

---

# Application of the Lottery Ticket Hypothesis in NLP and Early Pruning (Proposal)

---

Anwendung der "Lottery Ticket"-Hypothese in NLP und frühem Pruning (Proposal)

Bachelor-Arbeit

Tim Unverzagt

KOM-type-number ???

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Informatik  
Fachbereich ??? (Zweitmitglied)

Fachgebiet Natural Language  
Processing  
| Gutachter |

---

**Application of the Lottery Ticket Hypothesis in NLP and Early Pruning (Proposal)**  
Anwendung der "Lottery Ticket"-Hypothese in NLP und frühem Pruning (Proposal)

Bachelor-Arbeit  
Studiengang: Computational Engineering  
KOM-type-number ???

Eingereicht von Tim Unverzagt  
Tag der Einreichung: dd. month yyyy

Gutachter: ???  
Betreuerin: Anna Filighera

Technische Universität Darmstadt  
Fachbereich Informatik  
Fachbereich ??? (Zweitmitglied)

Fachgebiet Natural Language Processing (KOM)  
| Gutachter |

---

### **Erklärung zur Abschlussarbeit gemäß § 23 Abs. 7 APB der TU Darmstadt**

---

Hiermit versichere ich, Tim Unverzagt, die vorliegende Bachelor-Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Bachelor-Arbeit stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Darmstadt, den dd. month yyyy

---

Tim Unverzagt



---

## Contents

---

<b>1. Introduction</b>	<b>3</b>
1.1. Motivation . . . . .	3
1.2. Problem Statement and Contribution . . . . .	3
1.3. Outline . . . . .	3
<b>2. Background</b>	<b>5</b>
2.1. Basics of Neural Networks <i>WIP</i> . . . . .	5
2.2. The Lottery Ticket Hypothesis . . . . .	6
2.3. Basics of Natural Language Processing . . . . .	7
2.4. Combining LMs & CNNs . . . . .	7
<b>3. Related Work</b>	<b>9</b>
3.1. Lottery-Ticket-Hypothesis . . . . .	9
3.2. Related Work Dynamic Pruning . . . . .	9
3.3. Related Work Network Architecture Search . . . . .	9
3.4. Analysis of Related Work . . . . .	9
3.5. Summary . . . . .	9
<b>4. Design</b>	<b>11</b>
4.1. Requirements and Assumptions . . . . .	11
4.2. System Overview . . . . .	11
4.2.1. Component 1 . . . . .	11
4.2.2. Component 2 . . . . .	11
4.3. Summary . . . . .	11
<b>5. Implementation</b>	<b>13</b>
5.1. Design Decisions . . . . .	13
5.2. Architecture . . . . .	13
5.3. Interaction of Components . . . . .	13
5.4. Summary . . . . .	13
<b>6. Evaluation</b>	<b>15</b>
6.1. Goal and Methodology . . . . .	15
6.2. Evaluation Setup . . . . .	15
6.3. Evaluation Results . . . . .	15
6.4. Analysis of Results . . . . .	15
<b>7. Conclusions</b>	<b>17</b>
7.1. Summary . . . . .	17
7.2. Contributions . . . . .	17
7.3. Future Work . . . . .	17
7.4. Final Remarks . . . . .	17
<b>Bibliography</b>	<b>17</b>
<b>Appendices</b>	<b>21</b>



---

## Abstract

---

The abstract goes here...





---

## 1 Introduction

---

Hint:

This chapter should motivate the thesis, provide a clear description of the problem to be solved, and describe the major contributions of this thesis. The chapter should have a length of about two pages!

---

### 1.1 Motivation

---

What is the motivation for doing research in this area?

---

### 1.2 Problem Statement and Contribution

---

What is the problem that should be solved with this thesis?

---

### 1.3 Outline

---

How is the rest of this thesis structured?



---

## 2 Background

---

### Hint:

This chapter should give a comprehensive overview on the background necessary to understand the thesis. The chapter should have a length of about five pages!

