Machine Learning und tiefe neuronale Netze mit TensorFlow

David Baumgartner

BACHELORARBEIT

Nr. XXXXXXXXXX-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

Kapitel 1

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, ...).

1.1 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 bestecht 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.

1.2 Machine Intelligence

Maschine Intelligenz ist ein Begriff der so an sich noch nicht Definiert ist. Einige namhafte Unternehmen wie Google Inc. und Microsoft Corporation 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.

1.3 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 nach empfunden sind und Genetische Algorithmen die der Weiterentwicklung eines Lebewesen ähneln. Ein ganz andere Zugang wurde in der Sowjetunion verfolgte, mit sogenannte 'Support Vektor Machines', bei welchen man einen rein mathematischen Ansatz verfolgte.

1.4 Neuronale Netzwerke

Neuronale Netzwerke auch bekannt im Jahre 2016 unter dem Begriff 'Tiefes Lernen'.

Die Theorie und die ersten Grundlagen wurden im Jahre 1943 von Warrn McCulloch und Walter Pitts geschaffen, die ein Modell entwickelten. Dieses basierte auf Mathematik und Algorithmen und führte zur 'Threshold Logik'. Durch den 'Backpropagation'-Algorithmus im Jahre 1975, wurden es möglich Netzwerke zu trainieren welche mehrere Ebenen hatten.

Neuronale Netzwerke bestehen aus Neuronen die mit einander Verbunden sind und gemeinsam ein Netzwerk ergeben welches dem biologischen Gehirn nachempfunden ist. Die Verbindungen sind nicht fest vorgegeben sondern können auch zum Beispiel Schleife bilden.

TODO: you can do anything with it, if it can be designed as a function

1.5 Neuron

Abbildung (Aufbau eines Neurons)

Ein Neuron ist wie eine Zell in einem biologischen Gehirn nachempfunden und besteht aus:

Informationseingangsstrom ist der Dateneingang, wobei ein Neuron ein bis theoretisch unendlich viele solcher Eingänge haben könnte. Dies hängt von der jeweiligen Architektur des Netzwerks ab.

Informationsgewichtung bezeichnet die Gewichtung mit der der Eingangsstrom gewertet wird. Dies wird Gewichtung genannt. So wird ein Informationseingangsstrom mehr berücksichtigt oder nicht. Diese Gewichtung wird durch den Backpropagation-Algorithmus angepasst und nachjustiert.

Kernfunktion bewirkt das Verarbeiten der gewichteten Informationseingänge. Im einfachsten Fall werden alle Werte summiert zu einem Wert. Im Grunde ist es aber möglich jegliche Berechnung hier einfließen zu lassen, je nach Beinötigung.

Aktivierungsfunktion stellt den Ausgang eines Neuron dar. Dabei wird eine weitere Funktion auf das im Kern berechnete Ergebnis ausgeführt und führt dazu, dass ein Ergebnis noch stärker ausgeprägt weiter gegeben wird oder minimiert wird. Diese Aktivierungsfunktion ist meist die Sigmoid-Funktion oder eine Lineare-Funktion. Wobei in den vergangen Jahren die Linearen-Funktionen häufiger zum Einsatz kommen.

1.6 Bias Neuron

Ein Bias definiert einen spezial Fall eines Neurons, welches weder Dateneingänge somit auch keine Gewichtung hat, sowie keine Berechnung im Kern durchführt. Dieses liefert nur einen Konstanten Wert aus wie zum Beispiel ein Eins. Durch die konstante Auslieferung wird auch die Aktivierungsfunktion überflüssig. Das Bias Neuron stellt somit einen konstanten Wert für das Netzwerk dar, beziehungsweise für die darauffolgende Ebene.

1.7 Ebenen/Layer

Ebenen sind Zusammenschlüsse von Neuronen welche sich auf der selben Ebene befinden. Diese Neuronen sind aber nicht mit einander Verbunden, sondern bekommen Daten aus der Ebene davor und geben diese an die darauffolgende Ebene weiter. Diese Art wird **Hiddenlayer** genannt. Jedes Netzwerk benötigt zusätzlich jeweils zwei spezial Ausprägungen an Ebenen. Diese sind:

Inputlayer sind Ebenen die keine vorgeschaltete Ebene haben. Diese Ebene nimmt die Daten ohne Gewichtung auf und gibt sie verarbeitet an die darauffolgende Ebene weiter. In gewisser Weise stellt es einen Übergang zwischen der Welt außerhalb und des Neuronalen Netzwerkes dar.

Outputlayer befindet sich am Ende eines Netzwerkes. Dieser Layer hat die Aufgabe die Daten nach außerhalb weiter zu geben anstatt an das darauffolgende Netzwerk. Hierbei werden die Informationen meist nur mehr für die Ausgabe hergerichtet. So wird zum Beispiel die Wahrscheinlich für jeder möglichen Ausgänge berechnet.

