weights. The attempt is made to reduce the effect of these perturbations by carefully considering which components to remove and by fine-tuning the model. However, if a model is very sensitive to changes in the weights then there is very little the network pruning method can do to recover the lost performance. Therefore, models that lie in the basin of sharp minima are unlikely to be as compressible as the solutions lying within flat minima of the loss surface.

We have mentioned how the ratio of step size to batch size is typically related to the shape of the minima converged to during SGD. The connection between the size of this ratio and the tail index of the gradient noise distribution suggests that it could be the case that the heavy tails induced by minimising the persistent homology dimension may also produce similar results in minima shape. However, as discussed in Section 4.1 it is possible that this correlation between the persistent homology dimension and the ratio is not consistent and so casts doubt on the assumption that reducing the persistent homology dimension inherits the same benefits as increasing the ratio of step size to batch size.

Sparsity: Another factor that benefits compression - particularly magnitude-based pruning - is the sparsity of the model parameters. Vectors are considered to be sparse when a the largest contributions to the norm of the vector are made by very few of the components. As described in Section 2.2, network pruning methods will remove components with the lowest criteria score first. With magnitude-based pruning, many

more weights can be pruned from the network without inducing too significant a performance reduction as sparse vectors have many entries with almost-zero magnitude.

The sparsity of a vector with components drawn from a heavy tailed distribution is dependent on the tail index. Recall from Section 3.2 that the tail index is the parameter that determines the distribution of an α-stable distribution. Heavier tails will produce sparser vectors as the peak around zero becomes sharper as the tail index decreases. Although we are considering that the gradient noise of SGD to be modelled by a heavy tailed distribution and not the weights themselves, the heavy tails of the gradient noise will also be present in the distribution of the weights [38, 39].

Therefore, we expect that models trained with Dim_{PH} regularisation will have sparser weights than those that were not.

Previous Work on Heavy Tailed SGD and Compression: A question similar to the one we are proposing was asked by Barsbey et al [36]. Their paper discussed the compressibility of overparameterised models with heavy-tailed noise that is induced by varing the ratio of step size to batch size. Encouragingly, their results demonstrate that it is the case that these models become more compressible as the ratio is increased. The approach of Barsbey et al differes from ours in two key ways. Firstly, we alter the tail index of the noise distribution via minimising the persistent homology dimension rather than through increasing the ratio of step size to batch size. It is possible that controlling the noise in these two ways both lead to similar results but it is not guaranteed. Secondly, we do not consider heavily overparameterised models whereas this is an important focus for their paper. The reason for which is based on the feasibility of training a large number of large models with Dim_{PH} regularisation - the regularisation requires the intensive computation of many Vietoris-Rips complexes.

4.4 Problem Statement Breakdown

Based upon our discussion of the potential relationship between reducing the persistent homology dimension and compression, we propose the following decomposition of the main question of this dissertation.

- How does the persistent homology dimension of network trajectories correlate with generalisation?
- What effect does the minimisation of the persistent homology dimension have on the distribution of model weights? Do they become more sparse? Are they situated in flatter minima?
- How is the compressibility of models via pruning related to the persistent homology dimension? Specifically, what type of correlation with compressibility do we find and are there other correlations compression-favourable features?

Chapter 5

Experiments

This chapter will detail the experiments we conducted to answer the question of whether the improvements in generalisation induced by regularising the persistent homology dimension can be attributed to the compressibility of the resulting model. The structure of these experiments are broken into sub-questions as described at the end of Chapter 4. For each of these sub-questions, we describe the setup of each experiment and give an evaluation of out results. The basic setup of the experiments is described as follows.

Datasets: All experiments are conducted on the MNIST [40] and CIFAR10 [41] datasets. MNIST is a dataset consisting of 70,000 examples of 28x28 images of grey-scale handwritten digits where each image is classified as a number 0 through 9. CIFAR10 is a dataset of 60,000 32x32 colour images, each one belonging to one of 10 classes. Both MNIST and CIFAR10 are commonly used benchmark datasets in machine learning research.

Architectures: All models trained are 5-layer (3 hidden layers) fully connected networks with 50 neurons in each hidden layer.

Training Algorithm: On MNIST, each model is trained for up to 100 epochs of stochastic gradient descent on 48,000 training examples (80% of the training set). For CIFAR10, we train for up to 200 epochs on 20,000 training examples. For both datasets, the learning rate is set to $\eta = 0.01$ and the objective of the training algorithm is to minimise the cross entropy loss plus any regularisation terms in use. We implement early stopping once the models have reached at least 99% training accuracy on MNIST and 70% accuracy on CIFAR10, evaluating the model on a hold-out validation set (12,000 for MNIST and 5,000 for CIFAR10). We do not aim to train to convergence on CIFAR10 as it would not be possible given our chosen architecture. The only other form of regularisation used in our experiments is Dim_{PH} regularisation - the computation of which is done via the code developed by Birdal et al in their paper introducing the method[1]. The library *torchph* is used for the persistent homology computations[2].

[1] https://github.com/tolgabirdal/PHDimGeneralization

[2] https://github.com/c-hofer/torchph

5.1 How Does the Persistent Homology Dimension of Network Trajectories Correlate With Generalisation?

To examine the relationship in question, we produce models that have varying persistent homology dimensions of network trajectories and compare their performance against their persistent homology dimension estimate. It is important to note that we do not produce this varying dimension by altering the ratio of step size to batch size as it is well-established that changing this value induces changes in the generalisation of the model. As we aim to evaluate what contribution the persistent homology dimension makes to generalisation, we enforce these changes in the dimension via the strength of regularisation, keeping the learning rate and batch size fixed. We select values of regularisation constant in the range $\{0.001,0.01,0.1,0.5,1,2,5\}$ alongside training the model without any regularisation. We repeat the training procedure across five random seeds for consistency.

When training models, we estimate the value of the tail index, $\widehat{\alpha}$, via the multivariate estimator developed in [42]. This estimation has favourable convergence properties and so has been used in a number of works [37, 8]. As the value of the tail index is equal to that of the persistent homology dimension of network trajectories, discussions of correlation with the persistent homology dimension will be made with regards to the tail index estimate. The method of estimating the tail index requires the configuration of a hyperparameter - the details of which we give in Appendix A.

We would expect to observe that the generalisation ability of the model would be positively correlated with the persistent homology dimension as these were the findings of [9]. We anticipate that the persistent homology dimension would decrease as the regularisation strength increased as it is the regularising term.

