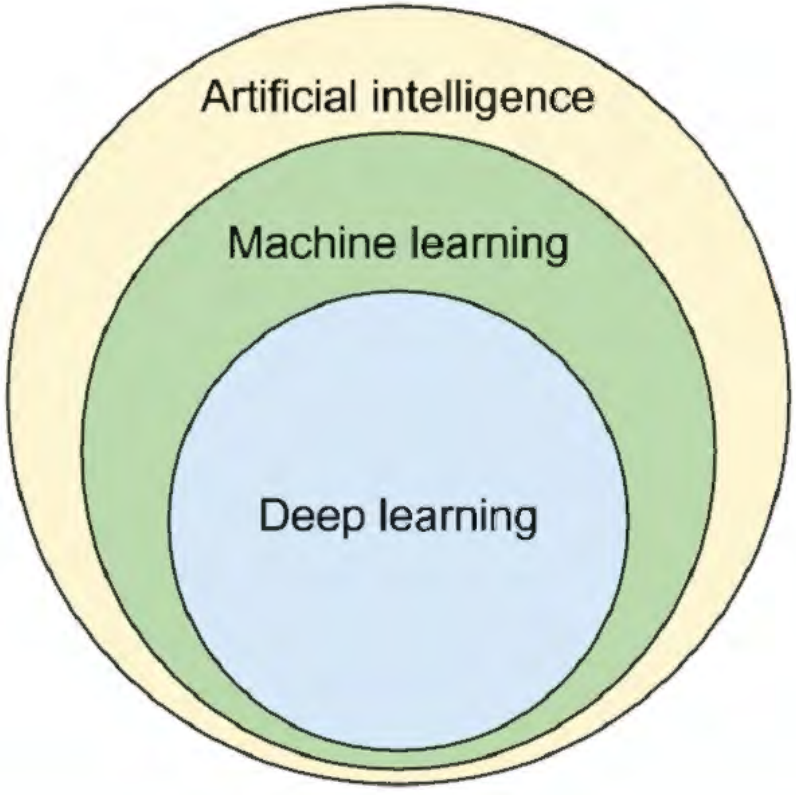
TAG 4

Deep Learning, Neural Network

DEEP LEARNING



Deep Learning = Machine Learning mit Neuralen Netzen als Modell

DEEP LEARNING - MOTIVATION - 2023

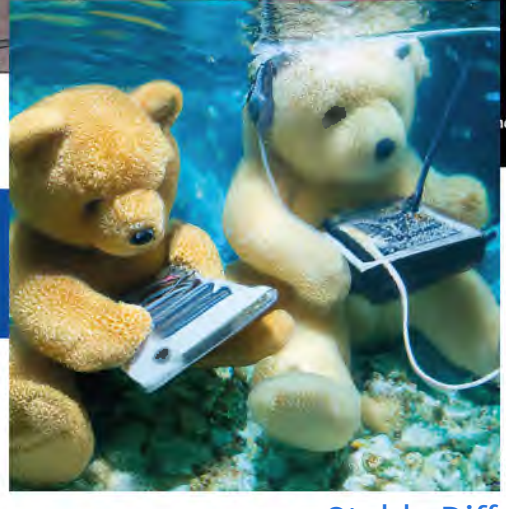
Facebook SAM

Dall-E

AlphaGo und AlphaZero

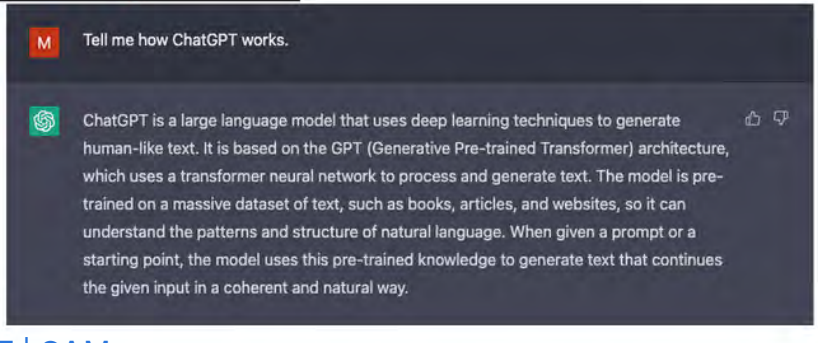
AlphaFold

Text2Speech



StableDiffiusion

ChatGPT



Stable Diffusion | ChatGPT | SAM

DIE WELT VOR DER DEEP LEARNING REVOLUTION

- Klassische Machine Learning Algorithmen trainiert auf manuellen Repräsentation

  Repräsentation = Feature Engineering oder Dimensionality Reduction
- Viele Annahmen, viel Wissen ausprogrammiert
- Beispiele für manuelle Repräsentation
  - Bilder: HAAR Wavelets, HOG
  - Text2Speech Concatenative synthesis, Formant synthesis
  - Schach: Ausprogrammierte Value Function

