

Masterarbeit – Zeit-Effizientes Training von Convolutional Neural Networks

Jessica Bühler

1. März 2021



Übersicht I

- 1** Einleitung
- 2 MorphNet
- 3 PruneTrain
- 4 Net2Net
- 5 weiteres Vorgehen
- 6 Anhang

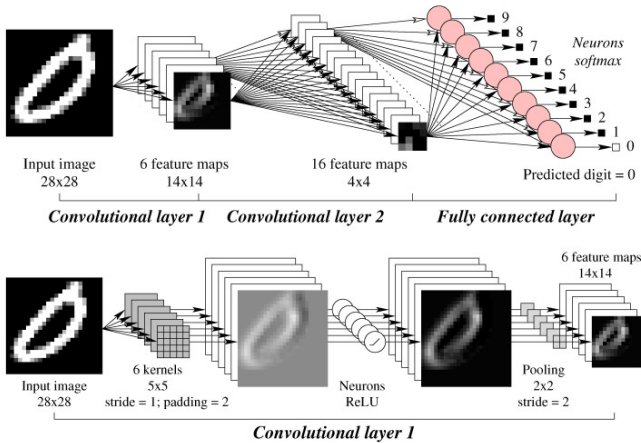
Ausgangssituation

Gegeben: Unbekannter Datensatz aus Bildern, mit gelabelten Trainingsdaten und ungelabelten Testdaten

Aufgabe:



Verwendete Architektur I



verwendete Architektur II

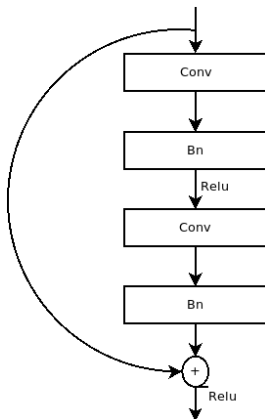


Abbildung: Basisblock

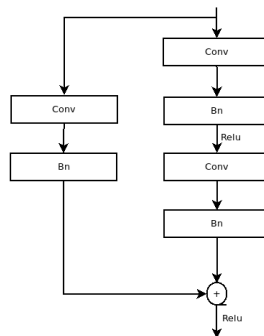


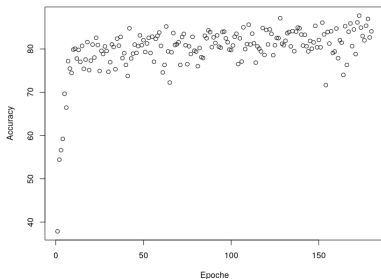
Abbildung: Übergangsblock

Abbildung: Grafische Darstellung Basis- und Übergangsblock

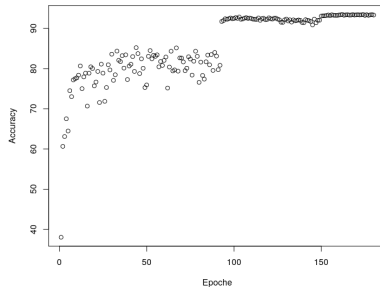
Bisherige Arbeiten zum Thema

- Automatische Architektursuche mit großem Suchraum
 - DVolver
- Architektursuche währenddem Trainieren
 - MorphNet

Baseline-Netz



(a)



(b)

Abbildung: Vergleich zwischen (a) Baseline-Netz ohne Anpassung der Lernrate und (b) Baseline-Netz mit Anpassung der Lernrate in Epoche 93 und 150.

Übersicht I

- 1 Einleitung
- 2 MorphNet
- 3 PruneTrain
- 4 Net2Net
- 5 weiteres Vorgehen
- 6 Anhang

Übersicht über MorphNet

- Morphnet vergrößert und verkleinert konkurrierend das Netz um die besten Hyperparameter zu finden

- 1: Trainiere das Netz um $\mathcal{W}^* = \underset{\mathcal{W}}{\operatorname{argmin}} \text{ Loss-Funktion des CNN} + \lambda \cdot$
Nebenbedingung, die die Breite des Netzes minimiert
 - 2: Finde die neue Breite $\mathcal{C}'_{1:J}$, die durch 1. errechnet wurde
 - 3: Finde das größte ω , so dass $\mathcal{F}(\omega \cdot \mathcal{C}_{1:J}) \leq \zeta$ gilt
 - 4: Wiederhole ab 1. so häufig wie gewünscht mit $\mathcal{C}_{1:J} = \mathcal{C}'_{1:J}$
- Ausgabe:* $\omega \cdot \mathcal{C}_{1:J}$

```
BasicBlock(  
    (0): Conv2d(76, 5, kernel_size=(3, 3), stride=(1, 1),  
padding=(1, 1))  
    (1): BatchNorm2d(5, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True)  
    (2): ReLU(inplace=True)  
    (3): Conv2d(5, 76, kernel_size=(3, 3), stride=(1, 1),  
padding=(1, 1))  
    (4): BatchNorm2d(76, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True))
```

Abbildung: Struktur eines Basisblocks als Ergebnis von MorphNet 2

```
BasicBlock(
```

```
    (0): Conv2d(76, 76, kernel_size=(3, 3), stride=(2, 2),  
padding=(1, 1))
```

```
    (1): BatchNorm2d(76, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True)
```

```
    (2): ReLU(inplace=True)
```

```
    (3): Conv2d(76, 76, kernel_size=(3, 3), stride=(1, 1),  
padding=(1, 1))
```

```
    (4): BatchNorm2d(76, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True))
```

Abbildung: Struktur eines Basisblocks als Ergebnis von MorphNet 1

Ergebnis der MorphNet Evaluierung

- Bei der gegebenen Struktur
- und den vorgegebenen Ressourcen

ist keine signifikante Verbesserung möglich.