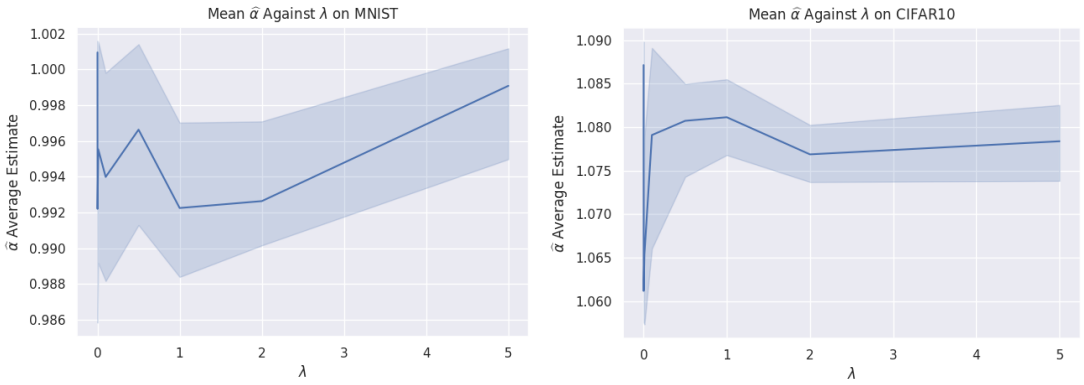
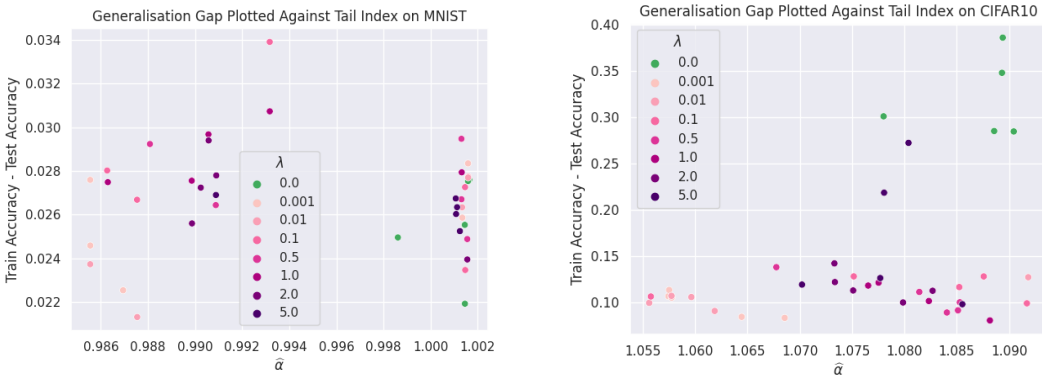


Figure 5.1: Plots of the tail index estimate against the strength of regularisation for both MNIST and CIFAR10. The shaded regions represent the 95% confidence interval computed over 5 runs.

The results of our training runs can be seen plotted in Figure 5.1. The trend in the tail index against the strength of Dim_{PH} regularisation is different to our hypothesis. For both datasets, the initial application of Dim_{PH} regularisation ($\lambda = 0.001$) produces downward spikes in the tail index estimate. As the strength continues to increase, both datasets see a 'hump' in the estimate before gradually increasing. The general trend of

the tail index seems to increase for regularisation strengths after $\lambda = 0.001$, although it is the case that the application of Dim_{PH} regularisation reduced the tail index.

With regards to the generalisation error and the tail index, we do not fully observe the correlation that was expected. The variables are positively correlated but the correlation is weak with a Pearson correlation coefficient of 0.0571 for MNIST and 0.3815 for CIFAR10. In the plots of Figure 5.2, the trend appears to be non-linear - particularly with MNIST. We therefore decide to compute the distance correlation coefficient [43]. The distance correlation gave much larger values of correlation, indicating that there is a relationship between these variables but it is non-linear.



Correlation Coefficient	MNIST	CIFAR10
Pearson	0.0571	0.3815
Distance	0.9429	0.6185

Figure 5.2: Trend of tail index estimates against regularisation constant used to train model alongside correlation coefficients that indicate the strength of relationship between these variables.

To answer the sub-question that this section focuses on, we observe a non-linear relationship between the persistent homology dimension and generalisation. Our results differ from those in the literature as we do not see a substantial improvement in generalisation as the tail index decreases.

5.2 What Effect Does the Minimisation of the Persistent Homology Dimension Have on the Distribution of Model Weights?

There are two main properties of model parameters that this section intends to analyse: the distribution of the weights and the flatness of the minima that SGD converges to. As the persistent homology dimension increases, we expect that the distribution of the weights becomes increasingly sparse and the shape of the minima in which SGD converges to become flatter. Our hypothesis derives from the explanations as given in Section 4.3.

To determine these properties, we compute three things. First, we plot kernel density estimates (KDE plots) of the model weights over each of the random seeds and regularisation strengths to visualise an estimate of the underlying distribution of the weights. Additionally, we measure the sparsity of each of the model weights by the use of the gini coefficient. Finally, we consider how changes in the persistent homology dimension affect the flatness of minima by using an approximation of the Hessian matrix.

The gini coefficient is, primarily, a measure of inequality that is often used in contexts of economics to quantify disparities in wealth within communities [44]. It takes values within the range of 0 and 1, where 0 denotes perfect equality in distribution and 1 denotes maximum inequality. Hurley and Rickard [45] conduct a review of different sparsity measures in which the gini coefficient was deemed to most accurately compute the sparsity. The calculation of this is detailed in Appendix [].

The results of the distribution of model parameters is interesting as we observe quite a clear trend in terms of regularisation strength. Around the mean of MNIST models, the weaker the regularisation strength, the more peaked around the mean the density plot is. This means that weaker regularisation for MNIST results in more model weights being grouped around zero and are therefore sparser. In the tails of the distribution, we observe the opposite. The stronger regularised models have more weights that have more extreme values. This behaviour is easily noted in Figure 5.3.

