

The title of my thesis

Any short subtitle

Lucas Charpentier



Thesis submitted for the degree of
Master in Computational Science
(Imaging and Biomedical Computing)
60 credits

Departement of Informatics
Departement of Physics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2020

The title of my thesis

Any short subtitle

Lucas Charpentier

© 2020 Lucas Charpentier

The title of my thesis

<http://www.duo.uio.no/>

Printed: Reprosentralen, University of Oslo

The title of my thesis

Lucas Charpentier

6th July 2020

Abstract

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Problem Statement	1
1.3	Thesis Outline	1
2	Planning the project	3
2.1	Machine Learning	4
2.1.1	Supervised Learning	4
2.1.2	Unsupervised Learning	4
2.2	Artificial Neural Networks	4
2.2.1	Perceptron	4
2.2.2	Multilayer Perceptron	4
2.2.3	Training a Neural Network	4
2.3	Convolutional Neural Network	4
2.3.1	Convolutional Layers	4
2.3.2	Pooling Layers	4
2.4	Neural Network Training Optimization	4
2.4.1	Weight Initialization	4
2.4.2	Training Batch Size	4
2.4.3	Dropout	4
2.5	Network Pruning	4
2.6	Datasets	4
2.6.1	MNIST	4
2.6.2	Fashion MNIST	4
2.6.3	CIFAR-10	4
2.7	Architectures	4
2.7.1	VGG-16	4
3	Single Layer ANN	5
3.1	Pruning Nodes at Random	5
3.1.1	MNIST	5
3.1.2	Fashion MNIST	7
3.2	Estimating Node Importance based on Loss and Accuracy	9
3.3	Pruning Nodes based on the Loss and Accuracy	9
3.4	Effects of Changing Training Batch Size on Node Importance	9
3.5	Effects of Using Dropout	9

4	Multi-Layer Perceptron	11
4.1	Pruning network with pre-calculated importance	11
4.2	Greedy approach to pruning instead of Exhaustive approach	11
4.3	Iterative weight initialization using Node importance	11
5	Convolutional Neural Network	13
5.1	Looking at effects of per class accuracy after pruning	13
5.2	Pruning based on class accuracy	13
6	Case study: Reducing a VGG-16 model trained on X dataset	15
7	Conclusion	17
7.1	Summary	17
7.2	Future Works	17

List of Figures

3.1	Short title	6
3.2	Short title	6
3.3	Short title	7
3.4	Short title	7
3.5	Short title	8
3.6	Short title	8
3.7	Short title	9
3.8	Short title	10
3.9	Short title	10
3.10	Short title	10

List of Tables

3.1	Short	5
3.2	Short	5
3.3	Short	6
3.4	Short	8
3.5	Short	9
3.6	Short	9

Preface

Chapter 1

Introduction

1.1 Background and Motivation

1.2 Problem Statement

1.3 Thesis Outline

Chapter 2

Planning the project

2.1 Machine Learning

2.1.1 Supervised Learning

2.1.2 Unsupervised Learning

2.2 Artificial Neural Networks

2.2.1 Perceptron

2.2.2 Multilayer Perceptron

2.2.3 Training a Neural Network

2.3 Convolutional Neural Network

2.3.1 Convolutional Layers

2.3.2 Pooling Layers

2.4 Neural Network Training Optimization

2.4.1 Weight Initialization

2.4.2 Training Batch Size

2.4.3 Dropout

2.5 Network Pruning

2.6 Datasets

2.6.1 MNIST

2.6.2 Fashion MNIST

2.6.3 CIFAR-10

2.7 Architectures

2.7.1 VGG-16

Chapter 3

Single Layer ANN

3.1 Pruning Nodes at Random

3.1.1 MNIST

Artificial Neural Network with single hidden layer of 128 nodes, using ADAM as optimizer with a learning rate of 0.001 trained on 5 epochs, batch size of 32. Final Accuracy and Loss on test set are: Loss: 0.0734 Accuracy: 0.9770