Nachteile von MorphNet:

- es ist nicht möglich, die Tiefe des Netzes zu verändern

Übersicht I

1 Einleitung

2 MorphNet

3 PruneTrain

4 Net2Net

5 weiteres Vorgehen

6 Anhang

Übersicht über PruneTrain I

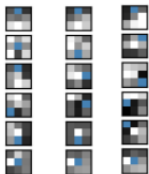
- Soll den Verkleinerungsoperator von MorphNet ersetzen



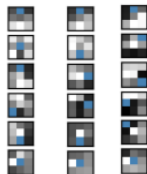
$$GL(\mathcal{W}) = \sum_{j=1}^J \left(\sum_{c_j=1}^{C_j} \|W_j(c_j, :, :, :)\|_2 + \sum_{k_j=1}^{K_j} \|W_j(:, k_j, :, :)\|_2 \right)$$

Übersicht über PruneTrain II

Epoche 29



Epoche 30



Epoche 31

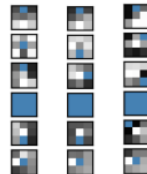
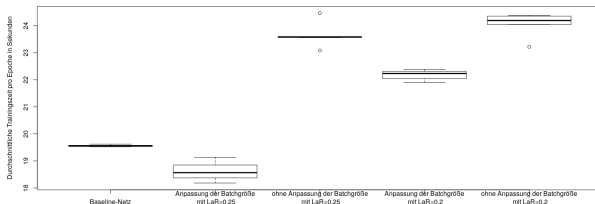
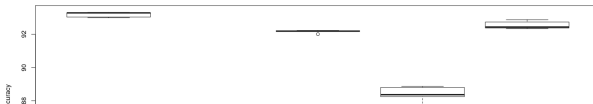


Abbildung: Grafische Darstellung des Prunings des ResNets

Evaluierung von PruneTrain bei Veränderung der Batchgröße



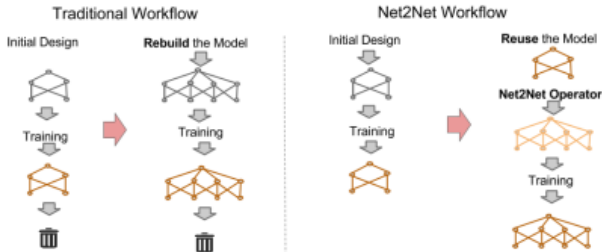
(a)



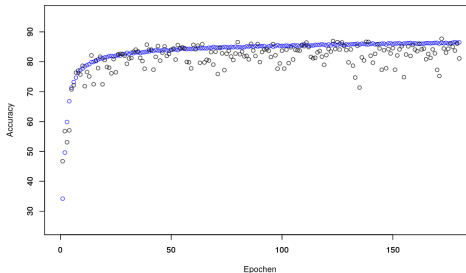
Übersicht I

- 1 Einleitung
- 2 MorphNet
- 3 PruneTrain
- 4 Net2Net**
- 5 weiteres Vorgehen
- 6 Anhang

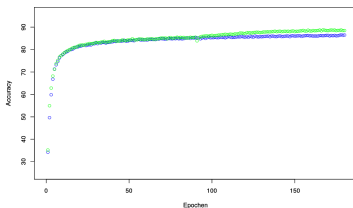
Übersicht über Net2Net



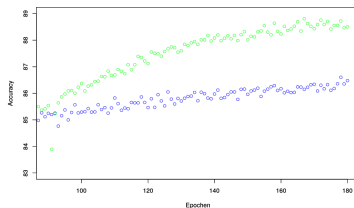
Evaluierung von Net2Net: Operator für ein breiteres Netz



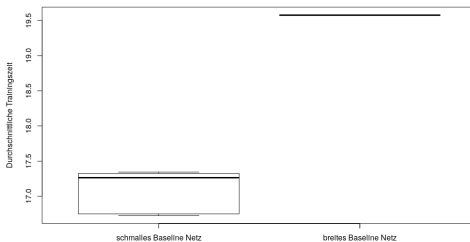
(a)



(b)



(c)



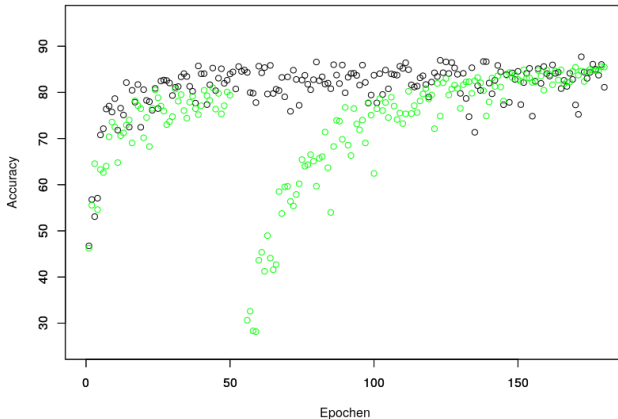
(d)

Abbildung

Ergebnis der Evaluierung von Net2Net: Operator für ein tieferes Netz I

todo: Grafik für das Einfügen eines neuen Blocks

Ergebnis der Evaluierung von Net2Net: Operator für ein tieferes Netz II



Quelle des verwendeten Codes

- MorphNet
- PruneTrain
 - ohne Anpassung der Batchgröße:
 - mit zusätzlicher Anpassung der Batchgröße: eigene Implementation
- Net2Net: eigene Implementetation

Ausblick

Übersicht I

- 1 Einleitung
- 2 MorphNet
- 3 PruneTrain
- 4 Net2Net
- 5 weiteres Vorgehen
- 6 Anhang**

Architektur