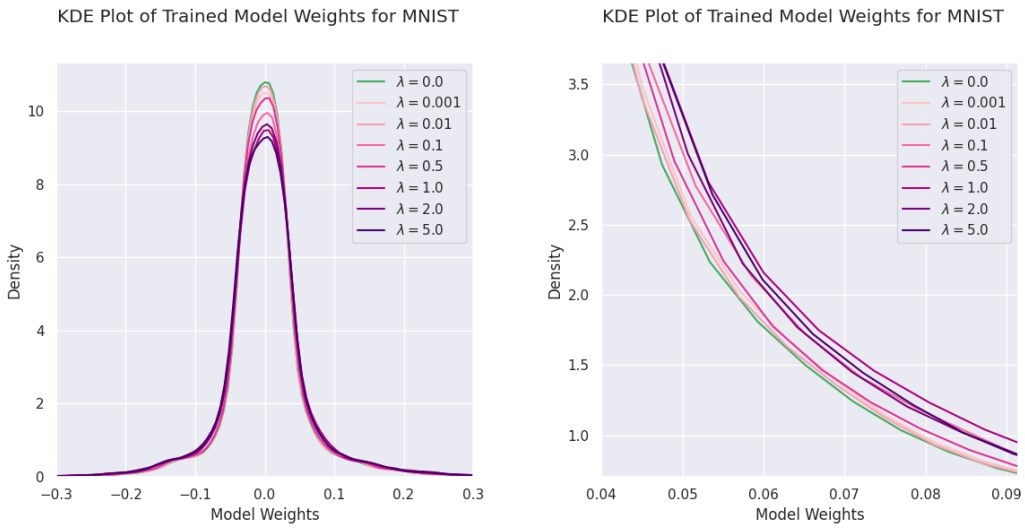


Figure 5.3: KDE plots of model weights for a training run on MNIST. The order of which strength of regularisation dominates is flipped at the mean and in the tails.

For CIFAR10, the weights of the regularised models behave in a similar way to what is observed in MNIST. The difference comes in when consider the KDE of the unregularised model - very low in comparison to the other models at the mean and completely dominates the other models in the tails. The unregularised model is typically the least sparse model but sparsity continues to deplete as the regularisation strength increases.

The plots of the gini coefficient against regularisation strength in Figure 5.5 resemble closely the trends that appeared in the KDE plots of our model weights. For MNIST, as

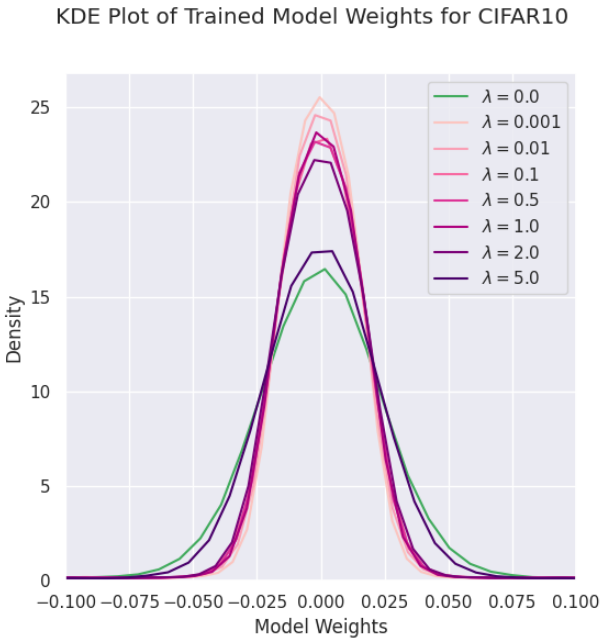


Figure 5.4: KDE plot of model weights for a training run on CIFAR10

the strength of regularisation increases, the gini coefficient reduces and so is becoming less sparse. For CIFAR10, we observe the same pattern as MNIST after $\lambda = 0.001$ but also see the sharp jump made at zero where the weight distribution suddenly increases in sparsity.

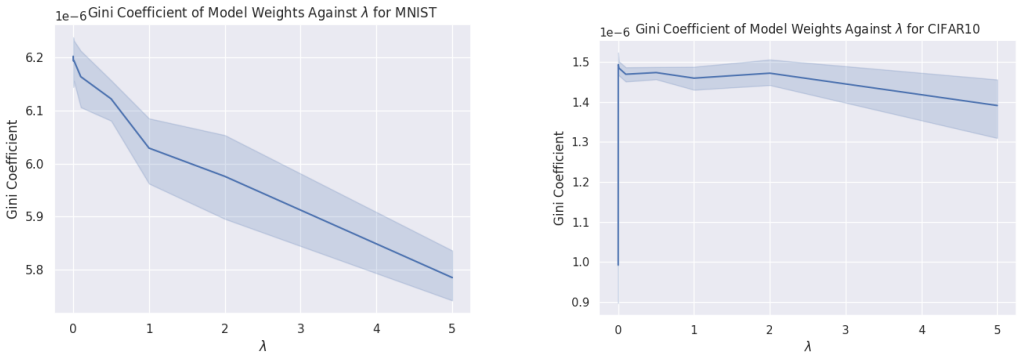


Figure 5.5: Gini coefficient of model weights plotted against regularisation strength.

In order to evaluate the flatness of minima, we turn to the Hessian matrix, H.

$$H = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial \theta_1^2} & \frac{\partial^2 \mathcal{L}}{\partial \theta_1 \theta_2} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial \theta_1 \theta_d} \\ \frac{\partial^2 \mathcal{L}}{\partial \theta_2 \theta_1} & \frac{\partial^2 \mathcal{L}}{\partial \theta_2^2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathcal{L}}{\partial \theta_d \theta_1} & \cdots & \cdots & \frac{\partial^2 \mathcal{L}}{\partial \theta_d^2} \end{bmatrix}$$

The eigenvalues of the Hessian measure the curvature of the loss on which it's defined, $\mathcal{L}$. A commonly used flatness measure considers the trace of the Hessian which is the sum of its eigenvalues[46]. The size of the Hessian usually means that the eigenvalue computation is untractable. As direct computation of the Hessian is not feasible, we have to approximate it. There are multiple approaches to approximating the Hessian but make use of one as described in [29]. When near an optimum, we can approximate the Hessian of the loss as the covariance of the gradients which itself may be estimated by the sample covariance. That is,

$$H \approx C \approx \frac{1}{N}\sum^{n} \nabla\mathcal{L}(f(x_i;\theta),y_i)\nabla\mathcal{L}(f(x_n;\theta),y_n)^T.$$

where H, C are the Hessian and covariance matrices, x_i, y_i are randomly sampled datapoints and N denotes the number of samples made. To measure the flatness of the minima around the optima of our models find, we compute the trace of this approximation of the Hessian.

