



Fakultät Informatik

Marco Philipp, matr
Niklas Röske, 70456600

Python Machine Learning: TensorFlow

Betreuer:
Prof. Dr. Claus Fühner

Salzgitter

Suderburg

Wolfsburg

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere, dass ich alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe, und dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist.

Wolfenbüttel, den 19. Januar 2020

Kurzfassung

Hier sollte eine halbseitige Kurzfassung der Arbeit stehen.

Abstract

Here, an abstract written in English should appear.

Inhaltsverzeichnis

Abkürzungsverzeichnis	VI
1. Einleitung	1
2. Maschinelles Lernen	2
2.1. Ziel	2
2.2. Aufbau	2
2.3. Funktionsweise	2
2.4. Anwendungsbereiche	2
3. TensorFlow	3
3.1. Allgemeines	3
3.2. Unterschiede zu anderen Frameworks	4
4. Arbeitsweise	5
4.1. Tensoren	5
4.1.1. Mathematische Definition	5
4.1.2. Tensoren in TensorFlow	5
4.2. Operationen	5
4.2.1. Typen	5
4.2.2. Beispiele	6
4.3. Graphen	6
4.3.1. Aufbau	6
4.3.2. Funktionsweise	6
4.3.3. Darstellung von Operationen	7
4.3.4. Vor- und Nachteile von Graphen zur Berechnung	7
4.4. Trainieren des Netzes	7
4.4.1. Eingaben	7
4.4.2. Ablauf	7
4.4.3. Ergebnis	7
5. Praktisches Beispiel	8
5.1. Zielsetzung	8
5.2. Planung	8
5.3. Datensätze	8
5.4. Erstellen des Netzes	8
5.5. Ergebnisse auswerten	9

5.6. Probleme	9
6. Fazit	10
6.1. Vor- und Nachteile von TensorFlow	10
6.2. Ausblick	10
6.2.1. TensorFlow Lite	10
Literaturverzeichnis	11
A. Anhang A	12

Abbildungsverzeichnis

Abkürzungsverzeichnis

1. Einleitung

MARCO PHILIPP

2. Maschinelles Lernen

MARCO PHILIPP

2.1. Ziel

Was ist die Motivation hinter dieser Arbeit

2.2. Aufbau

Ziele der Arbeit

2.3. Funktionsweise

Aufbau der Arbeit. Sehr kurz. Kann ist kein muss.

2.4. Anwendungsbereiche

Was ist die Motivation hinter dieser Arbeit

3. TensorFlow

NIKLAS RÖSKE

Worum geht es in Ihrer Arbeit [1] Insgesamt 2 Seiten

3.1. Allgemeines

"Die Daten fließen als Tensoren durch einen Graphen von Rechenoperationen, aus denen unser Deep-Learning-Netz besteht." [1]

Berechnung als Datenflussgraph

Knoten Darstellung von Operationen

Kanten Daten (Tensoren)

Berechnungen mit Datenflussgraphen

für maschinelles Lernen entwickelt

Portierbarkeit (unterschiedliche Hardware)

Hauptkomponenten in C++

Python am weitesten entwickelt und am meisten genutzt

Flexibilität, Modelle sind einfach zu erstellen

viele Optimierungsverfahren, der Nutzer wird bei Verwendung dieser unterstützt

Prozess kann durch Tensorboard beobachtet werden

Abstraktionsbibliotheken, wie Keras vereinfachen Benutzung Modelle können schneller in den Betrieb gehen

Modell für einzelnen Prozessor kann durch wenige Änderungen auf Prozessorclustern laufen

Hier wird auf die Textstelle 4.1 verwiesen, die sich auf der Seite 5 befindet.

3.2. Unterschiede zu anderen Frameworks

4. Arbeitsweise

4.1. Tensoren

NIKLAS RÖSKE

Insgesamt 1-2 Seiten

4.1.1. Mathematische Definition

Bild 5.9 Deep-Learning Seite 136 Ein Tensor ist eine Bezeichnung für ein n -dimensionales Feld. In der Abbildung sind unterschiedliche Dimensionen von Feldern dargestellt. "Beispielsweise ist ein 1×1 -Tensor ein Skalar, ein $1 \times n$ -Tensor ein Vektor, ein $n \times n$ -Tensor eine Matrix, und ein $n \times n \times n$ -Tensor ist einfach ein dreidimensionales Feld"[1].

4.1.2. Tensoren in TensorFlow

In TensorFlow hingegen werden die Daten, die durch den Graphen fließen Tensoren genannt. Da diese Daten wie die Tensoren in der Mathematik verschiedene Dimensionen annehmen können, wie zum Beispiel einfache Zahlenwerte oder dreidimensionale Felder eines Bildes, basiert der Begriff auf dem mathematischen Begriff. Tensor-Objekte entstehen bei der Ausführung einer Operation.

