

TOPOLOGICAL PRUNER
A NEURAL NETWORK PRUNER USING TOPOLOGICAL
DATA ANALYSIS

W.M.M.J.U. Perera

199481D

Dissertation submitted in partial fulfillment of the requirements for the degree Master
of Science

Department of Computational Mathematics

University of Moratuwa,

Sri Lanka

March 2022

Declaration

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

W. M. M. J. U. Perera

Signature of Student

Date:03/03/2022

Supervised by

Dr. Subha Fernando

Dr. Ashwini Amarasinghe

Signature of Supervisor(s)

Date:27/06/2021

Dedication

I dedicate my dissertation to my family and many friends.

A special feeling of gratitude to my loving parents, Mr. Susantha Perera and Mrs. Manel Irangani
whose words of encouragement and push for tenacity ring in my ears
and also to my wife Sherine Weerasinghe who was the best source of inspiration.

Acknowledgement

First and foremost, I am grateful to my supervisors, Dr. Subha Fernando and Dr. Ashwini Amarasinghe for their valuable advices, continuous support, and patience during my research. Their immense knowledge and ample experience encouraged me all the time during my academic research and daily life. I would also like to extend my thankfulness to Prof. Asoka Karunananda for his guidance to prepare thesis materials and directing in the correct path. My sincere gratitude goes to all other lecturers and non-academic staff members who helped me to make this project a success.

I would also like to thank all my teachers from the kindergarten to the Master's level. Specially, Prof. U.N.B. Dissanayeke and Dr. Shelton Perera for strengthening my foundation in Mathematics. I would not be able to complete this research without the passion and curiosity in Mathematics, which they planted within myself.

Finally, I would like to express my gratitude to my parents, my wife and my friends. Without their tremendous understanding and encouragement in the past few years, it would not be impossible for me to complete my study programme.

Abstract

Architectural damage due to neural network pruning has been a research problem. To recover the accuracy loss, after pruning, pruned neural network needed to be trained further for a certain time period to gain the accuracy back. If the damage done by the pruning process is severe, some layers can collapse and at worse, the entire model may become untrainable. Therefore, pruning process needs to be done carefully to prevent any significant damage to the neural network. Although some existing approaches have been used to overcome this issue by identifying the salience of a neuron with respect to the overall architecture, it is not computationally efficient. Further, the exiting solutions do not count the topological meaning of the neural network architecture during the pruning process. We believe that identifying the salience of neuron with respect to the layer is sufficient to avoid severe damages to the overall architecture.

Topology, the champion of mathematical shapes, has been introduced to solve the aforesaid problem. We introduce ‘Topological Pruner’, a novel pruner that uses a genetic algorithm powered by a topological fitness function to identify removable neurons of each layer of a pre trained neural network. After pruning is done, the model is retrained so that the parameters of the remaining neuron can be readjusted to recover the model. As per to our knowledge this is the first ever attempt to use persistence homology, a topological tool for pruning.

Number of parameters, FLOPs and recovery time of the new pruner is evaluated on CIFAR10 dataset on VGG-16 architecture against L1Filter Pruner, L2Filter Pruner and FPGM Pruner. Evaluation results show that the new pruner competes well with the existing pruners. We conclude that, topological data analysis can be used to explain the recoverability and mitigate damage cause by neural network pruning.

Table of Content

Declaration	ii
Dedication	iii
Acknowledgement	iv
Abstract	v
Table of Content	vi
List of Figures	x
List of Tables	xi
List of Abbreviations	xii
1. Introduction	1
1.1 Prolegomena	1
1.2 Aims and Objectives	1
1.3 Background and Motivation	1
1.4 Problem Definition	2
1.5 Proposed Solution	2
1.6 Resource requirements	3
1.7 Structure of the Thesis	4
1.8 Summary	4
2. Background Theory – Model Compression	5
2.1 Introduction	5
2.2 Neural Architecture Search	5
2.3 Model Compression Techniques	5
2.4 Neural Network Pruning	6
2.5 Summary	6
3. Literature Review – Neural Network Pruning	7
3.1 Introduction	7
3.2 Gestation in neural network pruning	7
3.3 Breakthrough in Neural network pruning	8
3.4 Challenges in neural network pruning	8
3.5 Problem Definition	10
3.6 Summary	10

4.	Technologies Adopted – Topological Data Analysis and Genetic Algorithms	11
4.1	Introduction	11
4.2	Topology as a Branch of Mathematics	11
4.2.1	Definition of a Topological Space	13
4.2.2	The World through the eyes of a Topologist	13
4.3	Algebraic Topology	16
4.3.1	Simplexes, polyhedrons and Simplicial Complexes	16
4.3.2	Persistence Homology, Bar codes and Persistence Diagrams	19
4.3.3	Bottleneck Distance	21
4.4	Topological Data Analysis	22
4.5	Genetic Algorithms	25
4.5.1	Encoding schemes	27
4.5.2	Selection techniques	28
4.5.3	Offspring Generation	29
4.6	Summary	30
5.	Approach – Neural Network Pruning using Topological Data Analysis	31
5.1	Introduction	31
5.2	Hypothesis	31
5.3	Input	31
5.3.1	Preprocessing	32
5.4	Output	32
5.5	Process	32
5.6	Users	32
5.7	Features	33
5.8	Summary	33
6.	Design – A topology based GA powered pruner	34
6.1	Introduction	34
6.2	Compressor Module	34
6.2.1	Configuration List	35
6.2.2	Pre-trained neural network	35
6.2.3	Mask	35
6.3	Data Collector Module	36
6.3.1	Weight Matrix	36