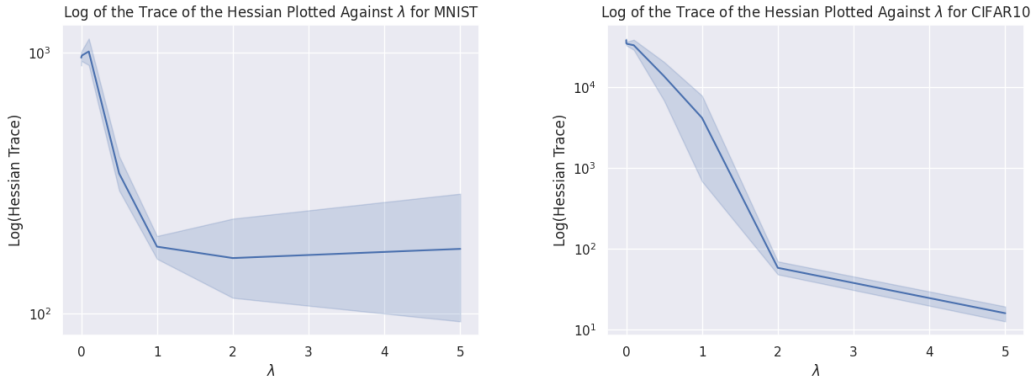


Figure 5.6: Regularisation strength plotted against the log of the trace of the Hessian. In both instances, increased λ leads to wider minima.

Plots of the Hessian estimate in Figure 5.6 demonstrate a fairly consistent decrease as the regularisation strength increases for both MNIST and CIFAR10. It is surprising that we have both that λ is correlated with minima width but not also a strong correlation between λ and the generalisation of the resultant model. As has been discussed, the flatness of minima has been quite well-established to correlate with improving the generalisation of models.

Plots of both the gini coefficient and the Hessian trace against the tail index do not show any kind of trend as distinct as those in the plots above. This is a surprising result as we have established that continuing to increase the regularisation strength coincides with the tail index increasing, making the SGD distribution less heavy tailed. This means that flatter minima are able to be found despite the increase in the tail index. The plots are given in Figure 5.7

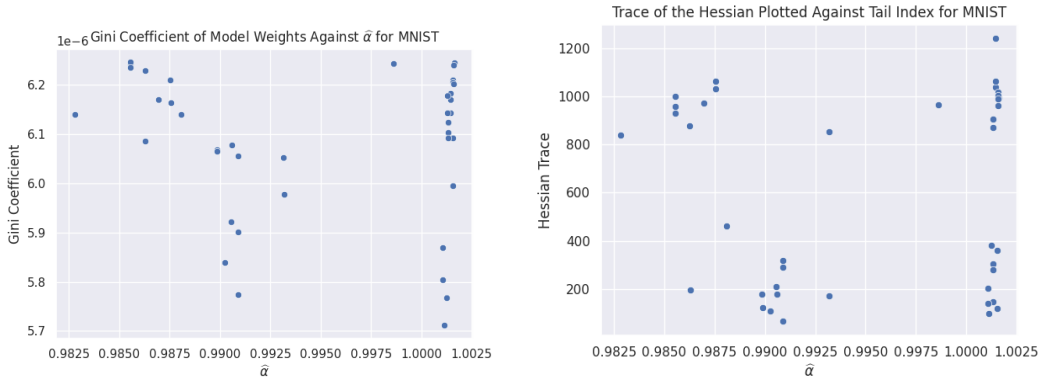


Figure 5.7: Both the gini coefficient and approximation of the Hessian Trace against the tail index.

5.3 How is the Compressibility of Models via Pruning Related to the Persistent Homology Dimension?

Finally, we explore the effect of the persistent homology dimension on the compressibility of networks. In our approach, we apply neural network pruning to a collection of models, each trained with varying strengths of Dim_{PH} regularisation. We prune to a wide variety of weights and across multiple seeds to achieve consistency in results. We prune across two types of pruning methods - iterative magnitude pruning and iterative random pruning. Both of these methods will be implemented globally and unstructured as global pruning tends to give better performing networks than local pruning and unstructured pruning as we do not intend to take away whole layers. In particular, we prune values in the range [0.1 - 0.995].

The algorithm of iterative magnitude pruning was introduced in Section 2.2. Iterative random pruning follows a similiar structure, with the exception that it does not make use of a criteria to prune weights but rather selects them at random. The algorithm for this method is given in Algorithm 3.

Algorithm 3 Iterative Random Pruning

procedure ITERATIVE RANDOM PRUNING(Model f, model parameters θ, pruning rate p)
  $\hat{\theta} \leftarrow \theta$ ▷ Save the initial model parameters
  $m[i] \leftarrow 1$ ▷ Initialise mask to 1s.
  for i in $1 \cdots$ n **do**
    $m \leftarrow$ randomly set $\frac{p}{n}\%$ of entries to 0 ▷ Update the pruning mask.
    $\theta \leftarrow \text{TRAIN}(f, \hat{\theta} \odot m, \mathcal{X}, \mathcal{Y})$ ▷ Apply mask to initial parameters and retrain.
  end for
end procedure

To facilitate a direct comparison of the prunability of networks, we devise a simple measure of compressibility.

Definition 30 (k-Sparse Approximation)**.** *A vector, $\boldsymbol{x}$, is a k-sparse approximation of $\boldsymbol{y}$, if $\boldsymbol{x} = \text{argmin}_{\|\tilde{\boldsymbol{x}}\|_0=k} \|\boldsymbol{y} - \tilde{\boldsymbol{x}}\|_2$.*

Definition 31 ($(k,\varepsilon) - Compressible$)**.** *A model f parameterised by vector θ and a loss function, $\mathcal{L}$, is said to be $(k,\varepsilon) - compressible$ with respect to $\mathcal{L}$ if ϕ is the k-sparse approximation of θ, and $\mathcal{L}(f(\mathcal{X};\phi),\mathcal{Y}) - \mathcal{L}(f(\mathcal{X};\theta),\mathcal{Y}) \geq \varepsilon$, for $\varepsilon \in \mathbb{R}$, observations $\mathcal{X}$ and labels $\mathcal{Y}$.*

Definition 32 (Average Compressibility)**.** *Let f be a model parameterised by θ and trained according to a loss function, $\mathcal{L}$. Let $\{k_i\} \subset \mathbb{N}$ be a set of size n and bounded by $|\theta|$. Then the average compressibility of f is defined as,*

$$\xi_{f,\theta} = \frac{1}{n} \sum_{k_i=k_1}^{k_n} \mathcal{L}(f(\mathcal{X};\theta),\mathcal{Y}) - \mathcal{L}(f(\mathcal{X};\theta_{k_i}),\mathcal{Y}),$$

where θ_{k_i} is the k_i-sparse approximation of θ.

Effectively, this value is just the average difference between the un-pruned network and the pruned networks over various levels of sparsity. The smaller the value of the average compressibility, the more compressible the model is.