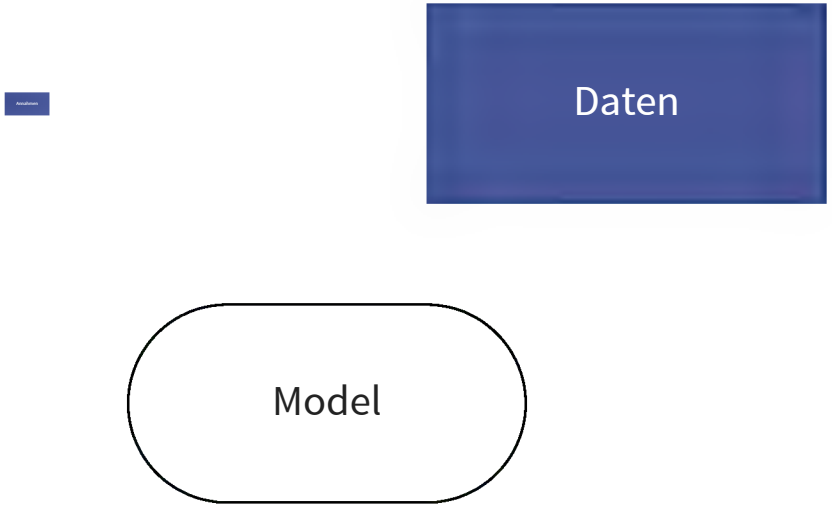
Manuelle Repräsentation bei unstrukturierten Daten wie Bildern und Text schwierig

DEEP LEARNING - MOTIVATION DER DAMALIGEN FORSCHUNG

- Repräsentation automatisch aus Daten lernen
  - Lernen von Symbolischen Regeln (relational learning task)
- Hierarchisch Repräsentationen (layer-by-layer)
- Limit der lokalen Methoden z.B. Decision Tree, rbf-Kernel
  - Interpolation vs. Extrapolation

DEEP LEARNING - WENIGER ANNAHMEN, VIELE DATEN

Weniger Annahmen, viele Daten



Benötigt viele Daten, um ein Modell komplett zu trainieren.

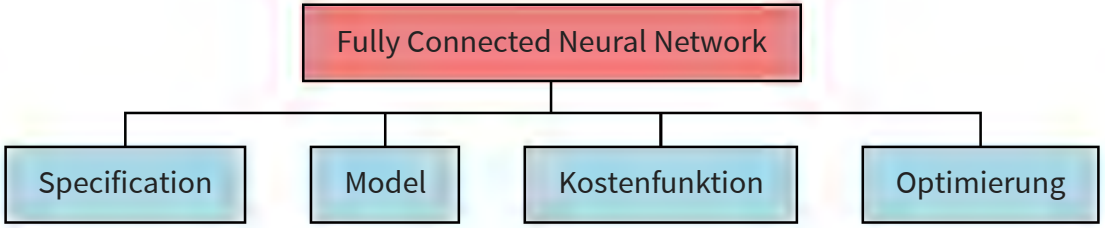
DEEP LEARNING - ÜBERSICHT

- Machine Learning Modell ist ein Neural Network
- Kann Supervised Learning
- Kann Unsupervised Learning
- Benötigt viele Daten
- Benötigt viel Rechenzeit (GPU)
- Meistens beste Wahl bei unstrukturierten Daten (Sequenzen, Text, Bilder, ...)
- Bei strukturierte Daten (z.B. ML Lab Tag 1) meistens nicht beste Wahl.

DEEP LEARNING - GRUNDIDEE - WAS IST EIN NEURALES NETZ?

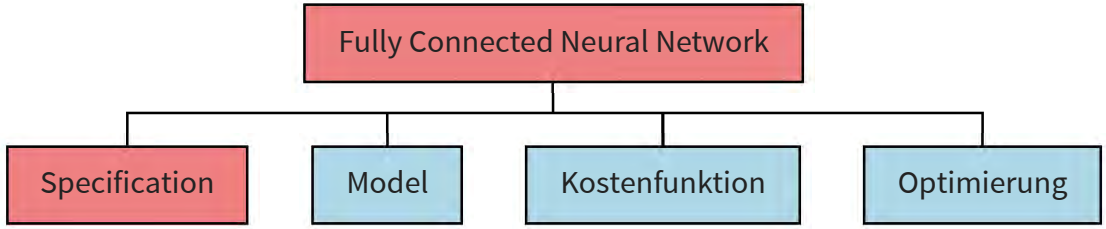
- Inspiriert vom biologischen Gehirn
- Neuronen die mit einander verbunden sind (Knoten in einem Graph).
- Neuronen haben eingehende und ausgehende Verbindungen (Kanten).
- Neuronen feuern (geben einen Wert weiter), wenn eingehende Neuronen genug stark feuern.

FULLY CONNECTED NEURAL NETWORK



Synonym: Fully Connected Neural Network =
Multi Layer Perceptron (MLP) = Neural Network (NN)

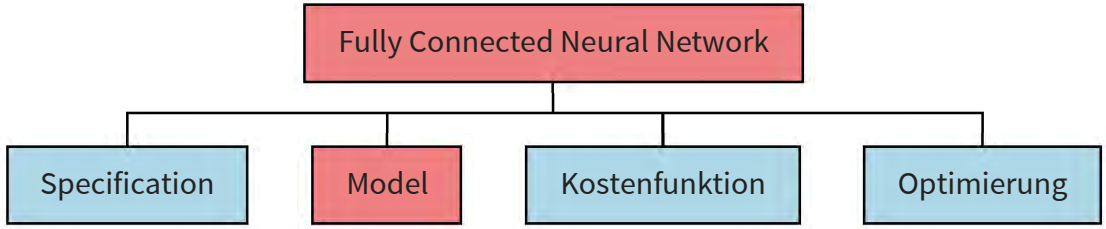
FULLY CONNECTED NEURAL NETWORK



NEURAL NETWORK - SPECIFICATION

- Was ist das Ziel
- Was ist die Kostenfunktion
- Welche Features wählen wir
- Kategorische Features müssen encoded werden.
- Numerische Features müssen standardisiert werden.

