

# Machine Learning und tiefe neuronale Netze mit TensorFlow

DAVID BAUMGARTNER



BACHELORARBEIT

Nr. XXXXXXXXXXXX-B

eingereicht am  
Fachhochschul-Bachelorstudiengang

Software Engineering

in Hagenberg

im Januar 2017

Diese Arbeit entstand im Rahmen des Gegenstands

## Einführung in die Tiefere Problematik 1

im

Wintersemester 2016/17

Betreuer:

Mag. Pjotr I. Czar  
Creative Director

# Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 14. Januar 2017

David Baumgartner

# Inhaltsverzeichnis

<b>Erklärung</b>	<b>iii</b>
<b>Kurzfassung</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Allgemeines und Motivation . . . . .	1
1.2 Ziel der Arbeit . . . . .	1
<b>2 Begriffe im Maschinellen Lernen</b>	<b>2</b>
2.1 Data Science . . . . .	2
2.2 Machine Intelligence . . . . .	3
2.3 Machine Learning . . . . .	3
2.4 Neuronale Netzwerke . . . . .	4
2.5 Neuron . . . . .	4
2.6 Ebenen/Layer . . . . .	4
2.7 Informationen Merken . . . . .	4
2.8 Allgemeine Probleme . . . . .	4
2.8.1 Overfitting . . . . .	4
2.9 Domänenklassen . . . . .	4
2.9.1 Clustering . . . . .	4
2.9.2 Regression . . . . .	4
2.9.3 Klassifikation . . . . .	4
2.9.4 Vorhersage . . . . .	4
2.9.5 Robotics . . . . .	4
2.9.6 Computer Vision . . . . .	4
2.9.7 Optimierungsprobleme . . . . .	4
2.10 Neuronale Netzwerktypen . . . . .	4
2.10.1 Self-Organizing Map . . . . .	4
2.10.2 FeedForward . . . . .	4
2.10.3 Hopfield . . . . .	4
2.10.4 Boltzmann Machine . . . . .	4

2.10.5	Deep Belief Network . . . . .	4
2.10.6	Deep Feedforward . . . . .	4
2.10.7	NEAT . . . . .	4
2.10.8	CPPN . . . . .	4
2.10.9	HyperNEAT . . . . .	4
2.10.10	Convolutional neural network . . . . .	4
2.10.11	Elman Network . . . . .	4
2.10.12	Jordan Network . . . . .	4
2.10.13	Recurrent Network . . . . .	4
2.11	Tensorflow Typen Unterstützung . . . . .	4
<b>3</b>	<b>TensorFlow Bibliothek</b>	<b>5</b>
3.1	Python API . . . . .	6
3.2	C++ API . . . . .	6
3.3	Go API . . . . .	6
3.4	TensorFlow Python . . . . .	6
3.4.1	Graphs / Dataflowgraph . . . . .	6
3.4.2	Operation . . . . .	6
3.4.3	Tensor . . . . .	6
3.4.4	Operationen . . . . .	6
3.4.5	Probleme . . . . .	6
<b>4</b>	<b>Facial Keypoints Detection</b>	<b>7</b>
4.1	Ausgangssituation . . . . .	7
4.2	Vorbereitung . . . . .	7
4.2.1	Daten vorbereiten und normalisieren . . . . .	7
4.2.2	Evaluation- und Error-Funktion . . . . .	7
4.3	Neuronale Ebenen vorbereiten . . . . .	7
4.4	Neuronale Ebenen verknüpfen . . . . .	7
4.5	Trainieren . . . . .	7
4.6	Validierungsergebnisse . . . . .	7
	<b>Quellenverzeichnis</b>	<b>8</b>

# Kurzfassung

Maschinelles lernen und tiefe neuronale Netze werden unter anderem in unserem Jahrzehnt sehr häufig eingesetzt um technische Problemstellungen zu lösen, für des eine in vernünftiger Zeit keine Brauchbaren iterative Algorithmen gibt. Zusätzlich werden maschinell lernende System immer häufiger in unserem allgemeinen Alltag eingesetzt um uns zu unterstützen und um von den Benützern zu lernen. Ein neuronales Netzwerk kann aber nicht einfach erstellt werden und im nächsten Schritt in der Praxis eingesetzt werden. Dies würde zu erheblichen Problemen führen. Diese Netzwerke müssen trainiert werden sowie getestet.

Im Rahmen dieser Bachelorarbeit werden die wichtigsten theoretischen Konzepte zu maschinellem lernen und tiefe neuronale Netze theoretisch zu vergleichen und empirisch zu überprüfen. Dazu wird das TensorFlow-Bibliothek als Beispiel verwendet und analysiert. Aus dieser Bibliothek werden die benötigten Teile heraus genommen und in einem Python-Script zusammen gefügt um in Bildern mit Gesichtern gewisse Züge zu erkennen und zu Klassifizieren. Durch die Analysephasen werden theoretische und berechnete Annahmen untermauert oder in Frage gestellt. Folgende Schlussfolgerungen gehen jedoch nach Auswertung der Ergebnisse über die theoretischen Annahmen hinaus: Ist das TensorFlow-System in der Lage, Muster aus unterschiedlichen Datentypen, wie zum Beispiel Bilder, Videos oder Videostreams, zu erkennen und von diesen selbst zu lernen?