To focus on the question of this section, we analyse the correlation between the persistent homology dimension and the compressibility of the network. Our results in Figure 5.8 give no clear indication of of a general relationship between these two variables. For MNIST, we find that our results differ depending on the pruning method used. For magnitude-based pruning, the increase of the persistent homology dimension benefits the model's compressibility. The opposite is said for random pruning where increasing the tail index hurts compression. CIFAR10 is more consistent with its behaviour than MNIST as its relation to the average compressibility remains unchanged at a correlation very close to zero over both pruning techniques.

Very little may be said about the relationship between the persistent homology dimension and the compressibility of models. It appears as if the presence of heavier-tailed noise **does not** have a particular effect on the prunability of the model. This is a result that sharply contrasts with the work of Barsbey et al where the heavy-tails were key in achieving the compression benefits.

We now conclude our experiments by tackling the overall question this dissertation aims to answer.

> Can the improvement in generalisation induced by minimising the persistent homology dimension be attributed to the compressibility of the resultant model?

Our answer to this question is no - at least not in the context of network pruning. Figure 5.9 plots the models that were trained in the previous section in accordance to their generalisation error against the average compressibility. No relationship between generalisation and compressibility may be inferred from the graph as, given the value of one the variables, it is not reliably possible to infer the other. The p-values as given in the table beneath the figure plots confirm this.

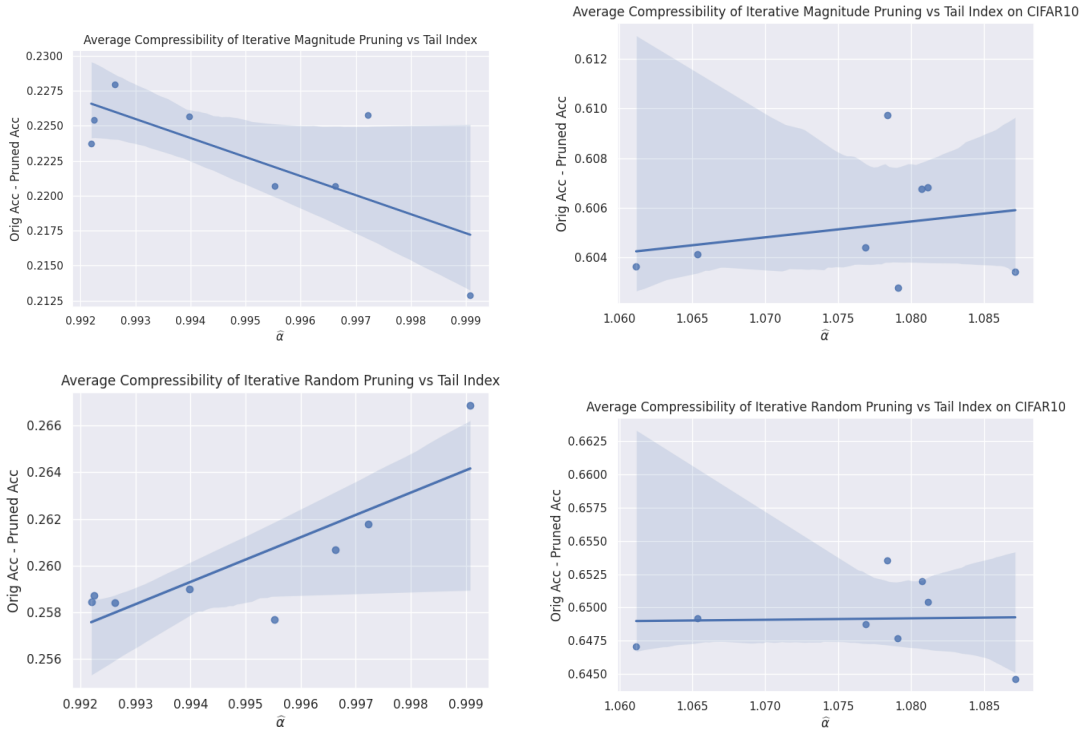
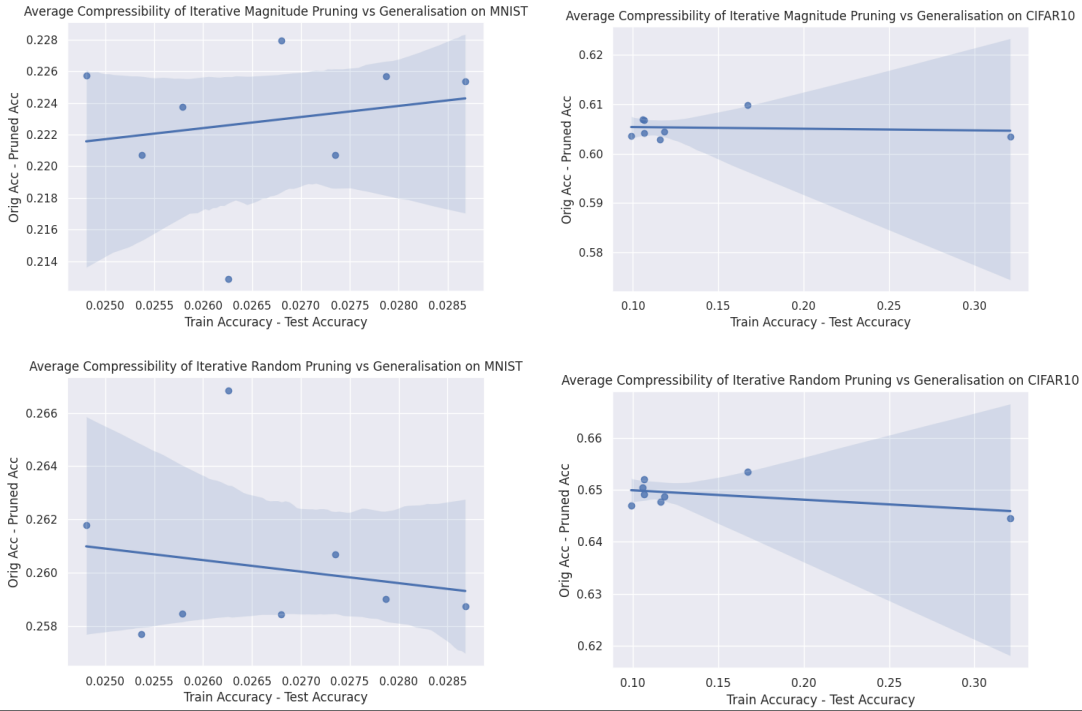


Figure 5.8: Plots of the average compressibility against tail index estimations for both iterative random and iterative magnitude pruning.