4.2. Operationen

MARCO PHILIPP

Ziele der Arbeit

4.2.1. Typen

Was ist die Motivation hinter dieser Arbeit

4.2.2. Beispiele

Was ist die Motivation hinter dieser Arbeit

4.3. Graphen

NIKLAS RÖSKE

Insgesamt 3 Seiten

4.3.1. Aufbau

Wie schon in Kapitel 3.1 erwähnt wurde, arbeitet TensorFlow auf einem Datenflussgraphen. Dabei handelt es sich um ein gerichtetes Diagramm, das ein mathematisches Problem darstellt. Diese Datenflussgraphen bestehen aus Tensoren 4.1 und Operationen 4.2, die miteinander verbunden sind. Durch eine Aneinanderreihung von unterschiedlichen Operationen mit den dazugehörigen Tensoren als Werte entsteht ein Datenflussgraph. Dieser wird als mathematische Grundlage für das Neuronale Netz genutzt.

4.3.2. Funktionsweise

Hier Bild 5.11 aus dem Buch Deep-Learning einfügen.

Die Abbildung zeigt eine einfache Berechnung als Datenflussgraphen, um den Ablauf zu verdeutlichen. Die Berechnung eines Graphen kann nur in einem `tf.Session()`-Block ausgeführt werden. Am Anfang dieses Blocks wird zusätzlich eine `tf.Session()` Variable angelegt. Dies wird durch den Ausdruck `with tf.Session() as sess` realisiert. Das `Session()`-Objekt wird später benötigt, um diese zu starten. Bevor die `Session` gestartet wird, werden die Konstanten, oder auch Tensoren, `a`, `b`, `c` und `d` erstellt und ein Wert zugewiesen. Die letzte Operation wurde dem Tensor `result` in der Form `result = tf.sqrt(x3, "Wurzel")` zugewiesen. Für den Start muss

der Operation der Tensor übergeben werden, der berechnet werden soll. Diese Operation muss zusätzlich einer Variablen übergeben werden, damit der Wert weiter verwendet werden kann. In diesem Fall sieht die Funktion wie folgt aus: `res = sess.run(result)`. Über die Variable `res` kann nun das Ergebnis ausgegeben werden.

Hier Bild 3-4 aus Buch Einführung einfügen Vor der Ausführung eines *Session*-Blocks enthalten die wird durch das Aufrufen einer Operation nur eine Instanz dieses Objekts erstellt. Dieses Objekt enthält nur Referenz der Daten und noch keine richtigen Daten. Dies ändert sich mit dem Start der Ausführung und dem Durchlaufen der Operationen. Wie in der Abbildung zu sehen ist werden die Referenzen der Tensor-Objekte in A durch Tensor-Objekt-Instanzen ersetzt.

4.3.3. Darstellung von Operationen

Im Kapitel 4.2 wurde im Allgemeinen auf die Operationen in TensorFlow eingegangen. Dies wird nun noch für die Darstellung vertieft.

4.3.4. Vor- und Nachteile von Graphen zur Berechnung

Durch die Verwendung von Graphen ist es möglich einzelne Operationen ohne große Umstände auszutauschen.

4.4. Trainieren des Netzes

NIKLAS RÖSKE

Insgesamt 1-2 Seiten

4.4.1. Eingaben

Für das Trainieren eines Neuronalen Netzes werden, wie auch bei anderen Frameworks Trainingsdaten benötigt. Je Was ist die Motivation hinter dieser Arbeit

4.4.2. Ablauf

Was ist die Motivation hinter dieser Arbeit

4.4.3. Ergebnis

Was ist die Motivation hinter dieser Arbeit

5. Praktisches Beispiel

Worum geht es in Ihrer Arbeit

5.1. Zielsetzung

MARCO PHILIPP

5.2. Planung

NIKLAS RÖSKE

1 Seite

5.3. Datensätze

MARCO PHILIPP

5.4. Erstellen des Netzes

NIKLAS RÖSKE

1-2 Seiten

5.5. Ergebnisse auswerten

MARCO PHILIPP

5.6. Probleme

NIKLAS RÖSKE

1 Seite

6. Fazit

NIKLAS RÖSKE

Insgesamt 1-2 Seiten

6.1. Vor- und Nachteile von TensorFlow

6.2. Ausblick

6.2.1. TensorFlow Lite

Literaturverzeichnis

- [1] I. L. Tom Hope, Yehezkel S. Resheff, *Einführung in TensorFlow Deep-Learning-Systeme programmieren, trainieren, skalieren und deployen*. O'REILLY, 2018.

A. Anhang A