|  |  |  |
| --- | --- | --- |
| **MA 3062** | | |
| Untersuchung und prototypische Umsetzung eines Lifelong Deep Neural Network Algorithmus | | |
| **Simon Kamm** | | |
|  | | |
| **Konzeption** | | |
|  | 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 |  |  | 27.05.19 | in Bearb. | Erstellung |
|  |  |  |  |  |  |

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

* *Wurden alle Anforderungen in der Konzeption umgesetzt?*
* *Ist die Beschreibung der Konzeption detailliert genug?*
* *Entspricht das Ergebnis den Erwartungen des Auftraggebers?*
* *Ist auf Basis der erstellten Konzeption die Realisierung eines Prototyps möglich?*

# Inhaltsverzeichnis

0 Inhaltsverzeichnis 2

1 Beschreibung der Konzeption 3

1.1 Modul A 3

1.1.1 AlexNet 4

1.1.2 VGG 4

1.2 Modul B 4

2 Zusammenfassung 5

3 Literaturverzeichnis 6

# Beschreibung der Konzeption

In diesem Dokument wird die konkrete Konzeption des L DNN Algorithmus beschrieben. Dafür werden unterschiedliche mögliche Architekturen untersucht und verglichen, und schließlich eine konkrete Architektur ausgewählt, welche im weiteren Verlauf der Arbeit implementiert wird.

Der L DNN Algorithmus besteht aus zwei Modulen, dem Modul A und Modul B. Details zum Aufbau und den grundlegenden Ideen des L DNN Algorithmus sind in den „Grundlagen“ genannt. Für eine Übersicht ist in Abbildung 1 die Architektur des L DNN Algorithmus dargestellt.

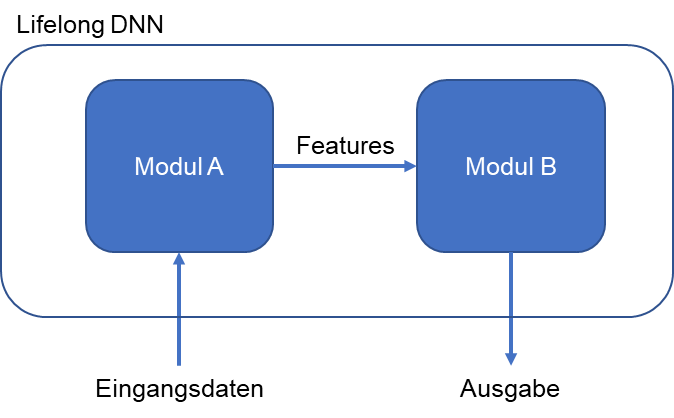


Abbildung : Graphische Darstellung des L DNN A

Im Folgenden werden nun unterschiedliche Architekturen für die einzelnen Module A und B verglichen und bewertet für die in dieser Arbeit gestellten Anforderungen.

## Modul A

Modul A ist das langsame oder nicht lernende Modul innerhalb des L DNN Algorithmus. Es ist ein klassisches DNN, welches zur Extraktion von relevanten Features auf Basis von Eingangsdaten genutzt wird. Im Rahmen dieser Arbeit werden vortrainierte, feste DNN-Architekturen genutzt, da das Trainieren solcher DNN-Architekturen sehr zeit- und rechenaufwändig ist.

Es gibt eine Vielzahl an DNN-Architekturen, die heutzutage zur Feature-Extraktion eingesetzt werden. Jede Architektur erfüllt dabei unterschiedliche Anforderungen, beziehungsweiße nutzt unterschiedliche Ansätze, die für den einen oder anderen Anwendungsfall besser geeignet sind. Für das Modul A innerhalb des L DNN Algorithmus ist zum einen wichtig, dass die extrahierten Features eine bestmögliche Klassifikation erlauben. Zum anderen soll das System auch fähig sein in Echtzeitsystemen auf mobilen Endgeräten (z.B. Smartphone) zu funktionieren [1], [2]. Dafür soll das Modul A Eingangsdaten schnell verarbeiten können und gleichzeitig darf der Speicher- und Rechenbedarf des Netzwerkes nicht zu groß sein, da sowohl Speicher als auch Rechenleistung auf mobilen Endgeräten limitiert vorhanden ist.