FULLY CONNECTED NEURAL NETWORK

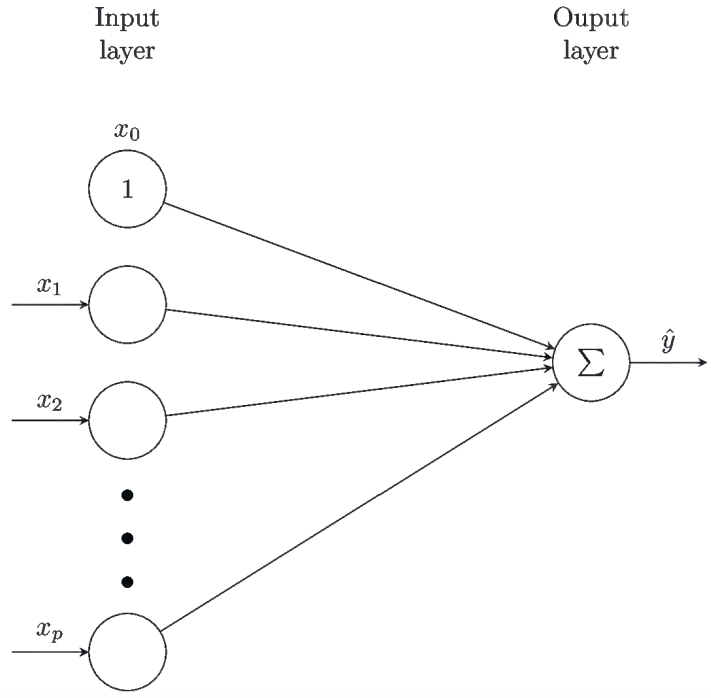


LINEARE REGRESSION - ALS (NEURAL) NETWORK

ALS FORMELL

$$\hat{y} = \beta_0 + \beta_1 * x_1 + \ldots + \beta_p * x_p$$

ALS (NEURAL) NETZWERK



Die Berechnung der Linearen Regression als (Neural) Network dargestellt.

HIDDEN LAYER(S) HINZUFÜGEN

ALS FORMELL

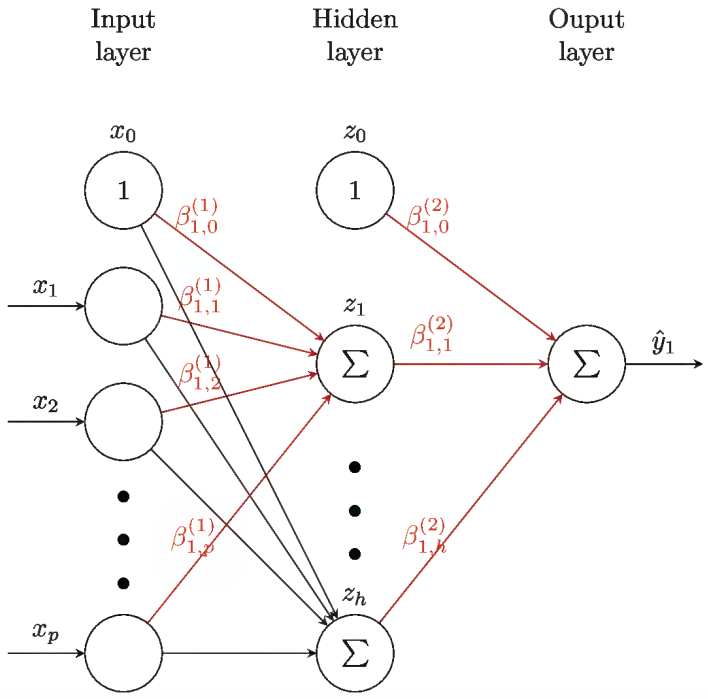
$$z_1 = \beta_{1,0}^{(1)} + \beta_{1,1}^{(1)} x_1 + \ldots + \beta_{1,p}^{(1)} x_p$$

$$\ldots$$

$$z_h = \beta_{h,0}^{(1)} + \beta_{h,1}^{(1)} x_1 + \ldots + \beta_{h,p}^{(1)} x_p$$

$$\hat{y} = \beta_{1,0}^{(2)} + \beta_{1,1}^{(2)} z_1 + \ldots + \beta_{1,h}^{(2)} z_h$$

ALS (NEURAL) NETZWERK



Hidden Layer entspricht mehreren Linearen Regressionen nebeneinander und hintereinander!

AKTIVIERUNGSFUNKTION HINZUFÜGEN

ALS FORMELL

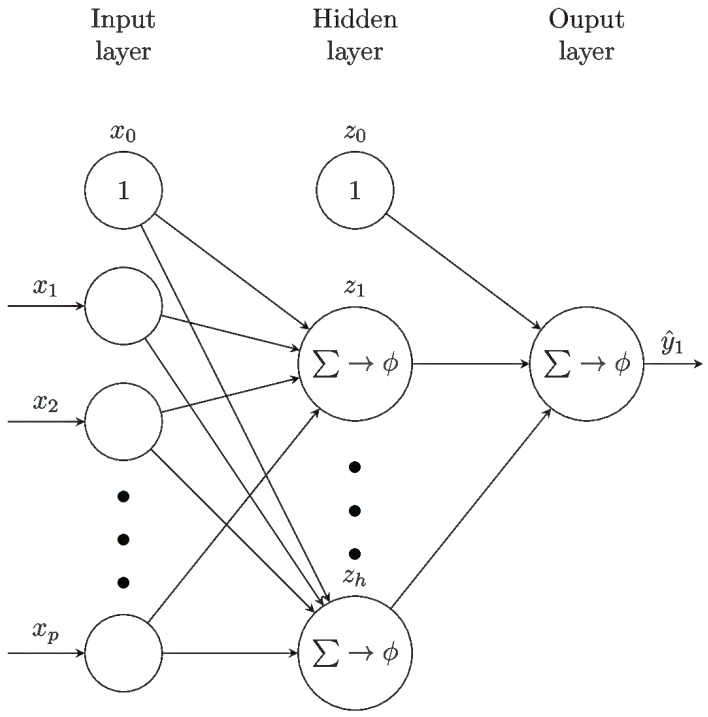
$$z_1 = \phi(\beta_{1,0}^{(1)} + \beta_{1,1}^{(1)} x_1 + \ldots + \beta_{1,p}^{(1)} x_p)$$