Given the results of the previous section, this is not a surprising result. The indicators of compressibility that we explored (sparsity and minima flatness) also had that neither the tail index nor compressibility were correlated.



	MNIST	CIFAR10
IMP	0.1932,0.6467	0.2581, 0.5372
IRP	-0.189, 0.654	0.2581, 0.5371

Figure 5.9: Plots of the average compressibility against generalisation error for both iterative random and iterative magnitude pruning. The table below gives the Pearson correlation coefficient on the left alongside its p-value on the right.

Chapter 6

Conclusion

Persistent homology dimension regularisation is an approach to increasing the 'heavy-tailedness' of stochastic gradient descent and improve upon generalisation performance. This dissertation explored whether this generalisation improvement may be attributed to compression, influenced by a number of works the link generalisation and compression [9, 11, 12]. After careful analysis of a sequence of experiments aimed to analyse indicators of compressibility, we conclude that it is not the case that the generalisation caused by Dim_{PH} regularisation is owed to compressibility.

We have therefore answered the question that we detailed in the introduction and the sub-questions that were formed as a part of the development of the problem statement.

6.1 Challenges

The main challenge experienced over the course of this dissertation is that its topic extends into many varied fields. To be able to formulate the question this dissertation asks incurs a high overhead cost of knowledge within these fields.

6.2 Limitations and Future Work

The limitations of our work surround the experimental setup itself that resulted in such results. For example, we only consider one type of neural network architecture. It is difficult to generalise results from a single model structure to a much wider range therefore the reliability of our experiments would have benefit from more diverse architectures such as convolutional networks. The analysis of more complex models would also for the easier learning of hard tasks, such as achieving reasonable performance on CIFAR10. We are also rather limited in out exploration of compressibility. We implement only two versions of network pruning - iterative random pruning and iterative magnitude pruning. As mentioned in the preliminaries, there is a wealth of literature within network pruning alongside other methods of compression, such as knowledge distillation, that could be considered.

Despite receiving negative results, we were able to uncover a number of unusual properties surrounding the use of Dim_{PH} regularisation:

- Referring back to Section 4.1, we discussed the existence of conflicting results regarding the correlation of the tail index with generalisation. In our experiments, we did not observe the same trend of low tail indices correlating with low generalisation error. In Figure 5.2, models with a lower tail index will often have a worse generalisation error.
- The experiments of Section 5.2 all produced results where a trend in the value of the regularisation constant could be found. No such patterns were found with the tail index. This raises the question of whether the it could be the case that the effect of Dim_{PH} regularisation influences some other property of SGD, besides the tail index of the process.
- The benefits that SGD usually receives from increasing the noise via the ratio of the step size to the batch size (sparser weights, improved compressibility) do not occur when the noise is induced by controlling the persistent homology dimension.

Appendix A

Tail Index Estimation Hyperparameter Tuning

Theorem 2. *Let $\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_K$ for $K = K_1 \cdot K_2$ be an i.i.d sequence of strictly $\alpha-$stable random vectors. Let $Y_i = \sum_{j=1}^{K_1} \mathcal{B}_{j+(i-1)K_1}$ for $i = 1, \cdots, K_2$ be a sequence constructed of sums of batches of random vectors from $\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_K$. Let*

$$\widehat{\frac{1}{\alpha}} = \frac{1}{K_1}\left(\frac{1}{K_2}\sum_{i=1}^{K_2}\log\|Y_i\| - \frac{1}{K}\sum_{i=1}^{K}\log\|\mathcal{B}_i\|\right).$$

converges to $\frac{1}{\alpha}$ almost surely as $K_2 \to \infty$.

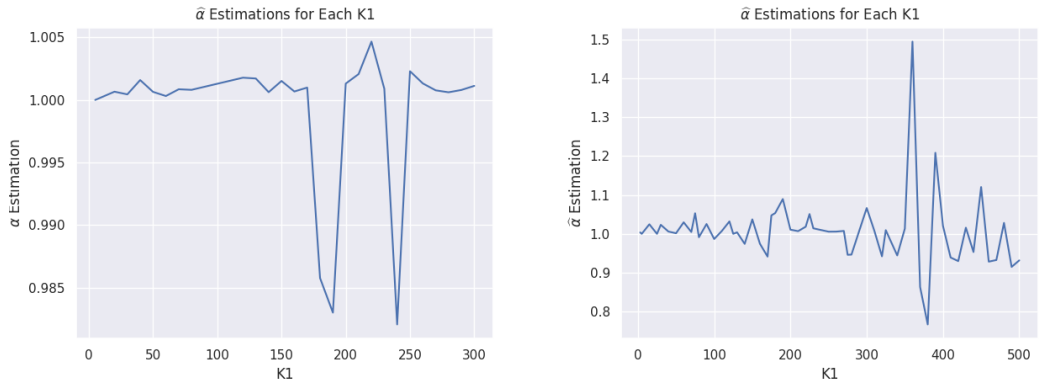


Figure A.1: Plots of the estimation of the tail index for a run of training a model on MNIST (left) and CIFAR10 (right) against the choice of K_1.

To implement the tail index estimation, a choice of K_1 needs to be made. To facilitate this choice, we calculate the estimate for multiple values of $K1$ on a run of unregularised training. For MNIST, the choice of K_1 causes little variance in the tail index estimate but for CIFAR10 the variance is much more significant. Due to the low variance around the point, we use a value of $K_1 = 150$ for both datasets. Pseudocode for the algorithm used to implement Theorem 2 is given in Algorithm 4, adapted from the algorithm used in [8] to be more space efficient.

Algorithm 4 Estimation of Tail Index α

procedure ALPHAESTIMATION($\mathcal{B} = \{\mathcal{B}_i\}_{i=1}^{K_2}, K, K_1, K_2$)
  $X \leftarrow 0, Y \leftarrow []$
  for Batch $\mathcal{B}_i \in \mathcal{B}$ **do**
    $X \leftarrow X + \sum_j \log\left(\|\mathcal{B}_{i,j}\|\right)$
    $Y[i] \leftarrow \|\sum_j \mathcal{B}_{i,j}\|$
  end for
  $\beta = \frac{1}{\log K_1}\left(\frac{1}{K_2}\sum_i \log Y[i] - \frac{X}{K}\right)$
  return $\frac{1}{\beta}$ ▷ $\widehat{\alpha}$ is the reciprocal of β
end procedure

Bibliography

[1] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. *In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning*. arXiv:1412.6614 [cs, stat]. Apr. 2015. URL: http://arxiv.org/abs/1412.6614 (visited on 04/06/2023).