	Number of Nodes						
	1	2	4	8	16	32	64
Mean	0.9765	0.9760	0.9750	0.9724	0.9658	0.9464	0.8555
σ	0.0007	0.0010	0.0015	0.0027	0.0053	0.0122	0.0394
min	0.9740	0.9660	0.9635	0.9494	0.9337	0.8688	0.6598
25%	0.9762	0.9756	0.9742	0.9710	0.9632	0.9410	0.8331
50%	0.9767	0.9762	0.9753	0.9728	0.9669	0.9489	0.8616
75%	0.9770	0.9767	0.9760	0.9742	0.9695	0.9543	0.8837
max	0.9778	0.9780	0.9779	0.9769	0.9769	0.9685	0.9334

Table 3.1: Long

Trial text

Trial text

	Number of Nodes						
	1	2	4	8	16	32	64
Mean	0.0751	0.0767	0.0801	0.0885	0.1090	0.1680	0.4298
σ	0.0023	0.0032	0.0047	0.0083	0.0157	0.0338	0.0989
min	0.0720	0.0714	0.0710	0.0735	0.0791	0.1006	0.2311
25%	0.07	0.	0.9742	0.9710	0.9632	0.9410	0.8331
50%	0.9767	0.9762	0.9753	0.9728	0.9669	0.9489	0.8616
75%	0.9770	0.9767	0.9760	0.9742	0.9695	0.9543	0.8837
max	0.9778	0.9780	0.9779	0.9769	0.9769	0.9685	0.9334

Table 3.2: Long

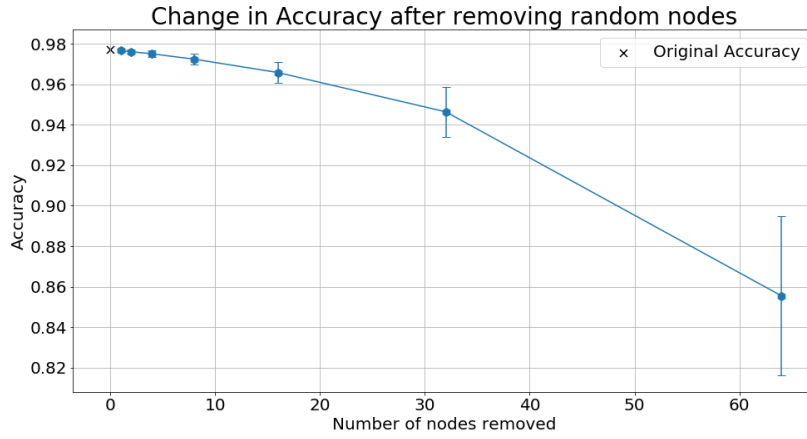


Figure 3.1: Testing

Trial text

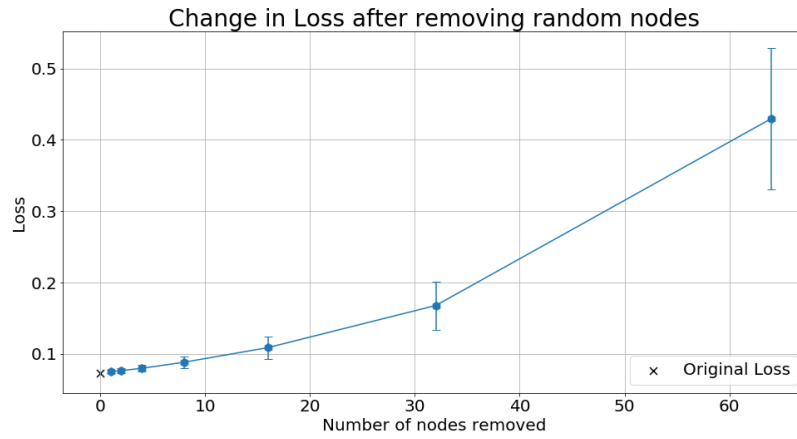


Figure 3.2: Testing

Trial text

	Number of Nodes						
	1	2	4	8	16	32	64
Mean	0.0751	0.0767	0.0801	0.0885	0.1090	0.1680	0.4298
σ	0.0023	0.0032	0.0047	0.0083	0.0157	0.0338	0.0989
min	0.0720	0.0714	0.0710	0.0735	0.0791	0.1006	0.2311
25%	0.	0.9756	0.9742	0.9710	0.9632	0.9410	0.8331
50%	0.9767	0.9762	0.9753	0.9728	0.9669	0.9489	0.8616
75%	0.9770	0.9767	0.9760	0.9742	0.9695	0.9543	0.8837
max	0.9778	0.9780	0.9779	0.9769	0.9769	0.9685	0.9334

Table 3.3: Long

Trial text

Number of nodes removed that improve model depending on the number removed at a time

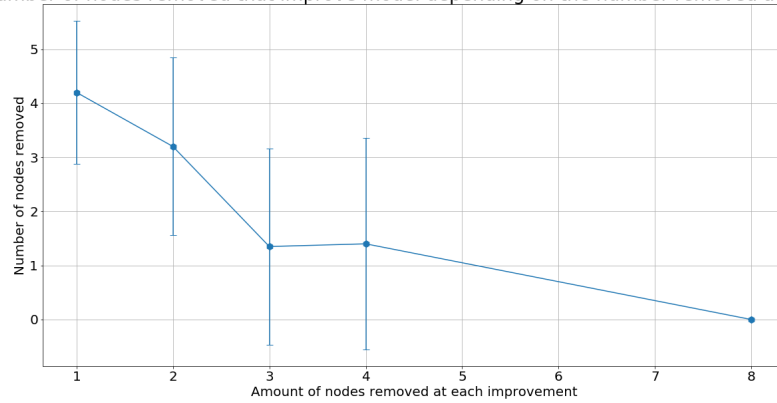


Figure 3.3: Testing

Trial text

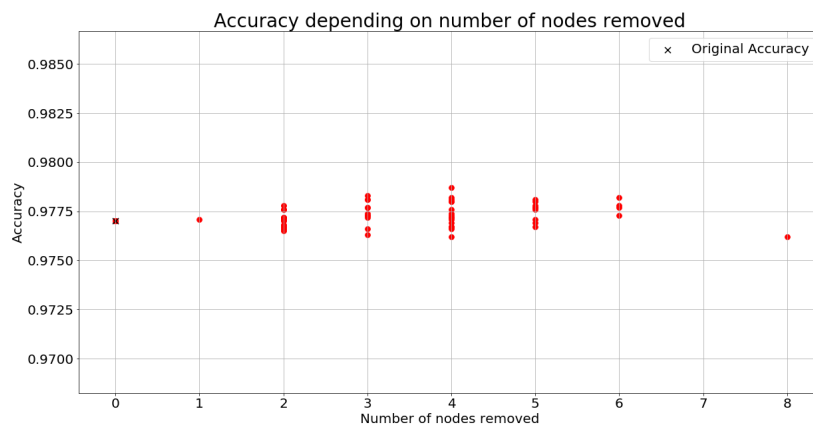


Figure 3.4: Testing

Trial text

Trial text

3.1.2 Fashion MNIST

Trial text

Trial text

Trial text

Trial text

There is trial text here

Trial text

Trial text

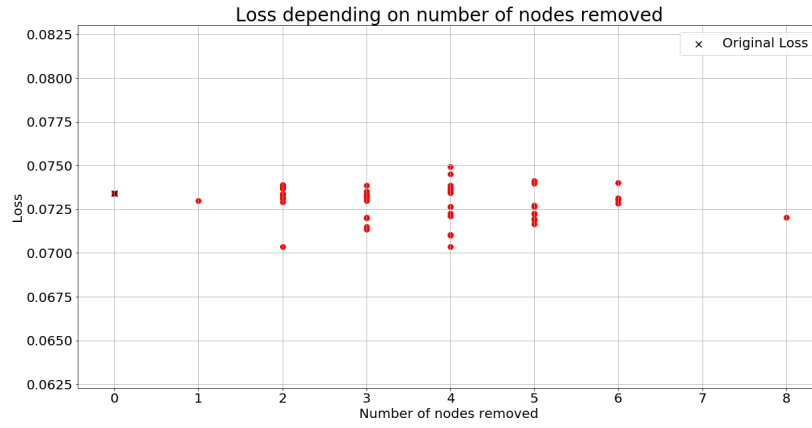


Figure 3.5: Testing

	Number of Nodes						
	1	2	4	8	16	32	64
Mean	0.9765	0.9760	0.9750	0.9724	0.9658	0.9464	0.8555
σ	0.0007	0.0010	0.0015	0.0027	0.0053	0.0122	0.0394
min	0.9740	0.9660	0.9635	0.9494	0.9337	0.8688	0.6598
25%	0.9762	0.9756	0.9742	0.9710	0.9632	0.9410	0.8331
50%	0.9767	0.9762	0.9753	0.9728	0.9669	0.9489	0.8616
75%	0.9770	0.9767	0.9760	0.9742	0.9695	0.9543	0.8837
max	0.9778	0.9780	0.9779	0.9769	0.9769	0.9685	0.9334

Table 3.4: Long

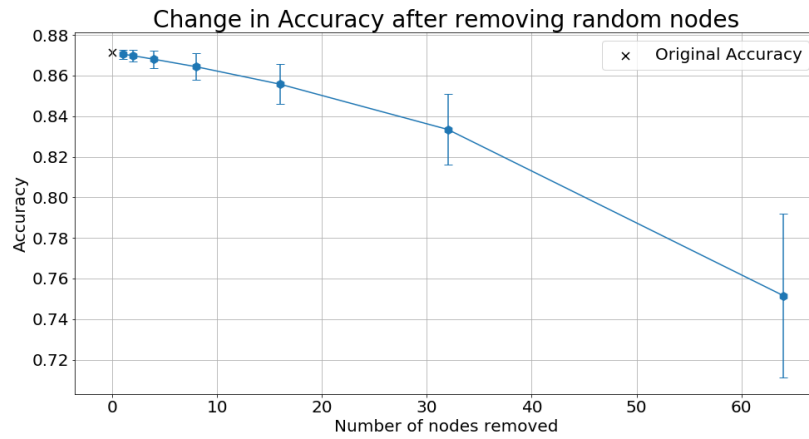


Figure 3.6: Testing

Trial text

Trial text

	Number of Nodes						
	1	2	4	8	16	32	64
Mean	0.9765	0.9760	0.9750	0.9724	0.9658	0.9464	0.8555
σ	0.0007	0.0010	0.0015	0.0027	0.0053	0.0122	0.0394
min	0.9740	0.9660	0.9635	0.9494	0.9337	0.8688	0.6598
25%	0.9762	0.9756	0.9742	0.9710	0.9632	0.9410	0.8331
50%	0.9767	0.9762	0.9753	0.9728	0.9669	0.9489	0.8616
75%	0.9770	0.9767	0.9760	0.9742	0.9695	0.9543	0.8837
max	0.9778	0.9780	0.9779	0.9769	0.9769	0.9685	0.9334

Table 3.5: Long

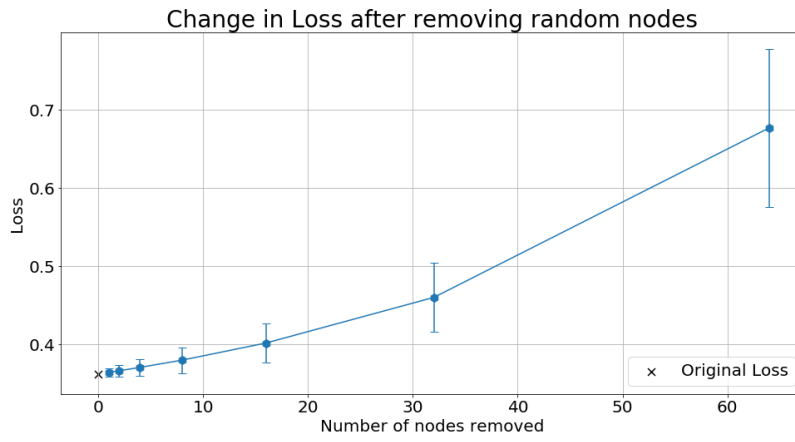


Figure 3.7: Testing

	Number of Nodes						
	1	2	4	8	16	32	64
Mean	0.0751	0.0767	0.0801	0.0885	0.1090	0.1680	0.4298
σ	0.0023	0.0032	0.0047	0.0083	0.0157	0.0338	0.0989
min	0.0720	0.0714	0.0710	0.0735	0.0791	0.1006	0.2311
25%	0.	0.9756	0.9742	0.9710	0.9632	0.9410	0.8331
50%	0.9767	0.9762	0.9753	0.9728	0.9669	0.9489	0.8616
75%	0.9770	0.9767	0.9760	0.9742	0.9695	0.9543	0.8837
max	0.9778	0.9780	0.9779	0.9769	0.9769	0.9685	0.9334