$$\ldots$$

$$z_h = \phi(\beta_{h,0}^{(1)} + \beta_{h,1}^{(1)} x_1 + \ldots + \beta_{h,p}^{(1)} x_p)$$

$$\hat{y} = \phi(\beta_{1,0}^{(2)} + \beta_{1,1}^{(2)} z_1 + \ldots + \beta_{1,h}^{(2)} z_h)$$

ALS NEURAL NETZWERK



ϕ nennt man die Aktivierungsfunktion

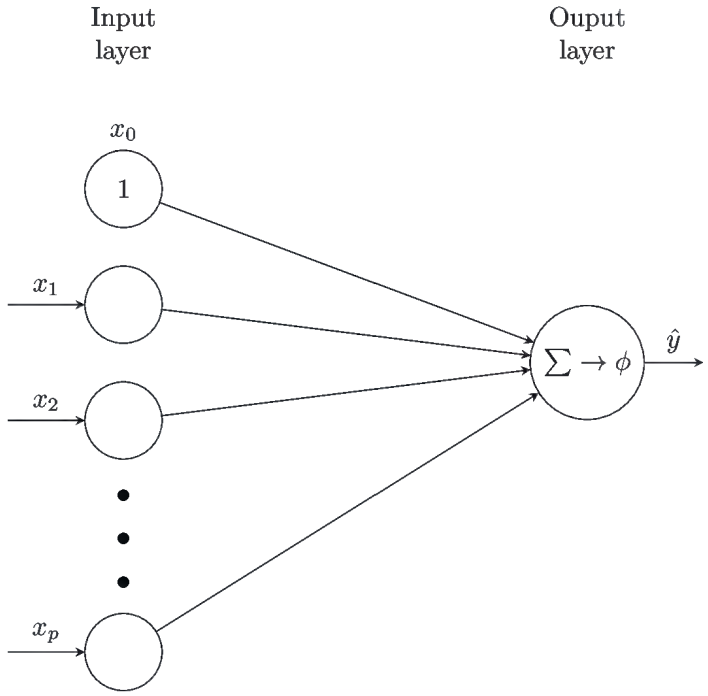
Die Aktivierungsfunktion macht das Modell non-linear. Heute wird oft ReLU verwendet (Liste von gängigen Aktivierungsfunktionen).

LOGISTIC REGRESSION - ALS (NEURAL) NETZWERK

ALS FORMELL

$$\hat{y} = \phi(\beta_0 + \beta_1 * x_1 + \ldots + \beta_p * x_p)$$

ALS (NEURAL) NETZWERK



Hier ist die Aktivierungsfunktion der Sigmoid.

NEURAL NETWORK - MEHRERE OUTPUTS MÖGLICH

ALS FORMELL

$$z_1 = \phi(\beta^{(1)}_{1,0} + \beta^{(1)}_{1,1} x_1 + \ldots + \beta^{(1)}_{1,p} x_p)$$

$$\ldots$$

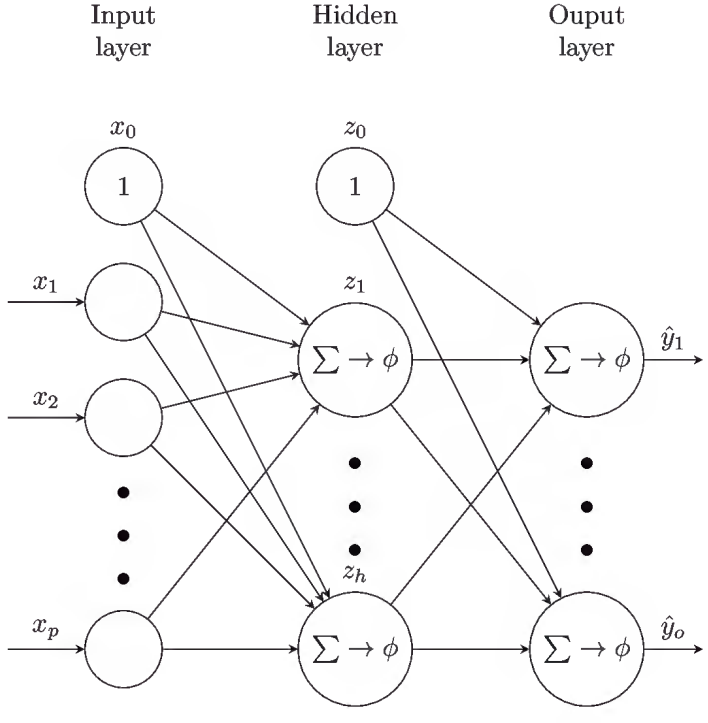
$$z_h = \phi(\beta^{(1)}_{h,0} + \beta^{(1)}_{h,1} x_1 + \ldots + \beta^{(1)}_{h,p} x_p)$$

$$\hat{y}_1 = \phi(\beta^{(2)}_{1,0} + \beta^{(2)}_{1,1} z_1 + \ldots + \beta^{(2)}_{1,h} z_h)$$

$$\ldots$$

$$\hat{y}_o = \phi(\beta^{(2)}_{o,0} + \beta^{(2)}_{o,1} z_1 + \ldots + \beta^{(2)}_{o,h} z_h)$$