Im Folgenden werden bekannte Klassifikator-Netzwerke auf Basis von DNN Architekturen vorgestellt und diese verglichen auf deren Einsetzbarkeit in dieser Arbeit. Als Grundlage für den späteren Vergleich und genannte Metriken wird der Bilddatensatz ImageNet [3] genutzt. Mit ca. 1,2 Millionen Bildern aus 1000 verschiedenen Klassen stellt ImageNet ein sehr komplexes Problem dar. Vortrainierte Netzwerke, welche zur Extraktion von Features genutzt werden sind heutzutage auf diesem Datensatz vortrainiert, da er aufgrund seiner großen Anzahl an Trainingsbildern und Klassen eine gute Generalisierung der extrahierten Features des Netzwerks erzeugt.

### AlexNet

AlexNet war eine der ersten DNNs welche die Klassifikationsgenauigkeit auf dem ImageNet Datensatz signifikant erhöhte. Im Rahmen der ILSVRC (*ImageNet Large Scale Visual Recognition Competition*) erreichte AlexNet einen Top-5 Klassifikationsfehler von 15,3%. Der zweitplatzierte dieses Wettkampfes erreichte lediglich einen Top-5 Klassifikationsfehler von 26%. Dadurch konnte eine erhebliche Steigerung der Klassifikationsperformanz erzielt werden. Dabei ist AlexNet im Vergleich zu aktuellen Architekturen sehr simpel, mit lediglich fünf Convolutional Schichten, verschiedenen Max-Pooling Schichten sowie drei Fully Connected Schichten [4], [5]. Neuerungen im Vergleich zu damaligen DNNs war der Einsatz von ReLU-Aktivierungsfunktionen sowie die Regularisierung mithilfe von Dropout. Beide Methoden sind seitdem in nahezu allen DNN-Architekturen im Einsatz. Das Netzwerk besteht aus 60 Millionen Parametern und 650.000 Neuronen. In Abbildung 2 ist die Modellarchitektur des AlexNet graphisch dargestellt, bei welcher die fünf Convolutional und die drei Fully Connected Schichten zu sehen sind.

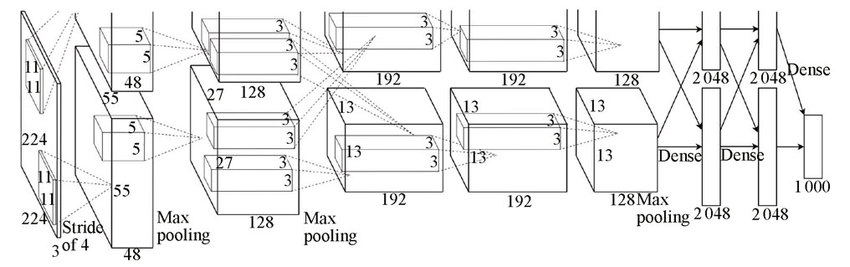


Abbildung : Modelarchitektur des AlexNet [5]

Die *Default*-Größe eines Parameters (Floating Point oder Integer) in TensorFlow ist mit 32-bit angegeben [6]. Unter der Annahme dieser Parametergröße benötigt AlexNet mit 60 Millionen Parametern 240 MB Speicher zur Sicherung des Modells.

### VGG

Die sogenannten VGG-Architekturen entstammen der *Visual Geometry Group* (VGG) der Universität Oxford. Dabei gibt es einige unterschiedliche bekannte Architekturen, die am häufigsten genutzten dabei sind das VGG-16 sowie das VGG-19 Netzwerk. Die Zahlen stehen dabei für die Anzahl an Gewichtsschichten innerhalb des DNN. Von der grundsätzlichen Architektur ähnelt es deutlich dem bereits beschriebenen AlexNet. Der große Unterschied ist der Einsatz kleinerer Kernel-Filter in den *Convolutional* Operationen (3x3 anstatt z.B. 11x11 oder 5x5 in AlexNet). Zum einen werden mehrere dieser *Convolutional* Schichten hintereinander gesetzt, wodurch finale Funktion diskriminativer werden soll. Ein weiterer Vorteil ist die geringere Anzahl an Parametern. Unter der Annahme, dass der Eingang und Ausgang eines *Convolutional Stacks* mit drei 3x3 *Convolutional* Schichten Kanäle besitzt, werden Parameter benötigt. Wird lediglich eine einzelne *Convolutional* Schicht mit 7x7 Kernel-Filtern genutzt, werden Parameter benötigt. Dadurch können tiefere Netzwerke aufgebaut werden mit mehr Schichten ohne einen deutlich erhöhten Speicherbedarf zu haben. Durch den Anstieg der Tiefe des Netzwerks ist das Netzwerk fähig komplexere Features zu erlernen [7]. Unterschiedliche Architekturen der VGG-Netzwerke für den ImageNet Bilddatensatz mit 1000 Klassen und RGB Bildern der Größe 224x224 sind in folgender Tabelle 1 dargestellt.

