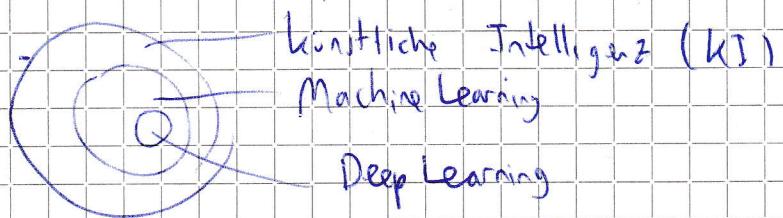


# Deep Learning (KLR - 341)

Monday, 04.12.2023

Bernd Ebenhoch

TensorFlow



supervised vs unsupervised Learning

Überwachtes vs Unüberwachtes Lernen

Regression,  
Klassifikation

{  
Clustering

Ausreißerdetektion  
(Outlier)

One Hot Encoding

Ordinal- Encoding

Skalierung

Anaconda

# TensorFlow & Pytorch

## MATLAB & Octave

MATLAB vs TensorFlow in Google Trend → MATLAB more

künstliche neuronale Netze / Artificial Neural Networks

- Aktivierungsfunction:
  - Logistische Funktion (Sigmoid)
  - tanh
  - ReLU = rectified linear unit

MLP = Multi Layer Perception / Mehrschichtiges Perzeptron

MLP Classifier in sklearn

~~Loss~~ Loss function → minimization

↳ Least Squares / Methode der Kleinsten Quadrate

Gradientenabstieg / Gradient Descent.

$$\text{loss} = \frac{1}{n} \sum_{i=1}^N [y_i - (w x_i + b)]^2 \rightarrow f(g(w))$$

$$\left. \begin{array}{l} g(w) = y - w x - b \\ f(g) = g^2 \end{array} \right\} \left. \begin{array}{l} \frac{\partial f}{\partial w} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial w} \end{array} \right\} \text{Rück}$$

$$\cancel{\frac{\partial f}{\partial g}} \frac{\partial f}{\partial g} = 2 \cdot g \quad \frac{\partial g}{\partial w} = -x \quad \frac{\partial g}{\partial b} = -1$$

Aufgabe: - numerische Netzwerk mit Bias

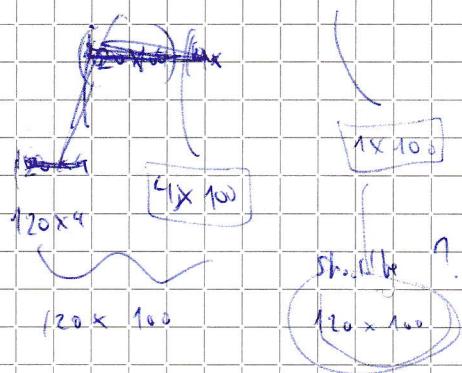
→ ↳ then with TensorFlow  
(und Gradient Tapes)

Mittwoch, 06.12.2023

$$\frac{\partial \text{loss}}{\partial w_1} = \frac{\partial \text{loss}}{\partial y_2} \cdot \frac{\partial y_2}{\partial f} \cdot \frac{\partial f}{\partial w_1} \neq w_1 x + b$$

$\downarrow$        $\downarrow$        $\downarrow$   
 $t_2(y_{\text{pred}} - y_2)$        $\left\{ \begin{array}{l} 1 \\ 0 \end{array} \right.$       ~~w1~~

$$y_1 = X \cdot w_1 + b_1$$



Pytorch

Theano

Keras

Tensorflow raw

Glorot Uniform  $\rightarrow$  book page 393

} initialization of parameters

$\hookrightarrow$  Xavier Glorot & Yoshua Bengio

{ Am Freitag, zu Hause...  
08.12.2014

V verzweigung = Branching

Risotto

Hyperparameter tuning

Verminderung von Overfitting.

Underfitting & Overfitting

Kalibrierung der Wahrscheinlichkeit  $\sim$  Calibration curve  
(shiegen)

Unbalancierte ~~Labels~~ Labels ; Sampling Methoden

- | L UnderSampling
- | OverSampling
- | Gewichtung der Klassen

Gesetz der großen Zahlen

Autoencoder  $\rightarrow$  book page 695  $\leftarrow$  Regularization

Overfitting verhindern:

- |  $\hookrightarrow$  L1 & L2 Regularisierung, Book page 428-429, 187-193
- |  $\hookrightarrow$  Drop-out, Book page 429-432
- |  $\hookrightarrow$  Early-Stopping, Book page 373, 193-194
- |  $\hookrightarrow$  Optimierung der Netz-Größe, Book page 378-385

Keras-tuner

Wie kann U1/U2 Regularisierung Overfitting verhindern?