ALS NETZWERK



LOGISTIC REGRESSION - MEHRERE OUTPUTS

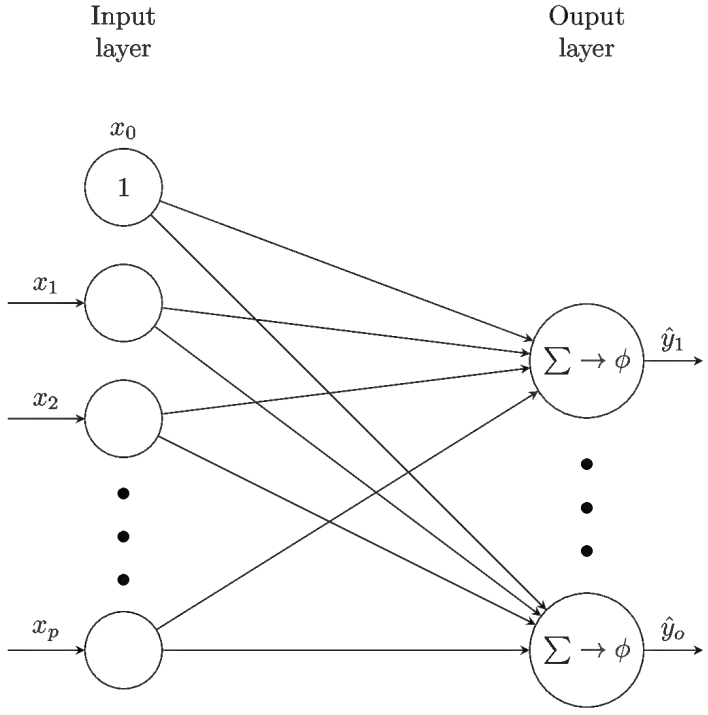
ALS FORMELL

$$\hat{y}_1 = \phi(\beta_{1,0} + \beta_{1,1} * x_1 + \ldots + \beta_{1,p} * x_p)$$

$$\ldots$$

$$\hat{y}_o = \phi(\beta_{o,0} + \beta_{o,1} * x_1 + \ldots + \beta_{o,p} * x_p)$$

ALS NETZWERK



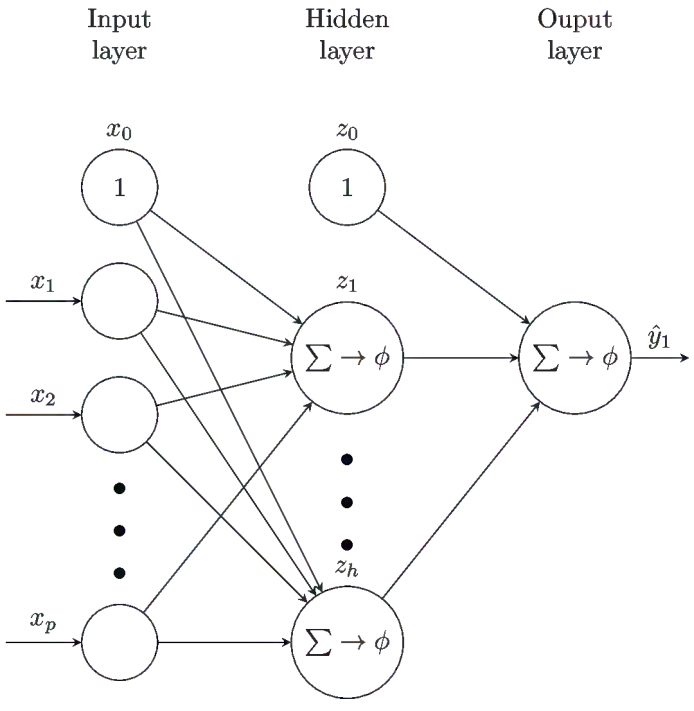
Logistische Regression mit mehreren Outputs.
Hier ist die Aktivierungsfunktion Softmax.

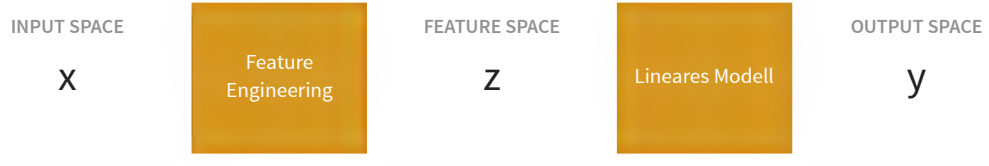
DEEP LEARNING - FRAMEWORK FÜR MODELLE

- Modellkomplexität einfach anpassbar
  - mehr Hidden Layers => Mehr lernbare Parameter
  - weniger Hidden Layers => Weniger lernbare Parameter
  - In einem Layer mehr Nodes => Mehr lernbare Parameter
  - In einem Layer weniger Nodes => Weniger lernbare Parameter
- Kostenfunktion
  - Kostenfunktion muss ableitbar sein.
  - Kann Problem spezifisch gewählt werden.
- Viele weitere Möglichkeiten (Deep Learning)
  - Wir können die Architektur (Verbindungen) vom Netzwerk anders gestalten, entsprechend dem zugrunde liegenden Problem (z.B. CNN und RNN).