Tabelle :VGG-Netzwerk Architekturen [7]

|  |  |  |  |
| --- | --- | --- | --- |
| A  11 Gewichts-Schichten | B  13 Gewichts-Schichten | C  16 Gewichts-Schichten | D  19 Gewichts-Schichten |
| Eingangssignal (z.B. 224x224 RGB Bild) | | | |
| Conv3-64 | Conv3-64  Conv3-64 | Conv3-64  Conv3-64 | Conv3-64  Conv3-64 |
| Max-Pooling | | | |
| Conv3-128 | Conv3-128  Conv3-128 | Conv3-128  Conv3-128 | Conv3-128  Conv3-128 |
| Max-Pooling | | | |
| Conv3-256  Conv3-256 | Conv3-256  Conv3-256 | Conv3-256  Conv3-256  Conv3-256 | Conv3-256  Conv3-256  Conv3-256  Conv3-256 |
| Max-Pooling | | | |
| Conv3-512  Conv3-512 | **Conv3-512**  **Conv3-512** | **Conv3-512**  **Conv3-512**  **Conv3-512** | **Conv3-512**  **Conv3-512**  **Conv3-512**  **Conv3-512** |
| Max-Pooling | | | |
| Conv3-512  Conv3-512 | Conv3-512  Conv3-512 | Conv3-512  Conv3-512  Conv3-512 | Conv3-512  Conv3-512  Conv3-512  Conv3-512 |
| Max-Pooling | | | |
| FC-4096 | | | |
| FC-4096 | | | |
| FC-1000 | | | |
| Softmax | | | |

Wie bereits geschrieben werden die Netzwerke anhand der Anzahl der Gewichtsschichten (*Convolutional* oder *Fully Connected* (FC) Schicht) benannt. So ist Netzwerk *C* als VVG-16 bekannt und Netzwerk D als VGG-19. VGG-16 und VGG-19 als häufig eingesetzte Feature-Extrahierer werden im Folgenden bezüglich ihrem Speicherbedarf und der Performanz untersucht. Dabei sind in Tabelle 2 die unterschiedlichen Kriterien nach [7] für die einzelnen Netzwerke gelistet.

Tabelle : Vergleich von VGG-16 und VGG-19

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Netzwerk-Architektur | Anzahl Parameter | Speicherbedarf | Top-1 Klassifikations-Fehler | Top-5 Klassifikations-Fehler |
| VGG-16 | 138 Millionen | 552 Megabyte | 25.6 % | 8.1 % |
| VGG-19 | 144 Millionen | 576 Megabyte | 25.5 % | 8.0 % |

Die Anzahl der Parameter steigt sinnvollerweise mit der Anzahl an Schichten, und damit auch der Speicherbedarf. Auf ImageNet ist zu beobachten, dass durch die 3 zusätzlichen Schichten im VGG-19 Netzwerk ein nahezu vernachlässigbarer Genauigkeitsgewinn erreicht werden kann. Es werden 6 Millionen zusätzliche Parameter benötigt, um den Top-1 und Top-3 Klassifikationsfehler um 0.1% zu verringern. Das erklärt, weshalb VGG-16 in den meisten Anwendungen bevorzugt eingesetzt wird.

### Inception

### Resnet

### MobileNet

## Modul B

# Zusammenfassung

*Hier soll das Ergebnis der Konzeption dargestellt werden.*

# Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | M. Luciw, S. Olivera, A. Gorshechnikov, J. Wurbs, H. M. Ames und M. Versace, „Systems and Methods to enable Continual, Memory-Bounded learning in Artificial Intelligence and Deep Learning Continuously operating Applications across networked Compute Edges“. United States of America Patent US 2018/0330238 A1, 15 November 2018. |
| [2] | Neurala Inc., „Lifelong Deep Neural Networks - Tech Summary,“ [Online]. Available: https://info.neurala.com/hubfs/docs/ Neurala\_LifelongDNNWhitepaper.pdf. [Zugriff am 7 Mai 2019]. |