

Regularisierung in Neuronalen Netzwerken

Son Tien Nguyen

LMU München

21.01.20

Inhalt

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

1 Einführung

2 Regularisierung

3 CNN

4 Schluss

Neuronale Netzwerke

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- Anordnung der Neuronen in Layer/Schichten
- Verbindung der Neuronen zwischen den Schichten
- Von Eingabeschicht fließen Informationen über Zwischenschicht(en) zur Ausgabeschicht

Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

Perceptron (P)



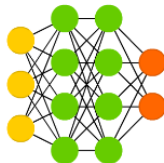
Feed Forward (FF)



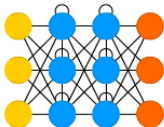
Radial Basis Network (RBF)



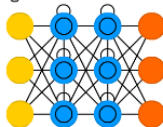
Deep Feed Forward (DFF)



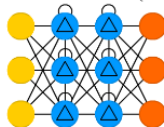
Recurrent Neural Network (RNN)



Long / Short Term Memory (LSTM)



Gated Recurrent Unit (GRU)



Quelle: <https://jaai.de/wp-content/uploads/2017/09/neuralnetworks.png>

- Trainingsset, Validationsset, Testset
- Testfehler soll kleingehalten werden
- Underfitting: Modell zu simpel; generell keine guten Ergebnisse
- Overfitting: NN Modell passt sich Trainingsdaten zu sehr an

Underfitting Overfitting

Regularisierung
in Neuronalen
Netzwerken

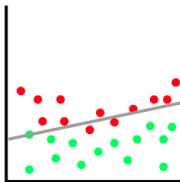
Son Tien
Nguyen

Einführung

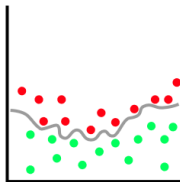
Regularisierung

CNN

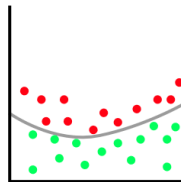
Schluss



Underfitting



Overfitting



Balanced

Quelle: <https://blog.floydhub.com/>

Regularisierung

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- Jede Änderung, die wir am Lernalgorithmus vornehmen, damit Test Error sich verbessert, aber nicht der Trainingerror
- Regularisierung benutzen, um Overfitting zu vermeiden!

L₂ Regularisierung

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- Idee: Gewichte, die zu groß werden, bestrafen
- umso kleiner das Gewicht, umso kleiner das Penalty
- Wert kann sich nur 0 annähern
- λ steht für die Strenge des Penaltys
- durch Testen richtigen Wert von λ finden

L₂ Regularisierung Loss Funktion

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

$$L_2 = (wx+b-y)^2 + \lambda \times \|W\|^2$$

L₂ Regularisierung: Formel neues Gewicht mit Gradient Descent

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

$$w_{neu} = (w_{alt} - \eta \lambda w_{alt})$$

L_1 Regularisierung

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- kleine Werte werden hart bestraft
- Resultat: viele Nullen
- Besser geeignet für Feature Selection

L₁ Regularisierung Loss Funktion

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

$$L_1 = (wx+b-y)^2 + \lambda \times \| W \|$$

L₁ Regularisierung: Formel neues Gewicht mit Gradient Descent

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

$$w_{neu} = (w_{alt} - H) - \lambda$$

Vergleich L_1 , L_2

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- $H = 0$
- $w, \text{lambda} = 0,2$
- $L_1: w_{neu} = (0,2 - 0) - 0,2 = 0$
- $L_2: w_{neu} = (0,2 - 0) - 2 \times 0,2 \times 0,2 = 0,12$