[2] Samuel L. Smith et al. *On the Origin of Implicit Regularization in Stochastic Gradient Descent*. arXiv:2101.12176 [cs, stat]. Jan. 2021. URL: http://arxiv.org/abs/2101.12176 (visited on 03/24/2023).

[3] Pan Zhou et al. *Towards Theoretically Understanding Why SGD Generalizes Better Than ADAM in Deep Learning*. arXiv:2010.05627 [cs, math, stat]. Nov. 2021. DOI: 10.48550/arXiv.2010.05627. URL: http://arxiv.org/abs/2010.05627 (visited on 02/24/2023).

[4] Alnur Ali, Edgar Dobriban, and Ryan J. Tibshirani. *The Implicit Regularization of Stochastic Gradient Flow for Least Squares*. arXiv:2003.07802 [cs, math, stat]. June 2020. URL: http://arxiv.org/abs/2003.07802 (visited on 04/07/2023).

[5] Difan Zou et al. *The Benefits of Implicit Regularization from SGD in Least Squares Problems*. arXiv:2108.04552 [cs, math, stat]. July 2022. DOI: 10.48550/arXiv.2108.04552. URL: http://arxiv.org/abs/2108.04552 (visited on 04/10/2023).

[6] Sepp Hochreiter and Jurgen Schmidhuber. "Flat Minima". In: Mar. 1996. URL: https://www.bioinf.jku.at/publications/older/3304.pdf (visited on 03/09/2023).

[7] Lei Wu, Zhanxing Zhu, and E Weinan. *Towards Understanding Generalization of Deep Learning: Perspective of Loss Landscapes*. 2017.

[8] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. *A Tail-Index Analysis of Stochastic Gradient Noise in Deep Neural Networks*. arXiv:1901.06053 [cs, stat]. Jan. 2019. DOI: 10.48550/arXiv.1901.06053. URL: http://arxiv.org/abs/1901.06053 (visited on 02/24/2023).

[9] Tolga Birdal et al. *Intrinsic Dimension, Persistent Homology and Generalization in Neural Networks*. en. arXiv:2111.13171 [cs, math, stat]. Nov. 2021. URL: http://arxiv.org/abs/2111.13171 (visited on 11/24/2022).

[10] Umut Şimşekli et al. "Hausdorff Dimension, Heavy Tails, and Generalization in Neural Networks". en. In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12 (Dec. 2021). arXiv:2006.09313 [cs, stat], p. 124014. ISSN: 1742-5468. DOI: 10.1088/1742-5468/ac3ae7. URL: http://arxiv.org/abs/2006.09313 (visited on 12/05/2022).

[11] Taiji Suzuki, Hiroshi Abe, and Tomoaki Nishimura. *Compression based bound for non-compressed network: unified generalization error analysis of large compressible deep neural network*. arXiv:1909.11274 [cs, stat]. June 2020. DOI: 10.48550/arXiv.1909.11274. URL: http://arxiv.org/abs/1909.11274 (visited on 02/24/2023).

[12] Daniel Hsu et al. *Generalization bounds via distillation*. arXiv:2104.05641 [cs]. Apr. 2021. DOI: 10.48550/arXiv.2104.05641. URL: http://arxiv.org/abs/2104.05641 (visited on 02/24/2023).

[13] Amir Gholami et al. *A Survey of Quantization Methods for Efficient Neural Network Inference*. arXiv:2103.13630 [cs]. June 2021. URL: http://arxiv.org/abs/2103.13630 (visited on 03/09/2023).

[14] Steven A. Janowsky. "Pruning versus clipping in neural networks". en. In: *Physical Review A* 39.12 (June 1989), pp. 6600–6603. ISSN: 0556-2791. DOI: 10.1103/PhysRevA.39.6600. URL: https://link.aps.org/doi/10.1103/PhysRevA.39.6600 (visited on 03/05/2023).

[15] Davis Blalock et al. *What is the State of Neural Network Pruning?* arXiv:2003.03033 [cs, stat]. Mar. 2020. URL: http://arxiv.org/abs/2003.03033 (visited on 12/23/2022).

[16] Hugo Tessier. *Neural Network Pruning 101*. en. Sept. 2021. URL: https://towardsdatascience.com/neural-network-pruning-101-af816aaea61 (visited on 12/23/2022).

[17] K. Erciyes. "Parallel and Sparse Matrix Computations". en. In: *Algebraic Graph Algorithms: A Practical Guide Using Python*. Ed. by K. Erciyes. Undergraduate Topics in Computer Science. Cham: Springer International Publishing, 2021, pp. 69–86. ISBN: 978-3-030-87886-3. DOI: 10.1007/978-3-030-87886-3_5. URL: https://doi.org/10.1007/978-3-030-87886-3_5 (visited on 04/11/2023).

[18] Jonathan Frankle and Michael Carbin. *The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks*. arXiv:1803.03635 [cs]. Mar. 2019. URL: http://arxiv.org/abs/1803.03635 (visited on 12/23/2022).

[19] Katharine Turner, Sayan Mukherjee, and Doug M. Boyer. *Persistent Homology Transform for Modeling Shapes and Surfaces*. arXiv:1310.1030 [math, stat]. July 2014. DOI: 10.48550/arXiv.1310.1030. URL: http://arxiv.org/abs/1310.1030 (visited on 01/06/2023).

[20] Talha Qaiser et al. *Fast and Accurate Tumor Segmentation of Histology Images using Persistent Homology and Deep Convolutional Features*. arXiv:1805.03699 [cs]. May 2018. DOI: 10.48550/arXiv.1805.03699. URL: http://arxiv.org/abs/1805.03699 (visited on 01/06/2023).

[21] Yuan Yao et al. "Topological Methods for Exploring Low-density States in Biomolecular Folding Pathways". In: *The Journal of Chemical Physics* 130.14 (Apr. 2009). arXiv:0812.3426 [q-bio], p. 144115. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.3103496. URL: http://arxiv.org/abs/0812.3426 (visited on 01/06/2023).

[22] Vin de Silva and Robert Ghrist. "HOMOLOGICAL SENSOR NETWORKS". en. In: ().

[23] Bastian Rieck et al. “Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology”. In: (Feb. 2019). arXiv:1812.09764 [cs, math, stat], 25 p. DOI: 10.3929/ethz-b-000327207. URL: http://arxiv.org/abs/1812.09764 (visited on 03/09/2023).