↳ Die Größen der Parameter im Modell werden reduziert.

↳ Das Modell kann sich verstärkt auf wichtige Merkmale konzentrieren.

Warum neigt ein Machine-Learning System zum Overfitting?

↳ Die Variabilität der Daten ist sehr groß.

Was ist ein Indiz für Underfitting?

↳ Die Metriken für die Trainingsdaten liegen unter der Anforderung.

Was kann bei Dropout beobachtet werden?

↳ Die Metriken der Trainingsdaten sind in den Lernkurven verfügblich.

~~Was gilt~~.

Was gilt es bei Verwendung von Early-Stopping mit der Option ~~restored~~:

↳ ~~restored~~ testsets\_weights = True zu beachten?

↳ Es kann zu einer Überanpassung an die Testdaten kommen.

↳ Die Patience muss genau an das gegebene System angepasst werden.

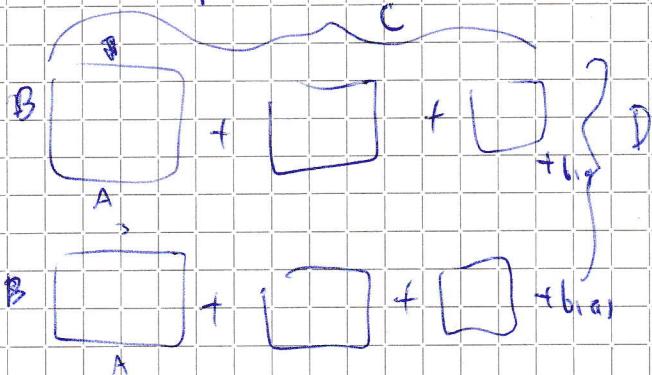
Zu kurze Patience können die Testmetrik verschlechtern.

Optimierung:

+ Lernrate

## Convolutional Neural Network

Kernel - shape  $(A, B, C, D)$



— Konvertierung mehrerer shape

Kernel has a convolutional filter & a bias.

Pooling → Max Pooling

(popular used)

Conv 2D

Max Pooling 2D

Flatten

Global Max Pooling 2D

Dienstag, 12.12.2023

## Convolutional Neural Network

(Seite 539 - 553)



## GoogleNet

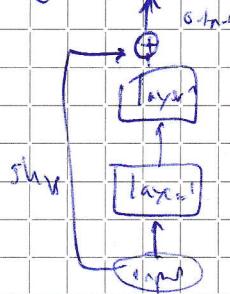


Block :  $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ , pool,  $3 \times 3$ , reduce,  $5 \times 5$ , reduce

$1$	$\downarrow$	$1$	$\downarrow$	$1$
$1 \times 1$		$1 \times 1$		$1 \times 1$
after pooling		before $3 \times 3$		before $5 \times 5$

## AlexNet

ReLU  $\rightsquigarrow$  Vanishing gradient handling with Shortcut / Skip-Verbindung



Mittwoch, 13.12.2013

Complex Convolutional Neural Network

LeNet-5

AlexNet

GoogLeNet

↳ 1x1 Convolution

VGG

↳ VGG-16 & VGG-19

ResNet

↳ Batch Normalization

Xception

Huggingface

Image-Processing & Transfer Learning

Donnerstag, 14.12.2023

## Transfer Learning

- ↳ Tag OS CNN Example Class
- ↳ Tag OS Image Dataset from directory

Objekt Erkennung → ~~Buch~~

## Coding Projekte

- ↳ Objekt lokalisierung (object localization) ↳ Buch Seite 561
- ↳ Semantische Segmentierung (semantic segmentation) ↳ Buch Seite 567
- ↳ PunktWolken (point clouds) ↳ Buch Seite 572  
(3D)

make sense ai → object labeling

YOLO v3 Architecture ~ Object Detec.

## U-Net

Transpose Layer & Upsampling 2D

↳ geht er herum.