Die Bachelorarbeit ist sowohl für Studierende im Studium Software Engineering sowie Informatik als auch für Lehrende in diesen Bereichen interessant.

# Abstract

This should be a 1-page (maximum) summary of your work in English.

Im englischen Abstract sollte inhaltlich das Gleiche stehen wie in der deutschen Kurzfassung. Versuchen Sie daher, die Kurzfassung präzise umzusetzen, ohne aber dabei Wort für Wort zu übersetzen. Beachten Sie bei der Übersetzung, dass gewisse Redewendungen aus dem Deutschen im Englischen kein Pendant haben oder völlig anders formuliert werden müssen und dass die Satzstellung im Englischen sich (bekanntlich) vom Deutschen stark unterscheidet (mehr dazu in Abschn. ??). Es empfiehlt sich übrigens – auch bei höchstem Vertrauen in die persönlichen Englischkenntnisse – eine kundige Person für das „proof reading“ zu engagieren.

Die richtige Übersetzung für „Diplomarbeit“ ist übrigens schlicht *thesis*, allenfalls „diploma thesis“ oder „Master’s thesis“, auf keinen Fall aber „diploma work“ oder gar „dissertation“. Für „Bachelorarbeit“ ist wohl „Bachelor thesis“ die passende Übersetzung.

Übrigens sollte für diesen Abschnitt die *Spracheinstellung* in LaTeX von Deutsch auf Englisch umgeschaltet werden, um die richtige Form der Silbentrennung zu erhalten, die richtigen Anführungszeichen müssen allerdings selbst gesetzt werden (s. dazu die Abschnitte ?? und ??).

# Kapitel 1

## Einleitung

### 1.1 Allgemeines und Motivation

Warum - wieso - wesshalb?

Um in unserern

Dieses Dokument ist als vorwiegend technische Starthilfe für das Erstellen einer Masterarbeit (oder Bachelorarbeit) mit LaTeX gedacht und ist die Weiterentwicklung einer früheren Vorlage<sup>1</sup> für das Arbeiten mit Microsoft *Word*. Während ursprünglich daran gedacht war, die bestehende Vorlage einfach in LaTeX zu übernehmen, wurde rasch klar, dass allein aufgrund der großen Unterschiede zum Arbeiten mit *Word* ein gänzlich anderer Ansatz notwendig wurde. Dazu kamen zahlreiche Erfahrungen mit Diplomarbeiten in den nachfolgenden Jahren, die zu einigen zusätzlichen Hinweisen Anlass gaben.

Das vorliegende Dokument dient einem zweifachen Zweck: *erstens* als Erläuterung und Anleitung, *zweitens* als direkter Ausgangspunkt für die eigene Arbeit. Angenommen wird, dass der Leser bereits über elementare Kenntnisse im Umgang mit LaTeX verfügt. In diesem Fall sollte – eine einwandfreie Installation der Software vorausgesetzt – der Arbeit nichts mehr im Wege stehen. Auch sonst ist der Start mit LaTeX nicht schwierig, da viele hilfreiche Informationen auf den zugehörigen Webseiten zu finden sind (s. Kap. 1).

### 1.2 Ziel der Arbeit

uiaae

---

<sup>1</sup>Nicht mehr verfügbar.



## Kapitel 2

# Begriffe im Maschinellen Lernen

Diese Erklärung der Begriffe und Elemente verfolgt zwei Ziele. Zum einen stellt dies Grundlage des gesamten Themas dar und soll für Interessierte die nicht so vertraut sind, eine Einführung in die Thematik bieten. Und zum anderen werden viele dieser Begriffe erläutert, welche noch häufig zum Einsatz kommen (u.A. Neuron, Aktivierungsfunktion, ...).

### 2.1 Data Science

Data Science wird generell als die Extraktion von Wissen aus Daten bezeichnet. Dabei werden die Teilbereiche, Statistik und Mathematik, Computer Science und Machine Learning sowie einige weiter, in diesem Begriff zusammen gefasst. Das Gebiet für sich, wird auch als Berufstätigkeit bezeichnet, wobei meist spezialisierte Formen für die Berufsbezeichnung verwendet wird.

Damit Wissen aus Daten überhaupt extrahiert werden kann, muss ein ganzer Prozess durchlaufen werden. Dieser beginnt mit dem zusammentragen von Rohdaten aus der Realität, welche aber zu diesem Zeitpunkt noch keinen Zusammenhang offenbaren. Im zweiten Prozessschritt werden diese Daten meist umgebaut und neu sortiert wobei dieser Schritt nicht immer erforderlich ist. Auf die zurecht gelegten Daten besteht nun die Möglichkeit Modelle, Algorithmen sowie weitere Extraktionen durchzuführen. Die erneut extrahierten Daten werden in dem zweiten Prozessschritt als Ausgangsdaten verwendet. Auf die Daten ausgeführte Modelle und Algorithmen liefern Ergebnisse die Visuell dargestellt für eine größer Gruppe an Personen geeignet ist. Aus den gelernten Wissen besteht zusätzlich die Möglichkeit diese zum Generieren von neuen Daten zu verwenden um neu Modelle zu entwickeln die zum Beispiel Vorgänge in der Natur noch akkurater wieder spiegeln.