Dropout

Regularisierung
in Neuronale
Netzwerken

Son Tien
Nguyen

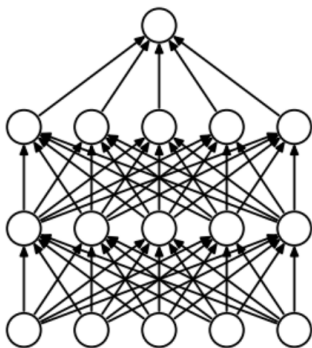
Einführung

Regularisierung

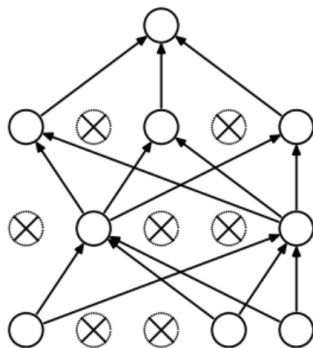
CNN

Schluss

- Eliminierung zufälliger Knoten
- Vorgang wird wiederholt
- nur Knoten aus Hidden Layer werden eliminiert
- Knoten und Layer sollen unabhängig voneinander werden



(a) Standard Neural Net



(b) After applying dropout.

Quelle: <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning>

Early Stopping

Regularisierung
in Neuronale
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- gehen zu Zeitpunkt zurück, mit niedrigsten Validation Error
- speichern jedes mal Parameter, wenn Error sich verbessert
- Training endet nach vorgegebener Zahl von Iterationen
- speichern jedes mal Parameter, wenn Error sich verbessert

Umgang Validation Set bei Early Stopping

Regularisierung
in Neuronale
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- brauchen Validation Set
- Teil der Daten wird nicht für das Training benutzt
- Wie diesen Teil sinnvoll nutzen?

Variante 1

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- Modell von vorne initialisieren
- gleiche Zahl der Trainingsschritte wie bei Initialdurchgang
- keine verlässliche Methode, optimale Trainingsschrittzahl zu finden

Variante 2

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- Parameter aus Initialdurchgang behalten und weitertrainieren
- keine vorgegebene Anzahl von Schritten
- können Loss Funktion beobachten
- Stoppen, bis Wert schlechter ist als bei Initialdurchgang
- nicht well-behaved, stoppt möglicherweise nie

Convolutional Neural Network

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

- Parameter Sharing: selbe Matrix der Parameter wird über bestimmte Einheiten genutzt
- dadurch werden gleichen Objekte an unterschiedlichen Stellen erkannt
- effektiv für Image Recognition und Klassifizierung
- effektiv in NLP

Aufbau CNN

Regularisierung
in Neuronalen
Netzwerken

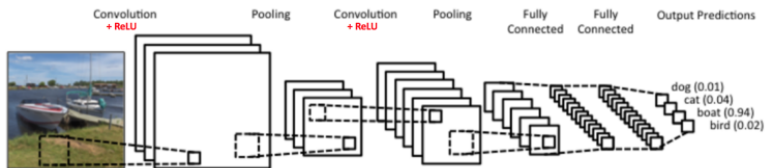
Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss



Quelle: <https://ujjwalkarn.me/>

Convolution

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

1	0	-1
1	0	-1
1	0	-1

=

6		

$$\begin{aligned} &7 \times 1 + 4 \times 1 + 3 \times 1 + \\ &2 \times 0 + 5 \times 0 + 3 \times 0 + \\ &3 \times -1 + 3 \times -1 + 2 \times -1 \\ &= 6 \end{aligned}$$

Quelle: <https://www.learnopencv.com/>

Convolution

Regularisierung
in Neuronalen
Netzwerken

Son Tien
Nguyen

Einführung

Regularisierung

CNN

Schluss

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

1	0	-1
1	0	-1
1	0	-1

=

6	-9	-8
-3	-2	-3
-3	0	-2

Quelle: <https://www.learnopencv.com/>

Pooling

Regularisierung
in Neuronalen
Netzwerken

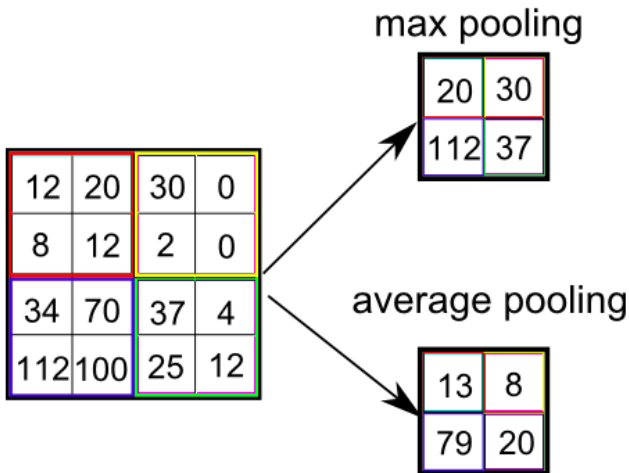
Son Tien
Nguyen

Einführung

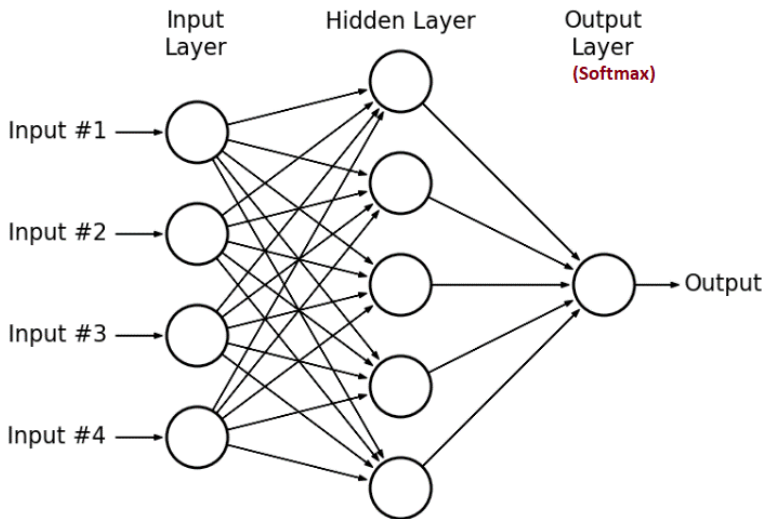
Regularisierung

CNN

Schluss



Fully Connected Layer



Dataset Augmentation

Regularisierung
in Neuronale
Netzwerken

Son Tien
Nguyen

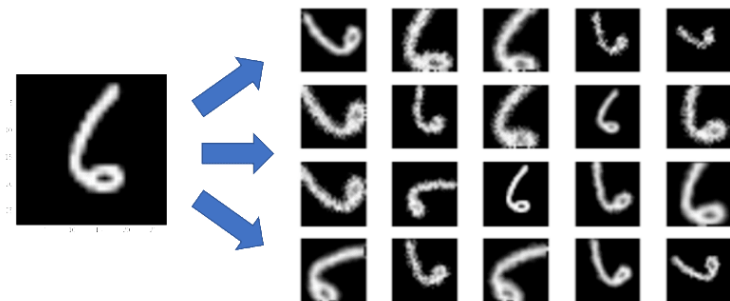
Einführung

Regularisierung

CNN

Schluss

- Training an großer Datenmenge hilft zum Generalisieren
- Lösung: Falsche Daten kreieren
- sehr hilfreich bei Objektidentifikation in Bildern
- durch leichte Veränderungen am Bild kann man Datenset größer machen
- !kann zu Transformation zu anderer Kategorie führen!



Quelle: <https://hazyresearch.github.io/snorkel/blog/tanda.html>

- Bishop, Christopher M. *Pattern recognition and machine learning*, Kapitel 5. springer, 2006.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*, Kapitel 7. MIT press, 2016.
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.
- Subir Varma and Sanjiv Das. *Deep Learning*
- <https://towardsdatascience.com/intuitions-on-l1-and-l2-regularisation-235f2db4c261>

Vielen Dank für die Aufmerksamkeit!