https://playground.tensorflow.org

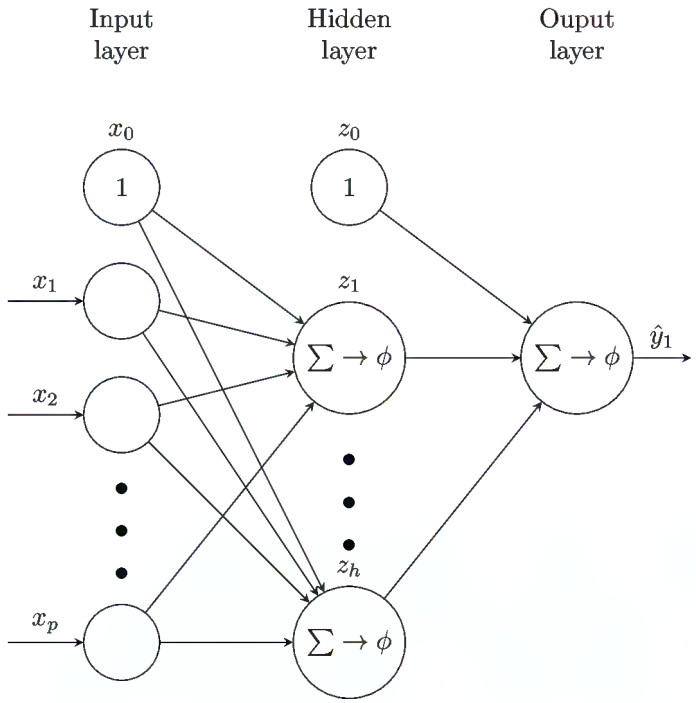
DEEP LEARNING - GELERNTES FEATURE ENGINEERING

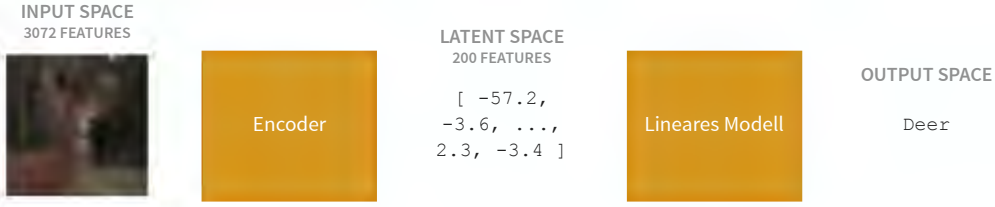




Die Hidden Layers können als lernbares Feature Engineering betrachtet werden. Die gelernten Features haben gewisse praktische Einschränkungen.

DEEP LEARNING - GELERNTE DIMENSIONALITY REDUCTION

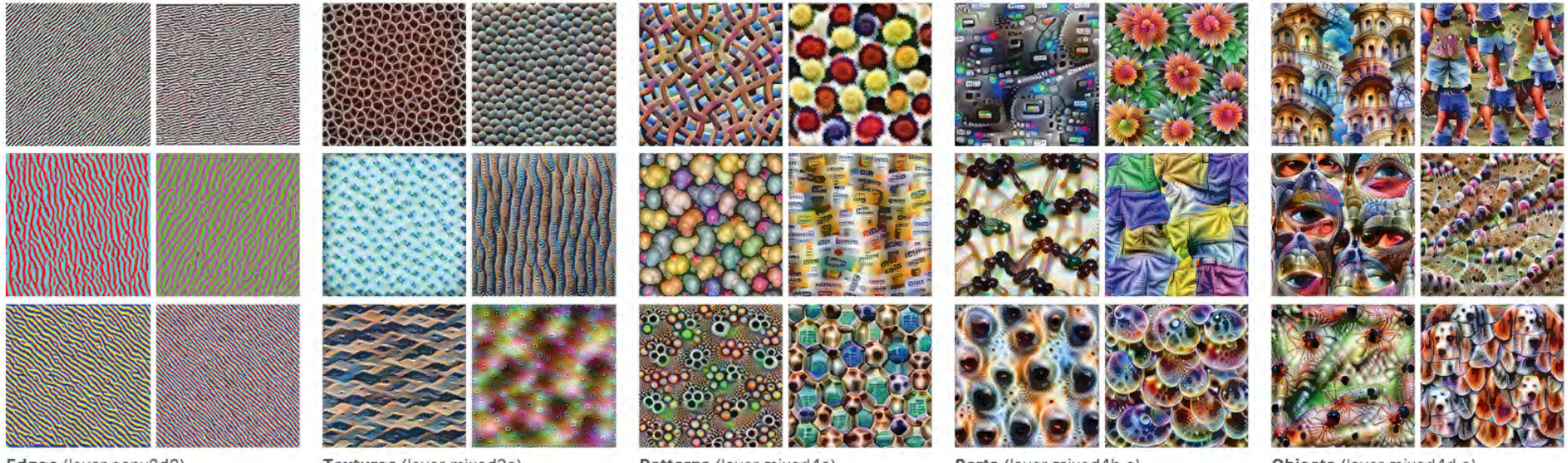




Die Hidden Layers können als lernbare Dimensionality Reduction betrachtet werden. Dabei lernt das Modell, welche Informationen für das spezifische Ziel weggeworfen werden können.