## 2.2 Machine Intelligence

Maschine Intelligenz ist ein Begriff der so an sich noch nicht Definiert ist. Einige namhafte Unternehmen wie Google Inc. und Microsoft Corporation usw. bieten jeweils unterschiedliche Definitionen oder Beschreibungen. Dieser wird aber von allen ähnlich beschrieben und definiert. Es bezeichnet einen Überbegriff über das gesamte Gebiet mit Machine Learning, Künstlicher Intelligenz, Konversationsintelligenz und alle Themen die in näherer Beziehung dazu stehen.

## 2.3 Machine Learning

Machine Learning definiert eine große Anzahl an Theorien und Umsetzungen von nicht explizit programmierten Abläufen. Diese wurden aus Studien in den Bereichen der Mustererkennung und Rechnerische Lerntheorien mit Künstlicher Intelligenz teilweise entwickelt. Dieses Gebiet umfasst im Jahr 2016 aber sehr viel mehr. So existieren zusätzlich Ansätze aus dem Bereich der Biologie wie zum Beispiel Neuronale Netzwerke die dem Gehirn nachempfunden sind und Genetische Algorithmen die der Weiterentwicklung eines Lebewesen ähnelt. Ein ganz andere Zugang wurde in der Sowjetunion verfolgt, mit sogenannte 'Support Vektor Machines'.

## **2.4 Neuronale Netzwerke**

## **2.5 Neuron**

## **2.6 Ebenen/Layer**

## **2.7 Informationen Merken**

## **2.8 Allgemeine Probleme**

### **2.8.1 Overfitting**

## **2.9 Domänenklassen**

### **2.9.1 Clustering**

### **2.9.2 Regression**

### **2.9.3 Klassifikation**

### **2.9.4 Vorhersage**

### **2.9.5 Robotics**

### **2.9.6 Computer Vision**

### **2.9.7 Optimierungsprobleme**

## **2.10 Neuronale Netzwerktypen**

### **2.10.1 Self-Organizing Map**

### **2.10.2 FeedForward**

### **2.10.3 Hopfield**

### **2.10.4 Boltzmann Machine**

### **2.10.5 Deep Belief Network**

### **2.10.6 Deep Feedforward**

### **2.10.7 NEAT**

### **2.10.8 CPPN**

### **2.10.9 HyperNEAT**

### **2.10.10 Convolutional neural network**

### **2.10.11 Elman Network**

### **2.10.12 Jordan Network**

### **2.10.13 Recurrent Network**

## **2.11 Tensorflow Typen Unterstützung**



## Kapitel 3

# TensorFlow Bibliothek

### 3.1 Python API

### 3.2 C++ API

### 3.3 Go API

### 3.4 TensorFlow Python

#### 3.4.1 Graphs / Dataflowgraph

#### 3.4.2 Operation

#### 3.4.3 Tensor

#### 3.4.4 Operationen

Konstanten, Zufallswerte

Variables

Transformationen

Mathematik

Flusskontrolle

Images / FFmpeg

Input und Readers

Neural Network

Running Graphs

Training

#### 3.4.5 Probleme

NaN Problem

## Kapitel 4

# Facial Keypoints Detection

### 4.1 Ausgangssituation

### 4.2 Vorbereitung

#### 4.2.1 Daten vorbereiten und normalisieren

#### 4.2.2 Evaluation- und Error-Funktion

### 4.3 Neuronale Ebenen vorbereiten

### 4.4 Neuronale Ebenen verknüpfen

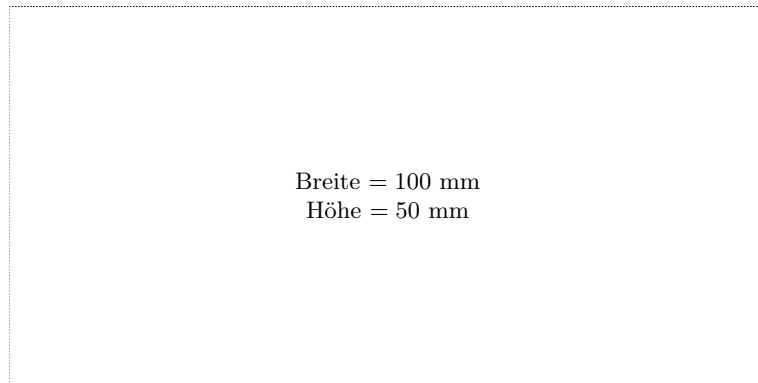
### 4.5 Trainieren

### 4.6 Validierungsergebnisse

# Quellenverzeichnis

# Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —