|  |  |  |
| --- | --- | --- |
| **MA 3062** | | |
| Untersuchung und prototypische Umsetzung eines Lifelong Deep Neural Network Algorithmus | | |
| **Simon Kamm** | | |
|  | | |
| **Evaluierungsspezifikation** | | |
|  | Prüfer: | Prof. Dr.-Ing. Michael Weyrich |
|  | Betreuer: | Benjamin Maschler, M.Sc. |
| Start: 29.04.2019 | | Abgabe: 29.10.2019 |
|  | |  |

**Dokument Versionsverwaltung**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Autor | QS | Datum | Status | Änderungen |
| 0.1 | Kamm |  | 24.06.19 | in Bearb. | Erstellung |
|  |  |  |  |  |  |

*Allgemeine Hinweise zur Benutzung von IAS-VM Dokumentvorlagen*

*Diese Hinweise und Erläuterungen in den einzelnen Kapiteln der Vorlage sind kursiv dargestellt und müssen im Dokument gelöscht werden.*

*Verwendete Symbole:*

*\* generisch (kann mehrfach vorkommen)*

*[...] optional (kann auch weggelassen werden)*

*<...> Platzhalter (Bereich inklusive Klammern muss entsprechend ersetzt werden)*

***Bitte ändern Sie die Formatierung des Deckblatts nicht sondern tragen Sie nur die abgefragten Daten ein!***

*Bei der Erstellung eines Dokumentes hat das Dokument zunächst den Status ‘in Bearb.’ und die Versionsnummer 0.1. Dokumente im Status ‘in Bearb.’ können beliebig geändert, gespeichert und gedruckt werden. Wichtige Änderungen sollten allerdings im Feld „Änderungen“ dokumentiert werden. Die erste fertige Version bekommt die Versionsnummer 1.0. Ab der Version 1.0 wird das Speichern im Zustand ‘vorgelegt’ erlaubt. Dokumente mit niedrigerer Versionsnummer dürfen nicht mit ‘vorgelegt’ gespeichert werden.* ***Diese Informationen müssen manuell eingetragen werden.***

*Dokumente unterliegen ab dem Status ‘vorgelegt’ dem Konfigurationsmanagement und dürfen nicht mehr überschrieben werden. Nach einer Änderung muss das Dokument unter Angabe der durchgeführten Änderungen mit neuer Versionsnummer gespeichert werden. Die Versionsnummern muss dabei um 0.1 hochgezählt werden.*

*Bei der Erstellung eines Dokuments ist folgendes zu beachten:*

* *Sind die allgemeinen Anforderungen zur Evaluierung der Konzeption definiert?*
* *Sind die ausgewählten Evaluierungsmethoden auf die Konzeption anwendbar?*
* *Wurden die Kriterien so ausgewählt, dass eine sinnvolle Ergebnisbewertung erfolgen kann?*
* *Sind die Evaluierungsfälle vollständig spezifiziert? (Was wird geprüft, Ausgangssituation, Eingaben, erwartete Ergebnisse)*
* *Decken die Evaluierungsfälle die im Pflichtenheft spezifizierten Anforderungen ab?*

# Inhaltsverzeichnis

0 Inhaltsverzeichnis 2

1 Anforderung an die Evaluierung 4

2 Methoden der Evaluierung 4

3 Evaluierungskriterien 5

4 Evaluierungsfälle 5

# Anforderung an die Evaluierung

In dieser Arbeit wird der L DNN Algorithmus untersucht und seine Performanz bewertet. Um eine belastbare Aussage über den Algorithmus treffen zu können, müssen Tests definiert werden, welche zur Evaluierung genutzt werden. Für eine generelle Aussage werden Tests definiert und genutzt, die auch für vergleichbare Algorithmen genutzt werden. In dieser Arbeit werden dafür öffentlich zugängliche Bilddatensätze genutzt. Zur Bewertung dieser Testfälle werden bekannte Metriken der Objekterkennung/-Klassifizierung genutzt.

# Methoden der Evaluierung

*Hier wird die Methode der Evaluierung aufgeführt.*

*Im Falle der Erstellung eines Prototyps soll dieser zur Evaluierung herangezogen werden.*

*Festzuschreiben sind die Art und Weise der Ergebnissicherung und -auswertung, insbesondere im Hinblick auf Wiederholung von Evaluierungen. Es wird geklärt, welche Daten während und nach der Prüfung wie festzuhalten sind.*

Zur Evaluierung werden öffentlich zugängliche Bilddatensätze genutzt, welche sehr häufig im Bereich DNNs zur Bestimmung des Potenzials eines Algorithmus genutzt werden. Zur Auswertung der Performanz wird die Klassifikationsgenauigkeit bewertet. In dieser Arbeit wird der MNIST- und ImageNet-Datensatz zur Evaluierung genutzt. Dabei werden dem Netzwerk neue, bisher unbekannte Klassen gezeigt, welche das Netzwerk auf Basis weniger Beispielbilder erlernen muss. Die Aufgabe des Netzwerks ist es, inkrementell neue Klassen zu erlernen, ohne dabei alte, bereits bekannte Klassen zu vergessen.

MNIST kann als Standard-Datensatz im Bereich des maschinellen Lernens angesehen werden. In diesem Datensatz sind insgesamt 70.000 handgeschriebene Zahlen von 0 bis 9 in *GreyScale-*Bildern im Format 32x32 gesammelt, aufgeteilt in 60.000 Trainings- und 10.000 Testbilder. Als Aufgabe müssen die Zahlen korrekt klassifiziert werden vom Netzwerk. Für die Aufgabe des inkrementellen Klassenlernens wird der sogenannte *Split-MNIST* Datensatz genutzt. Dafür werden jeweils 2 benachbarte Zahlen (z.B. 0/1 und 2/3) zu einer Gruppe zusammengefasst. Diese werden dem Netzwerk gezeigt. Nach einer gewissen Anzahl an Wiederholung wird die nächste Gruppe dem Netzwerk gezeigt. Nachdem alle Gruppen dem Netzwerk gezeigt wurden, werden zufällig Testbilder gezeigt und die Klassifikationsgenauigkeit bestimmt. Dieser Testfall des Split-MNIST Datensatz kann als Grundlagenuntersuchung verstanden werden. Der Datensatz stellt kein allzu komplexes Problem dar, kann jedoch in einem ersten Schritt genutzt werden, um das prinzipielle Funktionieren eines Ansatzes zu untersuchen und mit anderen Ansätzen zu vergleichen, da es für diesen Datensatz eine Vielzahl an Untersuchungen bereits gibt und somit die Performanz auch gut verglichen werden kann mit bekannten Algorithmen.

Als zweite Evaluierungsmethodik wird der ImageNet-Datensatz genutzt. Dieser Datensatz stellt ein komplexes Problem mit 1000 unterschiedlichen Klassen dar. Die Bilder sind RGB-Bilder im Format von 224x224. Allein durch das größere und farbige Bild ist die Aufgabe wesentlich komplexer als der MNIST-Datensatz. Weiterhin gibt es eine Vielzahl an unterschiedlichen Klassen, von speziellen Hunderassen zu einem Polizeiauto. Dadurch herrscht eine sehr große Varianz zwischen den einzelnen Bildern und Klassen. Auf Basis dieses Datensatz werden zwei Testfälle definiert. Zunächst ein kleiner Testfall, bei dem zehn unterschiedliche Klassen nacheinander inkrementell erlernt werden. Dabei soll das bei MNIST gezeigte Verhalten mit wenigen Klassen auf einer komplexeren Aufgabe nachgestellt werden. Die folgenden Klassen werden dabei genutzt. In Klammern ist der Index der jeweiligen Klassen zu finden: Königspinguin (145), Malteser (Hunderasse – 153), Schneeleopard (289), Passagierflugzeug (404), Zeppelin (405), Containerschiff (510), Fußball (805), Sportauto (817), Sattelzug (867) und Orange (950). Zudem wird auf Basis des ImageNet-Datensatzes ein Testfall mit vielen Klassen definiert. Dafür werden von den 1000 verfügbaren Klassen 900 genutzt um das Klassifikator-Netzwerk zu trainieren. Die weiteren 100 Klassen werden dann inkrementell erlernt. Diese fehlenden 100 Klassen werden zufällig ausgewählt und können variieren von Testfall zu Testfall, um eine generelle Aussage zu erhalten über die Fähigkeit neue Klassen zu erlernen, wenn das Netzwerk bereits eine große Anzahl an Klassen erlernt hat.

Diese genannten Fälle lassen sich auch für das verteilte Lernen nutzen, was später in Kapitel 4 detailliert beschrieben wird.

Relevante Daten und Parameter währen der Evaluierung sind die Klassifikationsgenauigkeit und die Anzahl an Wiederholungen, wenn neue Klassen einem Netzwerk gezeigt werden. Für eine aussagekräftige Bewertung werden mehrere Wiederholungen der Testfälle durchgeführt, um die Varianz abschätzen zu können.

# Evaluierungskriterien

*Hier werden die Kriterien jeder Evaluierung genannt. Sie sind derart festzulegen, dass die Evaluierung hinsichtlich ihrer erfolgreichen Durchführung bewertbar ist.*

Es werden zwei Kriterien für die Evaluierung genutzt. Die Klassifikationsgenauigkeit, welche auch als *True Positive* (*TP*) -Rate bezeichnet wird. Diese Genauigkeit gibt an, wie viele Prozent der Testbilder korrekt klassifiziert werden und somit der korrekten Klasse zugeordnet werden können. Für die Bestimmung dieser Genauigkeit werden die in den Datensätzen bereits getrennten Test-Bilder genutzt. Es werden dann die geschätzte Klasse des Netzwerks mit den korrekten Klasse des Bildes verglichen und die Anzahl an korrekten Klassifizierungen gezählt. Diese Anzahl in Relation zu allen Testbildern ergibt die Klassifikationsgenauigkeit (TP-Rate).