Semantische Segmentation

↳ Dimensionen (Anzahl der Bilder, Höhe, Breite, Anzahl der Klassen)

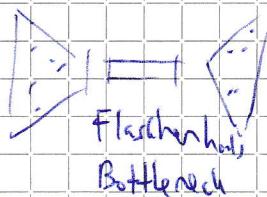
Explainable AI

- ↳ GradCam
- ↳ DeepDream - Verfahren
- ↳ Permutation Feature Importance
- ↳ SHAP (SHapley Additive explanations)
- ↳ Atentionschichten / Attention Layer

GAN (Generative Adversarial Network)

- Neural Style Transfer

- DeepFake



- Autoencoder with 2 Decoder

- Super-resolution ~ U-Net

- Nachcolorieren ~ U-Net

- Bildverwetzung → picture "extrapolation"

- GAN: unsupervised learning with Generator & Discriminator

- Bild-Generation auf Basis eines Textinputs → Dall-E, Stable-Diffusion

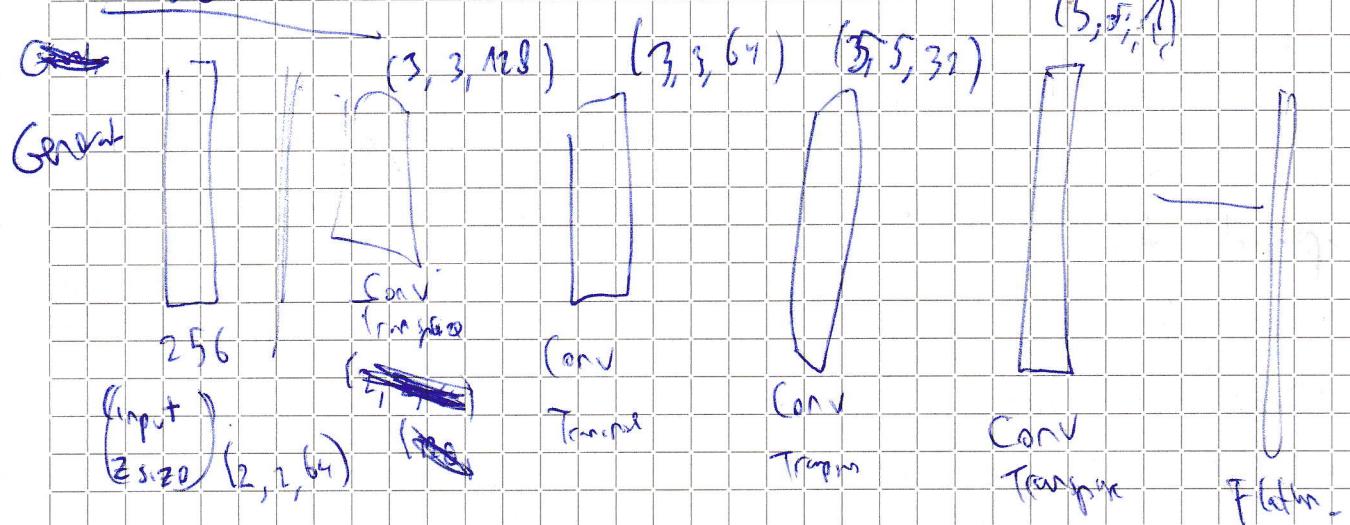
- DeepDream - Verfahren

Unsupervised GANs / Unüberwachte GANs

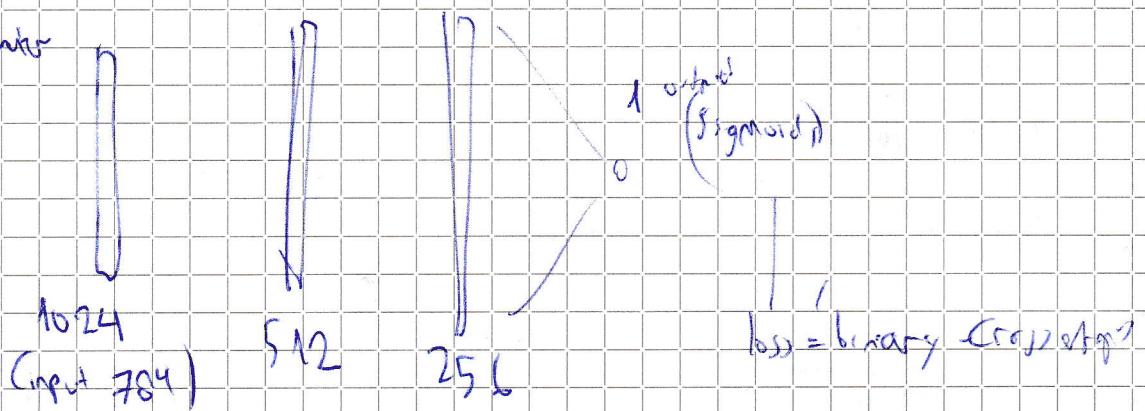
GAN für mnist-Ziffern

## MNIST GAN

~~Gen~~



Discriminator



discriminat. trainable = False  $\leftrightarrow$  to freeze discriminator

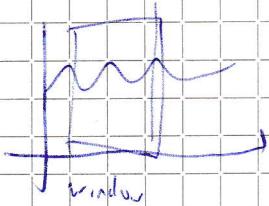
erium podcast

Meetup.com  $\rightarrow$  Data Science Meetup  
 $\rightarrow$  Machine Learning Meetup

GAN ~ 2014

Recurrent Neural Network

Zeitreihenanalyse - Time Series Analysis

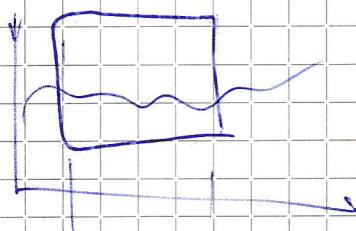
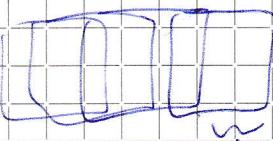
 $x(30, 20, 1)$ 

$\downarrow$   
30 points

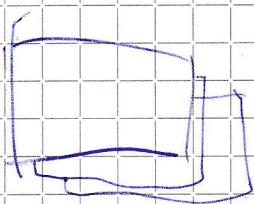
 $y = (30, )$ 

$\rightarrow$  sequence length.

(Datenpunkte, Sequenzlänge, Markomale)

 $(50, 1)$  $\downarrow$  $(29, 20, 1)$  $\rightarrow$ 

for 20 sequences means only need 29 Data points, if the window 50,



$\uparrow + \cancel{\text{shift}} \quad 1 \text{ step}$

Dienstag, 19.12.2023

## Recurrent Neural Network

Zeitreihen in einzelne Sequenzen aufteilen

→ moving window