Abbildung (Aufbau eines Einfachen FeedForward NN)

1.8 Informationen Merken und wieder Erkennung

Durch das Anpassen der Gewichtungen bei jedem Dateneingangsstrom mit Hilfe des Backpropagation-Algorithmus wird es möglich Zustände zu speichern und diese auch zu Merken. Dies führt dazu, dass sollte ein ähnlicher Dateneingang statt finden, wo zuvor schon ein ähnlicher in einem Trainingszyklus vorhanden war, dieser ähnlich wieder erkennt werden kann. Dieser könnte somit möglicherweise zu der selben Kategorie gehört wie der schon zuvor schon bekannte gemachte und gelernte Dateneingang.

1.9 Backpropagation

TODO: relative genau

1.10 Allgemeine Probleme

Neuronale Netzwerke sind nicht Fehler und Problem los und sind nicht die Lösung auf jegliche Frage die es auf der Welt gibt. Es existieren bekannte Probleme für die es Lösungen oder Lösungsansätze gibt. Es existieren aber auch noch genügend andere Probleme, welche noch nicht festgestellt wurden oder als solche noch nicht erkannt worden sind.

Diese Arbeit wird auf die bekanntesten Probleme eingehen und beschreiben und auch Lösungen wieder geben.

1.10.1 Overfitting

Overfitting bezeichnet ein Problem welches nicht nur Machine Intelligence betrifft sondern auch Menschen und andere Lebewesen. Zum Beispiel ein Student lernt auf eine Prüfung und ist im Besitz einer Klausur aus einem

Vorjahr. Nach öfteren durchspielen der Fragen und sich selbst testen, befindet er sich in der Lage diese Klausur mit einer sehr hohen Wahrscheinlichkeit zu bestehen. Dabei hat sich die Klausur in seinem Gehirn eingeprägt aber nicht das Stoffgebiet zu welchem eine Klausur schreiben muss. Das Problem wird als Overfitting bezeichnet und beschreibt das etwas gemerkt wurde aber nicht gelernt worden ist und somit eine Abwandlung nicht wieder erkannt werden kann.

1.11 Trainieren

1.11.1 Supervised Trainieren

1.11.2 Unsupervised Trainieren

1.12 Domänenklassen

Neuronale Netzwerk können sehr vielseitig eingesetzt werden. Grundsätzlich lässt sich jedes Problem, welches als Funktion repräsentiert werden kann, durch ein Neuronales Netzwerk approximieren.

In dieser Arbeit werde ich sieben Hauptdomänen erklären und beschreiben, in welchen Neuronale Netzwerke öfters zum Einsatz kommen und eingesetzt werden.

1.12.1 Clustering

Clustering Problem bezeichnen das Einordnen von Daten in Klassen oder Gruppierungen. Diese Gruppierungen können von einem Netzwerk selbst definiert werden oder festgelegt in wie viele Gruppen die Daten eingeordnet werden sollen. Diese Netzwerke agieren meist unsupervised und besitzen meist die Möglichkeit sich nach dem Trainieren, sich noch selbst weiter anzupassen. Dies erlaubt dem Netzwerk auf sich ändernde Daten anzupassen und das Clustern und Gruppieren weiter zu führen ohne ein neues Netzwerk erstellen zu müssen.

Kapitel 2

TensorFlow Bibliothek

2.1 Python API

2.2 C++ API

2.3 Go API

2.4 TensorFlow Python

2.4.1 Graphs / Dataflowgraph

2.4.2 Operation

2.4.3 Tensor

2.4.4 Operationen

Konstanten, Zufallswerte

Variables

Transformationen

Mathematik

Flusskontrolle

Images / FFmpeg

Input und Readers

Neural Network

Running Graphs

Training

2.4.5 Probleme

NaN Problem

Kapitel 3

Facial Keypoints Detection

3.1 Ausgangssituation

3.2 Vorbereitung

3.2.1 Daten vorbereiten und normalisieren

3.2.2 Evaluation- und Error-Funktion

3.3 Neuronale Ebenen vorbereiten

3.4 Neuronale Ebenen verknüpfen

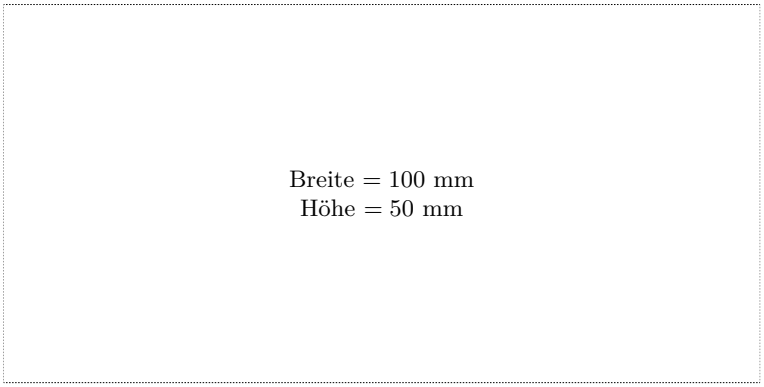
3.5 Trainieren

3.6 Validierungsresultate

Quellenverzeichnis

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —