

MSC ARTIFICIAL INTELLIGENCE
MASTER THESIS

Your Title Here

by

YOUR NAME

student number

December 1, 2022

Number of Credits

Period in which the research was carried out

Supervisor:

Dr A PERSON

Examiner:

Dr A PERSON

Second reader:

Dr A PERSON



UNIVERSITEIT VAN AMSTERDAM

Contents

1	Introduction	1
2	Related work	2
2.1	Pass-through trick	2
2.2	Self-supervision	2
3	Method	4
4	Experiments	5
5	Conclusions	6
A	Optional appendix	7

Abstract

Provide a short overview and description of your thesis here.

Chapter 1

Introduction

This document provides an empty template for your thesis. Please keep the following things in mind when writing your thesis:

- The chapter organization of this template is a generic suggestion, please customize to your needs and vision.
- The guideline is to have a 40 page upper limit for the thesis, with additional optional pages for appendices if needed, for example to provide additional experimental results or mathematical details. Most theses do not end up with an appendix.
- For references, you are free to choose your own style. A standard style is included in this template, for example [**lecun2015deep**].
- The official website of the thesis AI (<https://student.uva.nl/ai/content/az/master-thesis-ai/master-thesis-ai-2020.html>) contains all information regarding the procedures and details of the thesis process.

Chapter 2

Related work

2.1 Pass-through trick

BinnaryConnect: Use the passthrough trick to learn binary (-1 or 1) neural networks [2, 4]

Piggyback: Earliest paper I found (2018) which does the submasking method with the ‘pass-through’ trick. They actually use it for finetuning models like us. [7]

NASpaper: NAS paper which, independently from the Piggyback paper, used similar tricks for performing neural architecture search [12]

Ramanujan et al: Inspired by the NASPaper, they rediscovered the Piggyback method except applied it to training models from scratch and they also control the sparsity (piggy uses a threshold method). [10]

Movement Pruning: They prune models by applying ramanujan’s submasking and the Piggyback method in tandem with training the weights of the neural network (NLP). They use the term ‘hard’ movement pruning to refer to Ramanujan’s method (top k%) and ‘soft’ pruning to refer to the Piggyback method (prune if score higher than threshold). They additionally enhance the pruning by adding a knowledge distillation loss as well. [11]

Block Movement Pruning: They do movement pruning with also distillation but by grouping ‘blocks’ of weights together into the same masking unit (so all weights in the block share a mask). Seems to work quite well. [6]

Supsup: Very cool use of the submasking method: They make a random network learn thousands of different tasks, each task with it’s own supermask. They base it primarily on Ramanujan et al but also mention Piggyback and argue that their method is superior. [13]

Packnet: Paper doing iterative pruning and retraining to teach a network new tasks [8]

Very recent paper that explores the best ways to fine-tune several models including resnets, vits, and convnexts. Their significant contribution is to freeze the the transformer embeddings in order to get the same performance with SGD as you can get with Adam. They get the highest CLIP-finetuned cifar-10 performance I have seen so far so it is a very good baseline to compare accuracies against [5]

2.2 Self-supervision

Momentum contrast uses an embedding queue in combination with distinct key and query embedders such that the query embedder is trained through backpropagation on the current mini-batch (contrasted against all key embeddings in the queue) and the key embedder is updated by slowly adding the query embedder’s parameters to it. Importantly, batch normalization needs to be disabled or shuffled. [3]

Dino is similar to momentum contrast except rather than using embeddings and a queue it just applies a cross-entropy loss between the teacher (key embedder) and student (query

embedder) logits. It however needs to apply 'centering' and 'sharpening' to avoid collapse. [1]