## ~~RNN~~ LSTM & GRU

## Bayesian Neural Network

Tensorflow-probability

← be careful with version  
during pip install  
+  
0.22 ↑

## TensorFlow Probability Distribution

### Text Analyse

#### - Tokenizer:

"Aku makna nasi" → [ "Aku", "makna", "nasi" ]

#### - Text in Zahlen konvertion

↳ Dictionary

↳ count + Vectorizer (Bag of Words)

↳ Buchstaben basiert Tokenizer → character-to-ASCII

↳ One hot encoding

Word-vektoren / Word-vector

Donnerstag, 21.12.2018

Recurrent Neural Network  $\sim$  Natural Language Process,  
 $\sim$  Text Analysis

Tokenizer

$\hookrightarrow$  from sentence(s) to word / syllables / character

~~Wieso soll Tokenize only training data?~~  
why

$\hookrightarrow$  model fitzo

Was macht die Funktion pad-sequence?  
 $\hookrightarrow$  token  
- scan length per sequence

Was ist die optimale Länge der Sätze (Fragenz)?

$\hookrightarrow$  20? 500?

Was sind Wort-Vektoren?

Was macht eine Embedding Schicht?  $\leftarrow$  Embedding Layer in keras

$\hookrightarrow$  Token  $\xrightarrow{\text{into}}$  Word-Vector

Warum werden für die Textanalyse reziproke Schichten verwendet?

$\hookrightarrow$  Wörter hintereinander.

$\hookrightarrow$  Wörter hat andere Bedeutung

BLEU - Metrik / BLEU-Metrics : 0 (bad) - 1 (good)

Übersetzung / Translation  $\sim$  Encoder - Decoder - Architecture

Freitag, 22.12.2023

Projekt?

Bidirectional RNN  
(Recurrent Neural Network)

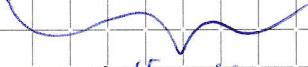
Beam Search

Attention layer

↳ no trainable parameter/weights

output = attention\_layer (input-a, input-b)

output = attention\_layer (vector-a, vector-a)

  
self attention, because 2 inputs are the same

Wozu werden Skip-Verbindungen verwendet?