---

### 2.1 Basics of Neural Networks *WIP*

---

Neural networks are a part of most major AI-breakthrough in the last decade enabling computers to compete in fields formerly championed by humans.<sup>1</sup> They implement a statistical understanding of AI, which is to say that they try to find a specific model optimizing the likelihood of reproducing input-output pairs similar to some training data. The competing philosophy directly divines behaviour rules, frequently from expert knowledge, and as such is far less dependant from data. [citation needed] For the former concept its model classes are the essential point of design. A multitude of properties maybe sought after in a model class of which a few important ones are:

- **Richness:**

The diversity of single models in the class and thus the ability to fit a wide field of different input-output landscapes.<sup>2</sup>

If a model class is inherently restricted the underlying relation between inputs and outputs might simply be beyond the expressive capabilities of all its models.

In other words: If a model class is not rich enough all of its models will underfit the given training data.

- **Stability:**

Tendency of similar models in the class to handle inputs in a similar way.

If your model class shows unstable behavior defining a sensible way to search it for good models becomes difficult.

- **Interpretability of Models:**

Ease of formulating knowledge out of any given model in the class.

As fields exist in which statistical AI outperform experts the extraction of knowledge understandable and applicable by humans naturally is of interest.

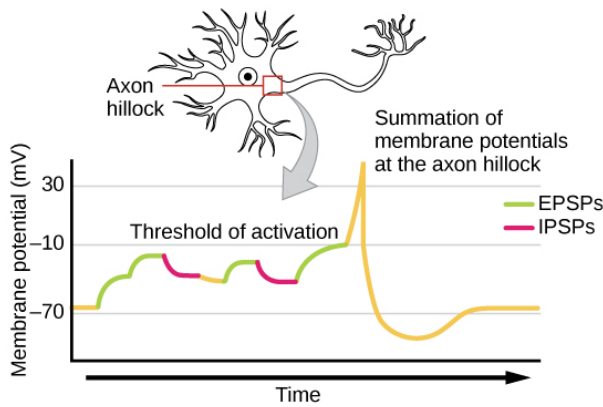
- [citation needed]

---

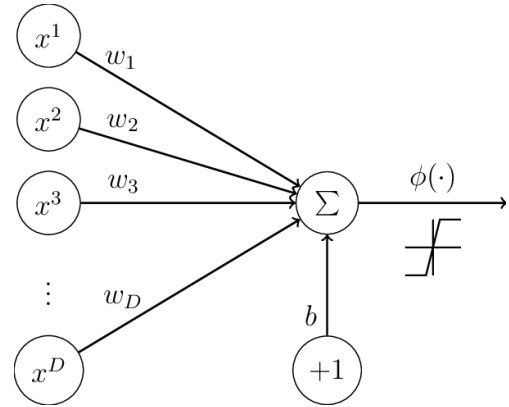
1

- 2011: "Watson" of IBM defeats two former grand champions in "Jeopardy!" [LF11]
- 2011: "Siri" enables users to use natural language to interact with their phones [Aro11]
- 2015: A convolutional neural network classifies images from the ImageNet dataset more accurately than human experts [RDS<sup>+</sup>15] [HZRS15]
- 2016: "AlphaGo" beats Lee Sedol, one of the world's strongest Go players [Gib16] [SSS<sup>+</sup>17]

<sup>2</sup> More formally the richness of a model class can be described as the amount of different functions from the input-space to the output-space which can be expressed through a model of said class.



**Figure 2.1.:** Representation of a biological Neuron



**Figure 2.2.:** Abstraction of a Neuron

If one knows an entity that already performs well on a given task it is a sensible approach to design ones model class to reproduce its decision process. Humans usually are such entities for many tasks of interest to AI research so they are a natural source of inspiration. Neural networks essentially are simplified models of a human central nervous system.

The most basic building block of the human central nervous system is a neuron which can receive multiple stimuli and is able to produce an output if the combined stimulation exceeds a threshold.[citation needed] One such neuron and its stimulus measure are depicted in Figure [2.1]. Another functionality observed in nature is the ability of a neuron to strengthen the connection to any source of stimulus thus giving said source more influence on whether the neuron produces an output. [citation needed]

The canonical mathematical model a neuron is defined as:

- **Inputs  $x_i$  :**  
All stimuli of a neuron are simply referred to as its inputs
- **Weights  $w_i$  :**  
The ability to assign importances is modelled as weights which are coupled to specific stimuli
- **Combined Weighted Inputs  $\sum_{i=1}^n w_i x_i$  :**  
After the inputs are scaled by their according weight they superpose to form the total excitation of the neuron
- **Activation Function  $\Phi(\sum_{i=1}^n w_i x_i)$  :**
- **Bias  $b$  :**  
As the threshold is implicitly modelled in the activation function...

seen in Figure 2.2

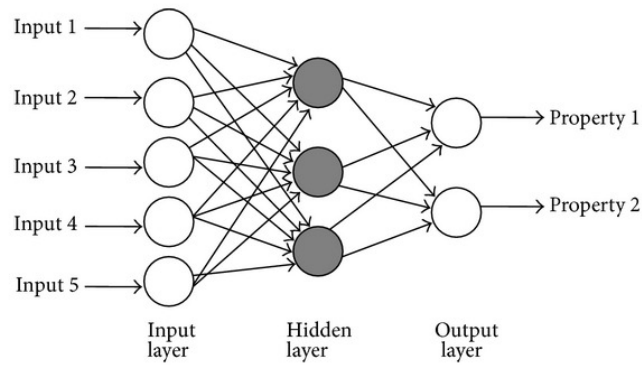
As the individual neurons is too simple to model any complex relation between inputs and outputs the next step is to aggregate multiple neurons. Figure 2.3 displays a few neurons coming together to form a simple fully-connected feed-forward network.<sup>3</sup>

- **TODO:**
- Issue of computational expense
- CNNs (and other forms of NN?)

## 2.2 The Lottery Ticket Hypothesis

- **TODO:**

<sup>3</sup> Inputs of neural networks are often called "features" and fully-connected networks are frequently referred to as "dense"



**Figure 2.3.:** A small fully-connected network

- Issue of overloading on parameters
- Clarifying the task (image classification)
- Idea of trainable subnets

---

## 2.3 Basics of Natural Language Processing

---

- **TODO:**
- (?) Corpora
- Tokenizing
- Language Models

---

## 2.4 Combining LMs & CNNs

---

- **TODO:**
- Interpreting the tensor representation of a sentence/document as an image to be classified
- (?) Validation through results
- (?) Handling different sizes of inputs
- (?) Handling missing words in the Language model



---

## 3 Related Work

---

### Hint:

This chapter should give a comprehensive overview on the related work done by other authors followed by an analysis why the existing related work is not capable of solving the problem described in the introduction. The chapter should have a length of about three to five pages!

---

### 3.1 Lottery-Ticket-Hypothesis

---

- **TODO:**
- Emergence of safe "return points" after a few training epochs
- Ablation study

---

### 3.2 Related Work Dynamic Pruning

---

---

### 3.3 Related Work Network Architecture Search

---

---

### 3.4 Analysis of Related Work

---

---

### 3.5 Summary

---





---

## 4 Design

---

### Hint:

This chapter should describe the design of the own approach on a conceptional level without mentioning the implementation details. The section should have a length of about five pages.

---

### 4.1 Requirements and Assumptions

---

---

### 4.2 System Overview

---

---

#### 4.2.1 Component 1

---

---

#### 4.2.2 Component 2

---

---

### 4.3 Summary

---



---

## 5 Implementation

---

### Hint:

This chapter should describe the details of the implementation addressing the following questions:

1. What are the design decisions made?
2. What is the environment the approach is developed in?
3. How are components mapped to classes of the source code?
4. How do the components interact with each other?
5. What are limitations of the implementation?

The section should have a length of about five pages.

---

### 5.1 Design Decisions

---

---

### 5.2 Architecture

---

---

### 5.3 Interaction of Components

---

---

### 5.4 Summary

---



---

## 6 Evaluation

---

### Hint:

This chapter should describe how the evaluation of the implemented mechanism was done.

1. Which evaluation method is used and why? Simulations, prototype?
2. What is the goal of the evaluation? Comparison? Proof of concept?
3. Which metrics are used for characterizing the performance, costs, fairness, and efficiency of the system?
4. What are the parameter settings used in the evaluation and why? If possible always justify why a certain threshold has been chosen for a particular parameter.
5. What is the outcome of the evaluation?

The section should have a length of about five to ten pages.