Zudem wird untersucht, welchen Einfluss die Anzahl an Wiederholungen von Bildern einer neuen Klasse auf die spätere Klassifikationsgenauigkeit hat. Damit wird untersucht, wie viele Wiederholungen notwendig sind, um eine neue Klasse zu erlernen und somit wie schnell das Netzwerk neue Klassen erlernen kann. Dafür werden die beschriebenen Anwendungsfälle durchgeführt und die Anzahl an Wiederholungen der Bilder einer neuen Klasse variiert. Zum Schluss wird dann die Klassifikationsgenauigkeit ermittelt. Damit kann zum Schluss eine Kurve der Klassifikationsgenauigkeit über die Anzahl an Wiederholungen gezeichnet werden und der Einfluss bewertet werden.

# Evaluierungsfälle

*Zu jedem Evaluierungsfall ist anzugeben*

1. *was zu prüfen ist,*
2. *welche Ausgangssituation hierfür erforderlich ist,*
3. *welche Eingaben (Daten und Signale mit allen für die Prüfung ausschlaggebenden Eigenschaften wie Zeitbedingungen) notwendig sind und*
4. *welche Ergebnisse (Ausgabedaten und Reaktionen/Effekte) zu erwarten sind.*

Wie bereits beschrieben, werden konkrete Evaluierungsfälle definiert und zur Bewertung genutzt. In allen Fällen wird die Objektklassifizierung auf Basis von Bildern (Grey-Scale – MNIST und RGB – ImageNet) untersucht. Diese Bilder stellen den Eingang der Testfälle dar. Als möglichen Vorverarbeitungsschritt kann eine Bildnormalisierung eingesetzt werden. Das Kriterium der Klassifikationsgenauigkeit wird dabei für die Auswertung eingesetzt um das Potenzial des Algorithmus zu bewerten. Die einzelnen Fälle werden im Folgenden beschrieben.

****Split-MNIST****

**Hier wird als Ausgangssituation ein untrainiertes Klassifikator-Netzwerk (hier: Modul B) eingesetzt. Der MNIST-Datensatz wird in fünf Gruppen mit jeweils zwei Zahlen eingeteilt (0/1, 2/3, 4/5, 6/7, 8/9). Trainingsbilder jeder Gruppe werden dem Netzwerk für eine bestimmte, variabel einstellbare Anzahl an Wiederholungen gezeigt. Nach einer gewissen Anzahl an Wiederholungen wird angenommen, dass das Netzwerk diese Klassen erlernt hat. Dann wird diese Prozedur für die nächste Gruppe wiederholt. Nachdem alle Klassen erlernt wurden, werden die Testbilder (10.000 gleichverteilte Bilder über alle Klassen) dem Netzwerk gezeigt und die gesamte Klassifikationsgenauigkeit bestimmt.**

**Ein weiterer Testfall auf diesem Datensatz untersucht die Fähigkeiten des verteilten Lernens. Dabei werden zwei identische Netzwerke trainiert, jedoch mit unterschiedlichen Klassen. So kann Netzwerk 1 mit den Gruppen 0/1, 2/3 und 4/5 trainiert werden, während Netzwerk 2 mit den Gruppen 6/7 und 8/9 trainiert wird. Diese Netzwerke tauschen nach dem Training ihre Gewichte aus. Dadurch sollen beide Netzwerke die Fähigkeit erhalten, die anderen Klassen bestimmen zu können. Somit sollen beide Netzwerke in der Lage sein, alle Klassen des Datensatzes bestimmen zu können.**

**Es wird erwartet, dass Ergebnisse ähnlich klassisch trainierten Netzwerken erzielt werden können. Als Referenz dienen in der Literatur vorhandene Resultate auf diesem Datensatz.**

****ImageNet-10****

Das Vorgehen für diesen Testfall ist identische zu dem bereits beschriebenen Vorfall für Split-MNIST. Es sind hier jedoch keine Gruppen vorhanden, sondern zehn unterschiedliche Klassen, die kontinuierlich vom Netzwerk erlernt werden. Ansonsten ist bis auf die eingesetzten Bilder (224x224 RGB Bilder statt 28x28 GrayScale) der Anwendungsfall identisch. Es soll untersucht werden, ob das schnell lernende Klassifikator-Netzwerk in der Lage ist, solche komplexeren Aufgaben korrekt zu lösen. Als Referenz werden auch hier vorhandene Ergebnisse aus der Literatur herangezogen.

ImageNet

Bei diesem Anwendungsfall wird der gesamte ImageNet-Datensatz mit seinen 1.000 Klassen genutzt. Dafür wird das Netzwerk zunächst auf 900 Klassen vortrainiert. Dies stellt den Ausgangspunkt dieses Evaluierungsfalls dar. Von diesem Punkt an werden die weiteren 100 Klassen kontinuierlich dem Netzwerk hinzugefügt, so dass dieses diese neuen Klassen erlernen kann. Dies stellt ein sehr komplexes Problem dar, da sowohl die Bilder und einzelnen Klassen komplex sind, zum anderen auch eine sehr hohe Anzahl an Klassen vorhanden sind, die unterschieden werden müssen. Dieser Fall mit seiner hohen Komplexität stellt die finale Bewertung des Algorithmus dar.