Welche Vorteile haben Convolutional-Neuronale-Netze gegenüber Multilayer-Perceptronen?

Warum gelten Rekurrente-Neuronale-Netze als veraltet?

Attention-Schichten

Was ist Monte-Carlo-Dropout?

Was sind Token?

Welche Schichten findet man in einer Transformer-Architecture?

Skip-Verbindungen ~ Vanishing-Gradient zu reduzieren

CNN-Vorteile: ~ Weniger Parameter da Input Daten sequentiell mit einem Filter abgerautert werden und weniger Overfitting durch verbesserte Mustererkennung

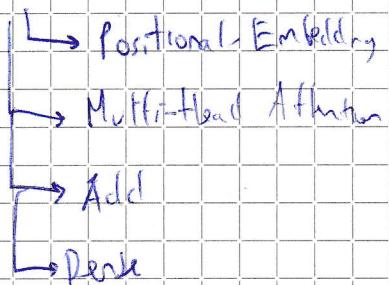
Warum RNN veraltet? ~ Transformer Modelle können sequentielle Daten parallel verarbeiten, dadurch wird zwar viel Arbeitsspeicher benötigt aber das Training ist deutlich schneller.

Attention-Schichten ~ Können Zeitpunkte in sequentiellen Daten untereinander in Bezug setzen ohne eigene trainierbare Parameter

Monte-Carlo-Dropout ~ Ein neuronales Netz mit Dropout-Schichten das für Anwendungsschichten im Trainingsmodus verwendet wird, um variable Ergebnisse zu liefern. Die Ergebnisse können dann statistisch beurteilt werden z.B. durch Mittelwert und Standardabweichung.

Was ist Token? ~ Bestandteil eines Textes, z.B. Wörter, Silben, Buchstaben

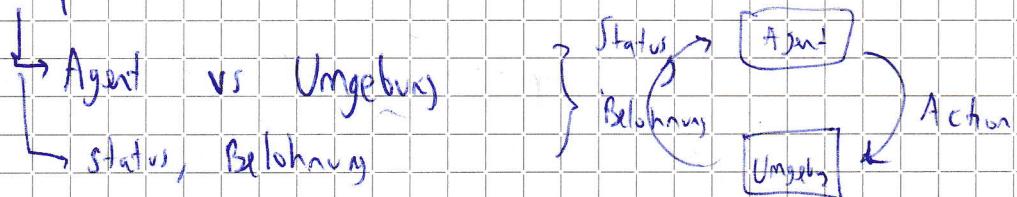
Welche Schichten in Transformer-Architektur? → Embedds



colab.TFResearch.google.com



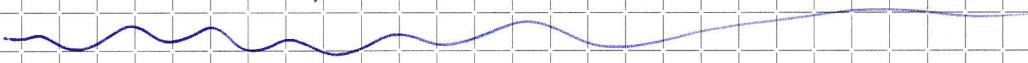
Reinforcement Learning



- Policy Gradient
- Reward function
- Abbruchkriterien

TensorFlow Agents

pip install tf-agents



CNN Bild ~ Verkleinen - Vergrößern.

Freitag, 05.01.2024

Mittwoch – Freitag, 05.01.2024 → Projekt

- Aerodynamische Felddaten von Strömungsprofilen bestimmen.
- Multilabel binär Klassifikation von Charakteren in Futurenabildern
- Galaxien in Teleskopaufnahmen klassifizieren & Lösen von Differentialgleichungen mit neuronalen Netzen
- Interaktive semantische Segmentation von Kleidungsstücken
- Segmentierung von dreidimensionalen Vulkangesteinen
- Durch Super-Resolution Bilder größer skalieren
- Automatisch Bildbeschreibungen mit neuronalen Netzen erstellen
- Mit KI-Tools kurze Videos zu Musik erzeugen
- Prognose der globalen Erwärmung mit rekurrenten und klasischen Modellen

↳ Time Series

↳ Trend

↳ unit root - unit root tests = DF, ADF, KPSS, PP

Note: 95% sehr gut

↳ Upscaling, ResNet ok.

Tensorflow 2.10 ← GPU ok. in NVIDIA { Linux is  
{ ~ CUDA Network ... } better  
compile Tensorflow myself  
or Windows 11 Virtualization (install Linux there)