6.3.2	Point Cloud	36
6.4	Metric Calculator Module	37
6.5	Evolutionary Module	37
6.5.1	Neuron Mask	38
6.6	Sparsity Allocator Module	38
6.6.1	Connector Mask	38
6.7	Speedup Module	39
6.7.1	Pruned Neural Network	39
6.8	Summary	39
7.	Implementation – Topological Pruner	40
7.1	Introduction	40
7.2	Frameworks used	40
7.2.1	Neural Network Intelligence	40
7.2.2	GUDHI	41
7.2.3	Pyeasyga	41
7.3	Special hardware, software and infrastructure used	41
7.4	Data collector implementation	42
7.5	Metric Calculator implementation	43
7.6	Evolutionary Algorithm Implementation	44
7.7	Sparsity Allocator Implementation	45
7.8	System overview	46
7.9	Summary	47
8.	Evaluation – A Quantitative Analysis	48
8.1	Introduction	48
8.2	Evaluation Strategy	48
8.2.1	FLOPS	48
8.2.2	Number of parameters left	49
8.2.3	Recoverability	49
8.2.4	Convergence of the network	49
8.3	Experimental Setup	49
8.4	Standard Tests	51
8.5	Recoverability Test	52
8.6	Convergence Test	53

8.7	Summary.....	54
9.	Conclusion	55
9.1	Introduction.....	55
9.2	Achievements.....	55
9.3	Limitations.....	56
9.4	Future work.....	56
9.5	Summary.....	57
	References.....	58
	Appendix I: Topological Genetic Algorithm	61
	Appendix II: Topological Data Analysis	62
	Appendix III: Topological Pruner.....	63
	Appendix IV: VGG Model Implementation.....	64

List of Figures

Figure 1.1: Top Level Architecture	2
Figure 1.2: Topological Pruner Abstract Model	3
Figure 2.1: Model Compression Taxonomy	6
Figure 4.1: Branches of Mathematics (Algebraic topology is in the shaded area)	11
Figure 4.2: Topological Similarities	12
Figure 4.3: Neighborhood of x	14
Figure 4.4: A Separated Space	14
Figure 4.5: Continuous Deformations of a Topological Space	15
Figure 4.6: Equivalence in Topology	15
Figure 4.7: Polygon with Triangles	16
Figure 4.8: First four simplexes	18
Figure 4.9: Examples for Polyhedrons	18
Figure 4.10: Decomposing a Polyhedral	18
Figure 4.11: Bar codes and Persistence Diagrams	19
Figure 4.12: Generating simplicial complexes	20
Figure 4.13: Filtration of the simplicial complexes	20
Figure 4.14: Features vs Noise	21
Figure 4.15: Pairing a point with its Projection	22
Figure 4.16: Three datasets with similar topological properties	22
Figure 4.17: Topological Data Analysis	23
Figure 4.18: Distinguish Zero from Eight	24
Figure 4.19: Mapper representation of a Hand shaped point cloud	24
Figure 4.20: Responsibilities of each layer of VGG-16	25
Figure 4.21: Taxonomy of Metaheuristics	26
Figure 4.22: General Algorithm for GA (Source: [34])	27
Figure 4.23: Operations use in GA (source [34])	28
Figure 6.1: Abstract composer module	34
Figure 6.2: Abstract Data Collector Module	36
Figure 6.3: Abstract Metric Calculator Module	37
Figure 6.4: Abstract Evolutionary Module	37
Figure 6.5: Abstract Sparsity Allocator Module	38
Figure 6.6: Abstract Speedup Module	39
Figure 7.1: Generating a Point Cloud	43
Figure 7.2: Generate persistence diagram	44
Figure 7.3: Fitness Function	44
Figure 7.4: Implementation of Topological Pruner	46
Figure 7.5: Compression pipeline	46
Figure 7.6: Model Speedup process	47
Figure 8.1: Standard Comparison	51
Figure 8.2: Recoverability Comparison	52

List of Tables

Table 3.1: Issues and Challenges in Current Technologies	9
Table 4.1: Faces of Simplexes	17
Table 7.1: Trainable parameters arranged by layers	42
Table 8.1: Models used in Evaluation	50
Table 8.2: FLOPS and Number of Parameters.....	51
Table 8.3: Recoverability Test Results	52
Table 8.4: Convergence Test Results	53

List of Abbreviations

CBIS : Component based software engineering

FLOPS: Floating-point Operations per Second

GA : Genetic Algorithms

NAS : Neural Architecture Search

NNP : Neural Network Pruning