---

### 6.1 Goal and Methodology

---

---

### 6.2 Evaluation Setup

---

---

### 6.3 Evaluation Results

---

---

### 6.4 Analysis of Results

---



---

## 7 Conclusions

---

### Hint:

This chapter should summarize the thesis and describe the main contributions of the thesis. Subsequently, it should describe possible future work in the context of the thesis. What are limitations of the developed solutions? Which things can be improved? The section should have a length of about three pages.

---

### 7.1 Summary

---

---

### 7.2 Contributions

---

---

### 7.3 Future Work

---

---

### 7.4 Final Remarks

---





---

## Bibliography

---

- [Aro11] Jacob Aron. How innovative is apple’s new voice assistant, siri? *New Scientist*, 212(2836):24, 2011.
- [Gib16] Elizabeth Gibney. Google ai algorithm masters ancient game of go. *Nature News*, 529(7587):445, 2016.
- [HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [LF11] Adam Lally and Paul Fodor. Natural language processing with prolog in the ibm watson system. *The Association for Logic Programming (ALP) Newsletter*, 2011.
- [RDS<sup>+</sup>15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.
- [SSS<sup>+</sup>17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.



---

# Appendices

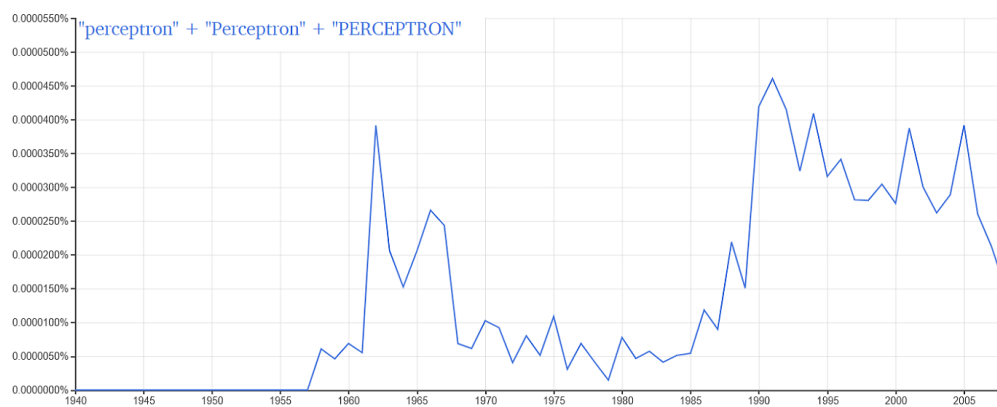


---

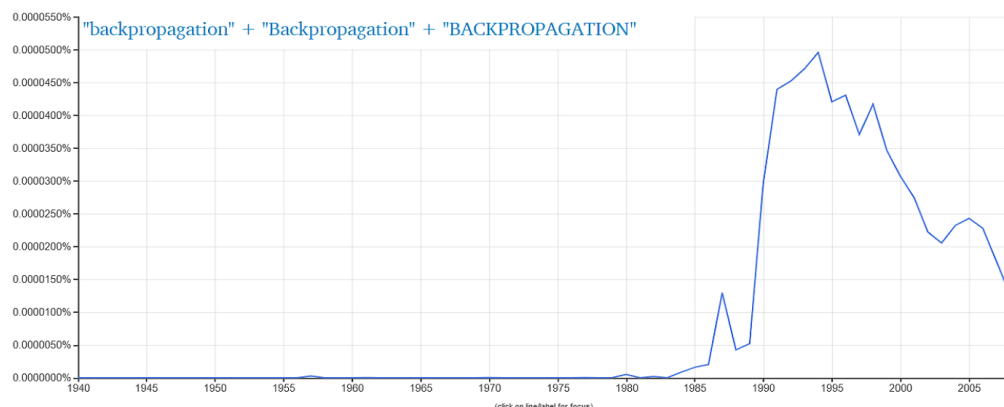
## A A history of neural networks

---

- 1. wave: 1955-1970
- 2. wave: 1985-2000 ?
- 3. wave: ???



**Figure A.1.:** Relative amount of occurrences of the word "Perceptron" in published books between 1940 and 2009



**Figure A.2.:** Relative amount of occurrences of the word "Backpropagation" in published books between 1940 and 2009