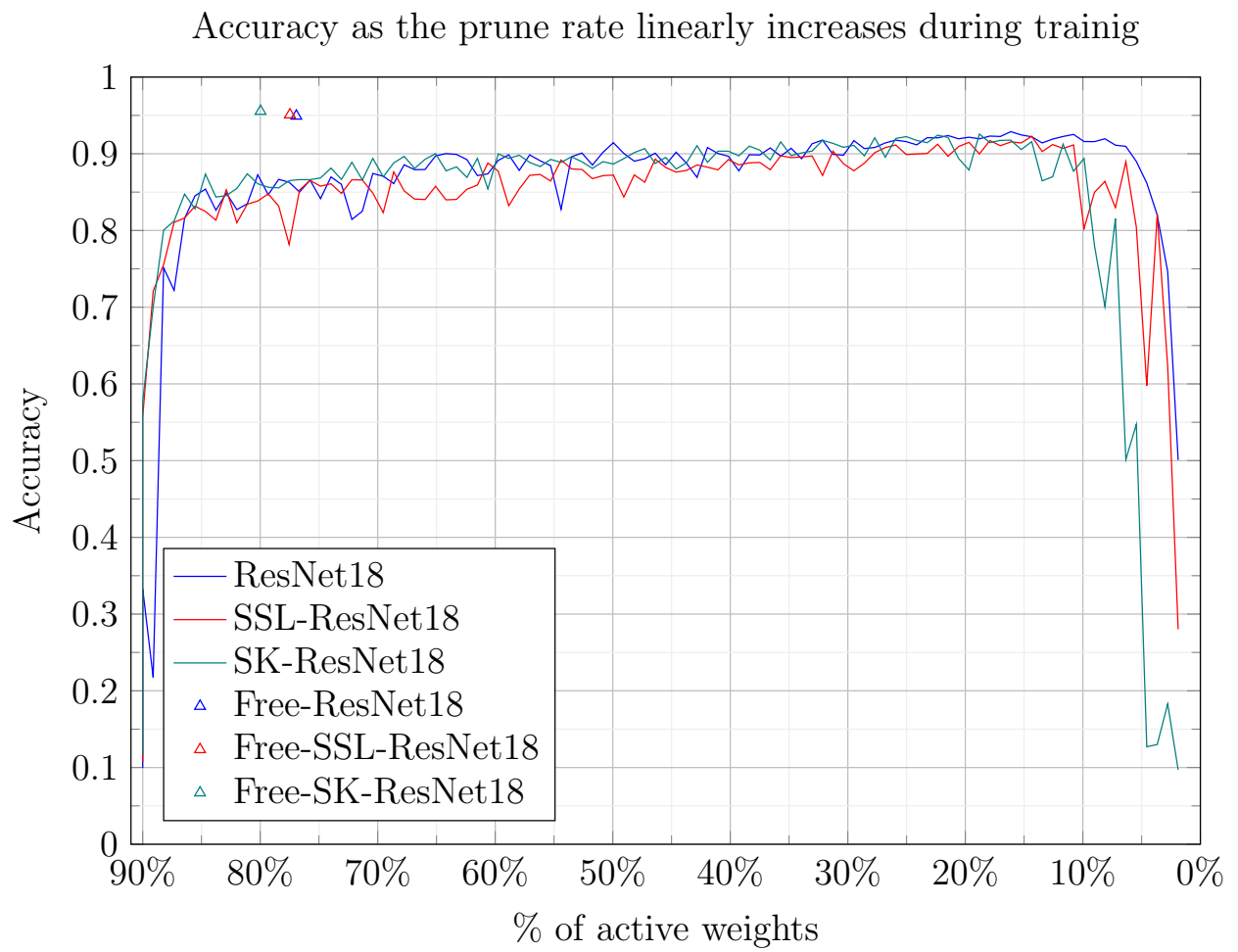
SLIP is an improvement over CLIP which also makes use of self-supervision for pretraining. [9]

Chapter 3

Method

Chapter 4

Experiments



Chapter 5

Conclusions

Appendix A

Optional appendix

Bibliography

- [1] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. May 24, 2021. DOI: 10.48550/arXiv.2104.14294. arXiv: 2104.14294 [cs]. URL: <http://arxiv.org/abs/2104.14294> (visited on 11/22/2022).
- [2] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. *BinaryConnect: Training Deep Neural Networks with Binary Weights during Propagations*. Apr. 18, 2016. DOI: 10.48550/arXiv.1511.00363. arXiv: 1511.00363 [cs]. URL: <http://arxiv.org/abs/1511.00363> (visited on 11/22/2022).
- [3] Kaiming He et al. *Momentum Contrast for Unsupervised Visual Representation Learning*. Mar. 23, 2020. arXiv: 1911.05722 [cs]. URL: <http://arxiv.org/abs/1911.05722> (visited on 11/22/2022).
- [4] Itay Hubara et al. “Binarized Neural Networks”. In: (), p. 9.
- [5] Ananya Kumar et al. *How to Fine-Tune Vision Models with SGD*. Nov. 17, 2022. arXiv: 2211.09359 [cs]. URL: <http://arxiv.org/abs/2211.09359> (visited on 11/27/2022).
- [6] François Lagunas et al. *Block Pruning For Faster Transformers*. Sept. 10, 2021. arXiv: 2109.04838 [cs]. URL: <http://arxiv.org/abs/2109.04838> (visited on 11/22/2022).
- [7] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. *Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights*. Mar. 16, 2018. DOI: 10.48550/arXiv.1801.06519. arXiv: 1801.06519 [cs]. URL: <http://arxiv.org/abs/1801.06519> (visited on 11/22/2022).
- [8] Arun Mallya and Svetlana Lazebnik. “PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT: IEEE, June 2018, pp. 7765–7773. ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00810. URL: <https://ieeexplore.ieee.org/document/8578908/> (visited on 11/22/2022).
- [9] Norman Mu et al. *SLIP: Self-supervision Meets Language-Image Pre-training*. Dec. 23, 2021. arXiv: 2112.12750 [cs]. URL: <http://arxiv.org/abs/2112.12750> (visited on 11/29/2022).
- [10] Vivek Ramanujan et al. “What’s Hidden in a Randomly Weighted Neural Network?” Mar. 30, 2020. arXiv: 1911.13299 [cs]. URL: <http://arxiv.org/abs/1911.13299> (visited on 07/27/2022).
- [11] Victor Sanh, Thomas Wolf, and Alexander M Rush. “Movement Pruning: Adaptive Sparsity by Fine-Tuning”. In: (), p. 12.
- [12] Mitchell Wortsman, Ali Farhadi, and Mohammad Rastegari. *Discovering Neural Wirings*. Nov. 16, 2019. arXiv: 1906.00586 [cs]. URL: <http://arxiv.org/abs/1906.00586> (visited on 11/22/2022).
- [13] Mitchell Wortsman et al. “Supermasks in Superposition”. In: (), p. 12.