

# Faculty of Engineering Sciences

Heidelberg University

Master Thesis  
in Computer Engineering  
submitted by  
Constantin Nicolai  
born in Bretten, Germany  
Day/Month/Year Here



YOUR TITLE HERE

This Master thesis has been carried out by Constantin Nicolai  
at the  
Institute of Computer Engineering  
under the supervision of  
Holger Fröning



## ABSTRACT

Briefly summarize the contents of your work in 150-250 words. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## ZUSAMMENFASSUNG

Deutsche Version. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

*Nice quote here.*  
— Some Author

## ACKNOWLEDGMENTS

Your acknowledgments here if desired





# CONTENTS

1	Introduction and Motivation	1
2	Background	3
2.1	Topic 1	3
2.1.1	Subsection	3
2.1.2	Other Subsection	4
2.2	Topic 2	5
3	State of the Art and Related Works	7
4	First Contribution	9
4.1	Section	9
4.1.1	Subsection 1	9
4.1.2	Subsection 2	9
5	Second Contribution	11
6	Discussion and Outlook	13
A	Appendix	15
	Bibliography	17



# 1

## INTRODUCTION AND MOTIVATION

Introduction to your topic and motivation of your work. Example citation [1] (good book!). Table 1.1 shows an example table and Figure 1.1 an example plot.

Table 1.1: An Example table

Memory [MiB]	Time [s]	Complexity
40	10	$\infty$

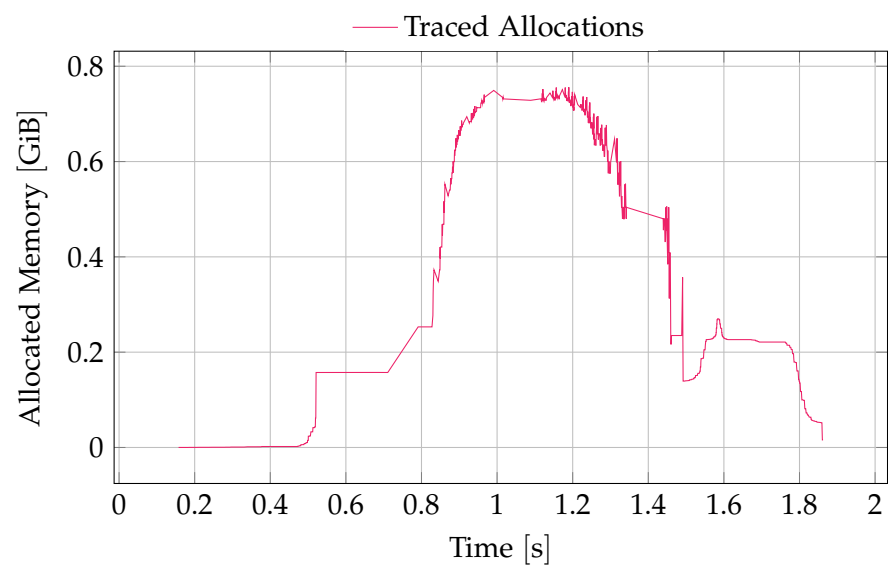


Figure 1.1: An example Figure

# 2 | BACKGROUND

## 2.1 TOPIC 1

The challenge of predicting neural network performance has invited a plurality of approaches. Apart from the methodological approaches they also differ in a number of aspects. While execution time is commonly the metric of choice, only few go further and also study metrics like power consumption and memory footprint. Another important distinction is the workload studied in the work, more specifically, whether both training and inference are studied. For practical reasons it is also relevant which machine learning framework is used and what hardware targets are required and can be predicted for. These many dimensions of possibility result in no work covering all possibilities, but allows for many different approaches which have use cases in a given situation.

### 2.1.1 Subsection

Kaufmann et al. take an approach of performance modeling by means of the computation graph. They are however limited to the Google Tensor Processing Unit in this work.

Justus et al. take an approach exploiting the modular and repetitive nature of DNNs. Given the same operations are repeated over and over in training, often only varying in a few key parameters, these execution time for these base building blocks is measured. This is then done for one batch in the training process and generalized to the whole training process from there. There is however no presentation of the methodology for the execution time measurements.

Qi et al. present PALEO which employs an analytical approach towards predicting the execution for both training and inference of deep neural networks. The analytical approach brings both advantages and disadvantages with it. It does not require a dataset of measured execution times as a training set in the same way many other works do, but on the other hand it also is based on more fixed assumptions about the DNN execution than a more data driven approach.

Wang et al. approach with a mult-layer regression model to predict execution time for training and inference. Their work is however rather limited in terms of hardware targets and different DNNs studied.

Cai et al. focus their work, NeuralPower, on CNNs running on GPUs.

For each target GPU, they collect a dataset and fit a sparse polynomial regression model to predict power, runtime, and energy consumption. While NeuralPower achieves good results, its usefulness has become limited due to its exclusive focus on CNNs, as other DNN architectures have grown in popularity.

Gianitti et al. also exploit the modular nature of DNNs in their approach. They define a complexity metric for each layer type, optionally including backpropagation terms, allowing them to predict execution times for both training and inference. However, their method faces significant limitations, as the complexity metric is only defined for a specific set of operations, making it incompatible with networks that include layers not covered in the original work. As a result, their approach is essentially limited to classic CNN architectures.

Velasco-Montero et al. also take the familiar per-layer approach. Their predictions are based on linear regression models per type of layer, but again for a specific set of predefined operations. Given their focus on low-cost vision devices these restrictions are reasonable, but limit generalizability.

Sponner et al. take a broad approach in their work. It works in the TVM framework giving it high flexibility in target hardware and studied metrics. It is in fact the only work to include execution time, power consumption and memory allocation. Given the automated data collection used to create the dataset basis for the predictions, there are also few limitations to the networks that can be studied with this. The predictions are based on an extremely randomized tree (ERT) approach with XGBoosting applied. The only major drawback for this work is its limitation to only inference.

### 2.1.2 Other Subsection

With all these very different works no single one was able to cover all possible angles to interest, although Sponner et al. got rather close. But given the current landscape of available publications our work will focus on finding the best GPU for a PyTorch job. In order to achieve that, we will cover execution time, power and energy consumption and will provide inference and training predictions for these metrics. Our approach also employs a different method of automatic dataset collection, which allows for a broad field of study. In order to obtain power readings are collected directly through `nvidia-smi`. While due to the scope of this work this limits us to Nvidia GPUs, the methodology could just as well be applied to any other hardware target which supports reporting power readings.

## 2.2 TOPIC 2

Second topic.





# 3

## STATE OF THE ART AND RELATED WORKS

Talk about related works and state of the art, plus possibly problems with SOTA that you are fixing.



# 4 | FIRST CONTRIBUTION

First contribution here.

## 4.1 SECTION

A section.

### 4.1.1 Subsection 1

Details.

### 4.1.2 Subsection 2

Yet another detail.



# 5 | SECOND CONTRIBUTION

The second contribution goes here.



# 6 | DISCUSSION AND OUTLOOK

Summary and discussion of the results and outlook/future work.





# a | APPENDIX

Appendix here



## BIBLIOGRAPHY

- [1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer, 2006. 738 pp. ISBN: 978-0-387-31073-2.



# ERKLÄRUNG

Ich versichere, dass ich diese Arbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

*Heidelberg, den Day/Month/Year Here*

---

Constantin Nicolai