technische universität
dortmund

Department of Computer Science

# Masterarbeit

## Zeit-Effizientes Training von Convolutional Neural Networks

Jessica Bühler

11. November 2019

**Supervisors:**
Prof. Dr. Heinrich Müller
M.Sc. Matthias Fey

Lehrstuhl VII
Informatik
TU Dortmund

# Inhaltsverzeichnis

# Mathematical Notation

| Notation | Meaning |
|---|---|
| $\mathbb{N}$ | Set of natural numbers $1, 2, 3, \ldots$ |
| $\mathbb{R}$ | Set of real numbers |
| $\mathbb{R}^d$ | $d$-dimensional space |
| $\mathcal{M} = \{m_1, \ldots, m_N\}$ | Set $\mathcal{M}$ of $N$ elements $m_i$ |
| $\mathbf{p}$ | Vector |
| $\mathbf{p}_i$ | Element $i$ of the vector |
| $\mathbf{v}_i^{(j)}$ | Element $i$ of the vector $j$ |
| $\mathbf{A}$ | Matrix |

# 1. Einleitung

Thema dieser Arbeit ist die Frage wie man für ein gegebenes Bildklassifikationsproblem Zeit beim Trainieren des neuronalen Netzes sparen kann. Es geht dabei aber nicht nur um das direkte Einsparen während eines Trainingsdurchlaufs sondern auch darum wie man effizient ein gegebenes Netz verbessert.

# Teil I.

# Literatur zum Zeit-Effizienten Training von CNNs

# 2. Übersicht über die Methoden zur Beschleunigung des Trainings

Dies ist die Einleitung für Part A – Literatur

# 3. Beschleunigung der Convolution

Die Zeit, die ein Convolutional Layer braucht um berechnet zu werden hängt ab von:

Fehlt hier noch etwas

- der Filtergrösse
- der Bildgrösse
- dem verwendeten Zahlenformat

Beim Verändern der Filter- oder der Bildgrösse, um Trainingszeit zu sparen, verändert sich auch die Erkennungsleistung . Dies ist beim Verändern des verwendeten Zahlenformats nicht umbedingt gegeben. Standardformat ist eine 32 Bit Gleitkommazahl. Die einfachste Methode hier Trainingszeit zu sparen ist das Halbieren der Bitanzahl auf 16 Bit. Eine weitere Methode ist das Benutzen von Dynamischen Festkommazahlen.
Quelle: [DMM+18]

cite

# 4. Beschleunigung der Berechnung des Gradienten

## 4.1. Verdünnung des Aktivierungsgradienten zur Beschleunigung des Gradienten

## 4.2. Beschleunigung des Trainings durch Graditenapproximation

# 5. Verfahren um weniger Trainingsdaten zu verwenden

## 5.1. Stochastisches Pooling

## 5.2. Lernen von Struktur und Stärke von CNNs

# 6. Strukturelle Veränderungen zur Beschleunigung des Trainings

## 6.1. Prune Train

## 6.2. Net 2 Net

## 6.3. Kernel rescaling

## 6.4. Resource Aware Layer Replacement

# 7. Weitere Herangehensweisen

## 7.1. Tree CNN

## 7.2. Standardization Loss

## 7.3. Wavelet

# Teil II.

# Praktischer Teil– Arbeitstitel

# 8. Einleitung

Dies ist die Einleitung für Part B

# 9. Durchführung

- Baseline Training ab der ersten Epoche.
- Prune währenddem Training
- Training bis zu dem Zeitpunkt wo durch ein weitere Epoche nichts besser wird
- Überprüfe wieviel in den letzten Epochen gepruned wurde um zu entscheiden ob das Netz weiter odertiefersein soll

# Teil III.

# Additional information

# Abbildungsverzeichnis

# List of Algorithms

# List of Listings

# Literaturverzeichnis

[AS18] Menachem Adelman and Mark Silberstein. Faster neural network training with approximate tensor operations. *CoRR*, abs/1805.08079, 2018.

[DMM+18] Dipankar Das, Naveen Mellempudi, Dheevatsa Mudigere, Dhiraj D. Kalamkar, Sasikanth Avancha, Kunal Banerjee, Srinivas Sridharan, Karthik Vaidyanathan, Bharat Kaul, Evangelos Georganas, Alexander Heinecke, Pradeep Dubey, Jesús Corbal, Nikita Shustrov, Roman Dubtsov, Evarist Fomenko, and Vadim O. Pirogov. Mixed precision training of convolutional neural networks using integer operations. *CoRR*, abs/1802.00930, 2018.