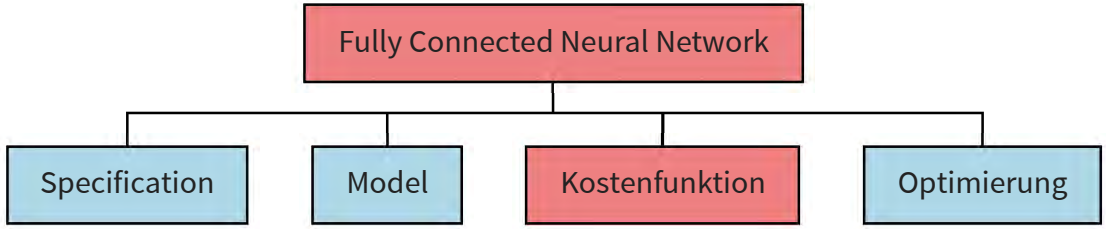
GOOGLENET - GELERNTE FEATURES, GELERNTE DIMENSIONALITY REDUCTION

GoogLeNet



Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a) **Parts** (layer mixed4b,c) **Objects** (layer mixed4d,e)

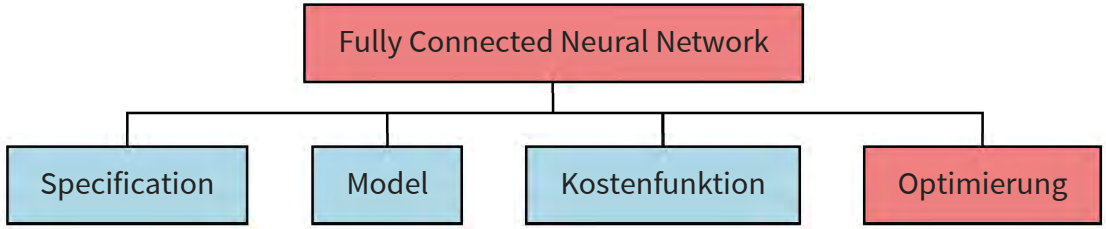
FULLY CONNECTED NEURAL NETWORK



PROBLEMSPEZIFISCHE KOSTENFUNKTION

- Bei Regression: MSE / MAE
- Bei Klassifikation: Maximum Liklihood
- Bei Unsupervised: Reconstruction Error
- ...

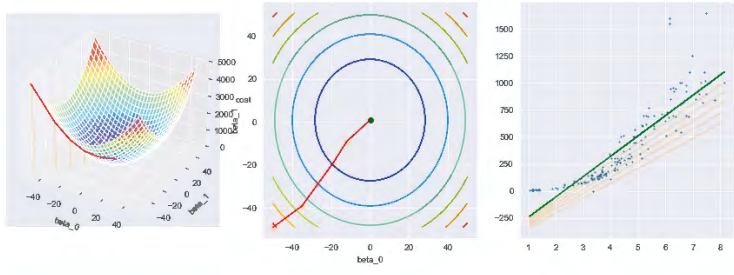
FULLY CONNECTED NEURAL NETWORK



GRADIENT DESCENT

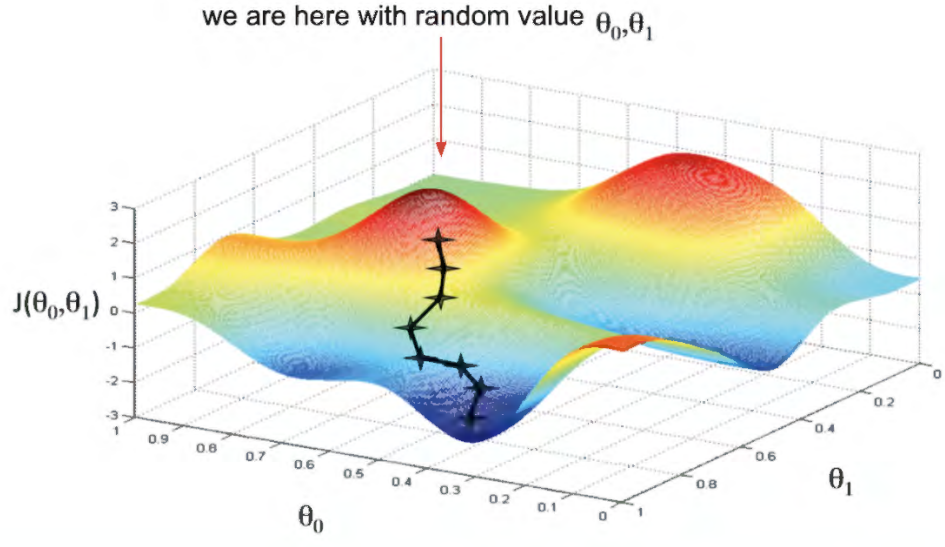
REMINDER: CONVEX PROBLEM (TAG 1)

Bei Linearen Modellen



NON-CONVEX PROBLEM

Bei NN mit Hidden Layer(s)



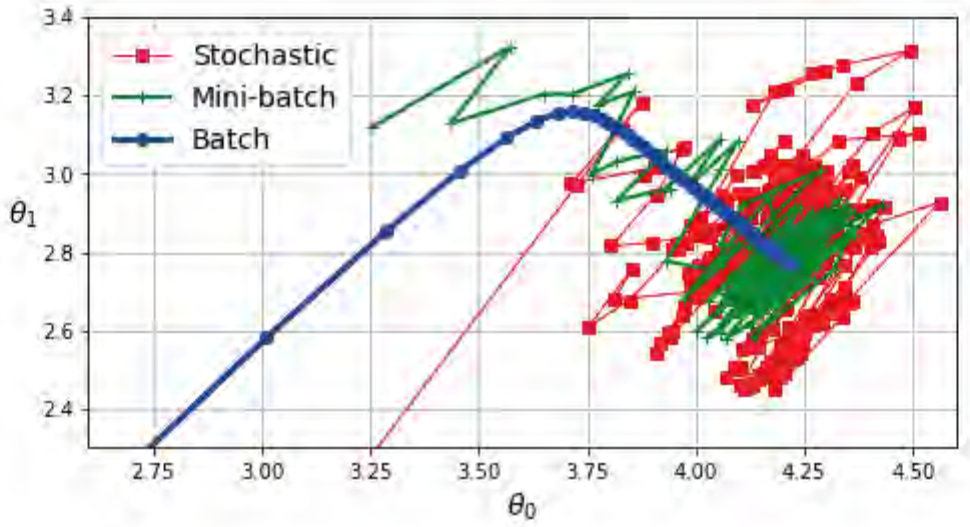
Gleicher Algorithmus. Aber bei non-convex Kostenfunktionen, können wir in einem lokalen Minimum landen.