[24] Felix Hensel, Michael Moor, and Bastian Rieck. “A Survey of Topological Machine Learning Methods”. In: *Frontiers in Artificial Intelligence* 4 (2021). ISSN: 2624-8212. URL: https://www.frontiersin.org/articles/10.3389/frai.2021.681108 (visited on 01/06/2023).

[25] Tamal Krishna Dey and Yusu Wang. *Computational Topology for Data Analysis*. en. 1st ed. Cambridge University Press, Feb. 2022. ISBN: 978-1-00-909995-0 978-1-00-909816-8. DOI: 10.1017/9781009099950. URL: https://www.cambridge.org/core/product/identifier/9781009099950/type/book (visited on 01/06/2023).

[26] David Hilbert. “Ueber die stetige Abbildung einer Line auf ein Flächenstück”. de. In: *Mathematische Annalen* 38.3 (Sept. 1891), pp. 459–460. ISSN: 1432-1807. DOI: 10.1007/BF01199431. URL: https://doi.org/10.1007/BF01199431 (visited on 02/12/2023).

[27] Kenneth Falconer. *Fractal Geometry*. English. URL: http://archive.org/details/FractalGeometry (visited on 02/12/2023).

[28] Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. *On the Validity of Modeling SGD with Stochastic Differential Equations (SDEs)*. arXiv:2102.12470 [cs, stat]. June 2021. DOI: 10.48550/arXiv.2102.12470. URL: http://arxiv.org/abs/2102.12470 (visited on 03/30/2023).

[29] Stanisław Jastrzebski et al. *Three Factors Influencing Minima in SGD*. arXiv:1711.04623 [cs, stat]. Sept. 2018. URL: http://arxiv.org/abs/1711.04623 (visited on 03/03/2023).

[30] Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. en. 1st ed. Cambridge University Press, Apr. 2019. ISBN: 978-1-108-18673-5 978-1-316-51008-7 978-1-316-64946-6. DOI: 10.1017/9781108186735. URL: https://www.cambridge.org/core/product/identifier/9781108186735/type/book (visited on 03/27/2023).

[31] Yimin Xiao. “Random fractals and Markov processes”. en. In: *Proceedings of Symposia in Pure Mathematics*. Ed. by Michel Lapidus and Machiel van Frankenhuijsen. Vol. 72.2. Providence, Rhode Island: American Mathematical Society, 2004, pp. 261–338. ISBN: 978-0-8218-3638-5 978-0-8218-9378-4. DOI: 10.1090/pspum/072.2/2112126. URL: http://www.ams.org/pspum/072.2 (visited on 04/13/2023).

[32] R M Blumenthalo and R K Getoor. “SOME THEOREMS ON STABLE PROCESSES”. en. In: ().

[33] Benjamin Schweinhart. *Fractal Dimension and the Persistent Homology of Random Geometric Complexes*. en. Aug. 2018. URL: https://arxiv.org/abs/1808.02196v6 (visited on 04/12/2023).

[34] Jonathan Jaquette and Benjamin Schweinhart. *Fractal Dimension Estimation with Persistent Homology: A Comparative Study*. en. July 2019. DOI: 10.1016/j.cnsns.2019.105163. URL: https://arxiv.org/abs/1907.11182v2 (visited on 04/12/2023).

[35] Umut Şimşekli et al. *On the Heavy-Tailed Theory of Stochastic Gradient Descent for Deep Neural Networks*. arXiv:1912.00018 [cs, math, stat] version: 1. Nov. 2019. URL: http://arxiv.org/abs/1912.00018 (visited on 03/09/2023).

[36] Melih Barsbey et al. *Heavy Tails in SGD and Compressibility of Overparametrized Neural Networks*. arXiv:2106.03795 [cs, stat]. June 2021. URL: http://arxiv.org/abs/2106.03795 (visited on 02/11/2023).

[37] Mert Gurbuzbalaban, Umut Şimşekli, and Lingjiong Zhu. *The Heavy-Tail Phenomenon in SGD*. arXiv:2006.04740 [cs, math, stat]. June 2021. URL: http://arxiv.org/abs/2006.04740 (visited on 03/09/2023).

[38] Gerard Ben Arous and Alice Guionnet. *The spectrum of heavy-tailed random matrices*. arXiv:0707.2159 [math-ph]. July 2007. DOI: 10.48550/arXiv.0707.2159. URL: http://arxiv.org/abs/0707.2159 (visited on 04/12/2023).

[39] Stefano Favaro, Sandra Fortini, and Stefano Peluchetti. *Stable behaviour of infinitely wide deep neural networks*. arXiv:2003.00394 [cs, stat]. Feb. 2020. DOI: 10.48550/arXiv.2003.00394. URL: http://arxiv.org/abs/2003.00394 (visited on 02/25/2023).

[40] Yann LeCun, Corinna Cortes, and Chris Burges. *MNIST handwritten digit database*. June 2010. URL: http://yann.lecun.com/exdb/mnist/ (visited on 03/03/2023).

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html (visited on 03/09/2023).

[42] Mohammad Mohammadi, Adel Mohammadpour, and Hiroaki Ogata. "On estimating the tail index and the spectral measure of multivariate $$\alpha $$-stable distributions". en. In: *Metrika* 78.5 (July 2015), pp. 549–561. ISSN: 1435-926X. DOI: 10.1007/s00184-014-0515-7. URL: https://doi.org/10.1007/s00184-014-0515-7 (visited on 02/24/2023).

[43] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. "Measuring and testing dependence by correlation of distances". In: *The Annals of Statistics* 35.6 (Dec. 2007). Publisher: Institute of Mathematical Statistics, pp. 2769–2794. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/009053607000000505. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-35/issue-6/Measuring-and-testing-dependence-by-correlation-of-distances/10.1214/009053607000000505.full (visited on 04/12/2023).

[44] *The Gini coefficient - Office for National Statistics*. URL: https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/families/methodologies/theginicoefficient (visited on 04/12/2023).

[45] Niall P. Hurley and Scott T. Rickard. *Comparing Measures of Sparsity*. arXiv:0811.4706 [cs, math]. Apr. 2009. URL: http://arxiv.org/abs/0811.4706 (visited on 12/21/2022).

[46] Yucong Liu, Shixing Yu, and Tong Lin. *Regularizing Deep Neural Networks with Stochastic Estimators of Hessian Trace*. arXiv:2208.05924 [cs]. Feb. 2023. URL: http://arxiv.org/abs/2208.05924 (visited on 04/12/2023).