Table 3.6: Long

3.2 Estimating Node Importance based on Loss and Accuracy

3.3 Pruning Nodes based on the Loss and Accuracy

3.4 Effects of Changing Training Batch Size on Node Importance

3.5 Effects of Using Dropout

Number of nodes removed that improve model depending on the number removed at a time

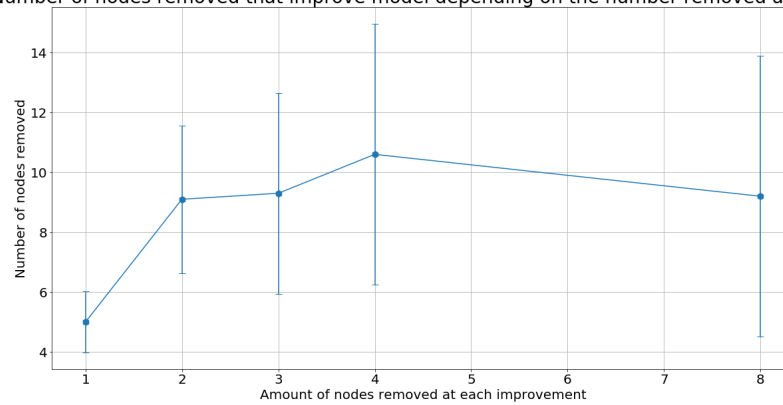


Figure 3.8: Testing

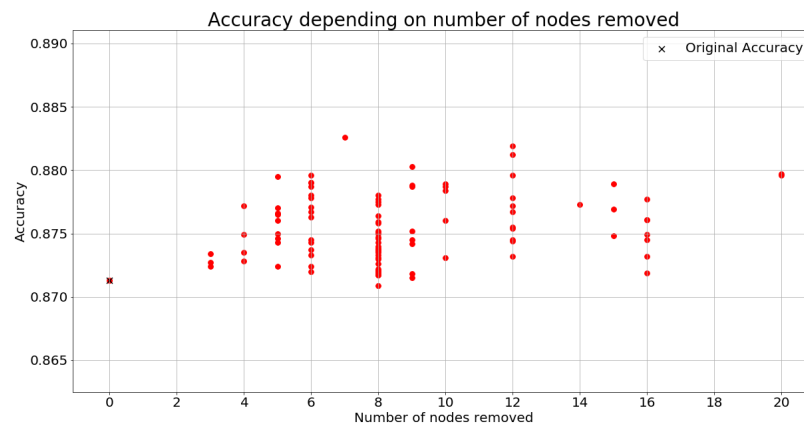


Figure 3.9: Testing

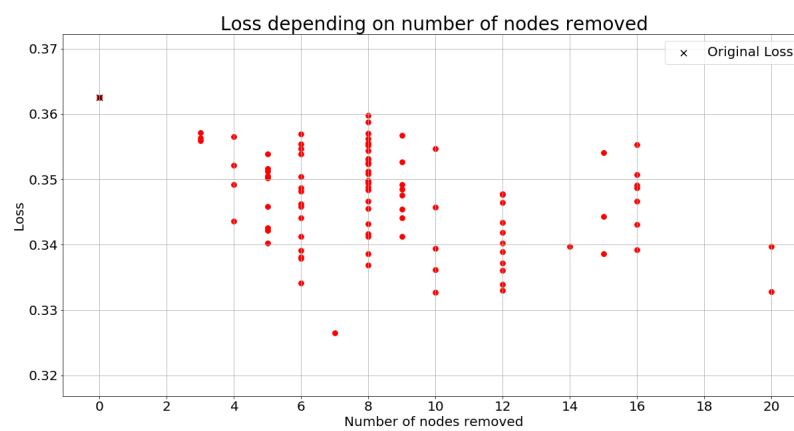


Figure 3.10: Testing

Chapter 4

Multi-Layer Perceptron

- 4.1 Pruning network with pre-calculated importance**
- 4.2 Greedy approach to pruning instead of Exhaustive approach**
- 4.3 Iterative weight initialization using Node importance**

Chapter 5

Convolutional Neural Network

5.1 Looking at effects of per class accuracy after pruning

5.2 Pruning based on class accuracy

Chapter 6

Case study: Reducing a VGG-16 model trained on X dataset

Chapter 7

Conclusion

7.1 Summary

7.2 Future Works