STOCHASTIC UND BATCH GRADIENT DESCENT

- Wie funktionierte (Batch) Gradient Descent? (Tafel)
- Stochastic Gradient Descent berechnet die Richtung zum Minimum mit nur einem Sample.
  - Update viel schneller zu berechnen.
  - Update sehr noisy (oft in falsche Richtung)
- Mini Batch Gradient Descent berechnet die Richtung zum Minimum mit z.B. 128 Sample.
  - Update schneller zu berechnen.
  - Update noisy (ab und zu in falsche Richtung)

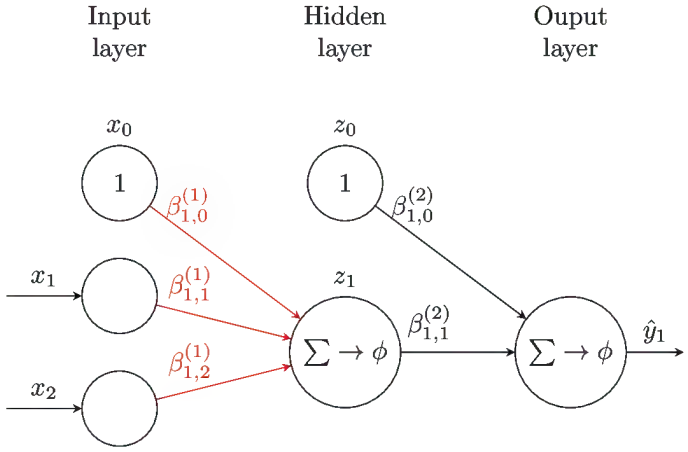
In Literatur heisst Mini Batch Gradient Descent oft auch Stochastic Gradient Descent.

BATCH VS. STOCHASTIC VS. MINI BATCH GRADIENT DESCENT



BACKPROPAGATION

- Algorithmus für das effiziente Berechnen der Gradienten der Kostenfunktion in einem Neural Network
- Nutzt die Chain-Rule von der Analysis um Zwischenergebnisse zu cachen.



$$\frac{\partial}{\partial \beta^{(2)}_{1,0}} = ? \; ? \; ? \qquad \frac{\partial J}{\partial \beta^{(1)}_{1,0}} = \frac{\partial J}{\partial z_1} ? \frac{\partial z_1}{\partial \beta^{(1)}_{1,0}} ?$$

$$\frac{\partial}{\partial \beta^{(1)}_{1,0}} = ? \; ? \; ?$$

$$\frac{\partial}{\partial \beta^{(1)}_{1,1}} = ? \; ? \; ? \qquad \frac{\partial J}{\partial \beta^{(1)}_{1,1}} = \frac{\partial J}{\partial z_1} ? \frac{\partial z_1}{\partial \beta^{(1)}_{1,1}} ?$$

$$\frac{\partial J}{\partial \beta^{(1)}_{1,2}} = \frac{\partial J}{\partial z_1} \frac{\partial z_1}{\partial \beta^{(1)}_{1,2}}$$

Optimierungsalgorithmus von Neuralen Netzen ist (Batch) Gradient Descent.
Backpropagation hilft "nur" beim effizienten Berechnen der Gradienten.

[ADDITIONAL RESOURCES]

- Neuronen anders vernetzen
  - ResNet (skip connection)
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Transformer (behandelt im NLP)
- Learning und Regularization: Momentum, Adam, Learning Rate Decay, Dropout, Batch Normalization

Grundidee immer: Annahmen treffen, um das Lernen zu vereinfachen ohne es (stark) einzuschränken.

FEEDFORWARD NEURAL NETWORK - CODE

`neural_network.ipynb`

ÜBUNGSZEIT (60 MINUTEN)

`exercise/neural_network.ipynb`

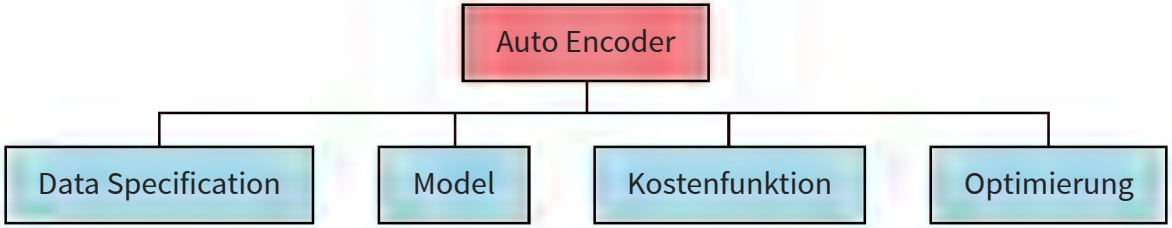
QUESTIONS

QUESTIONS

1. Was ist ein Hidden Layer?

2. Was ist eine Aktivierungsfunktion?

AUTO ENCODER

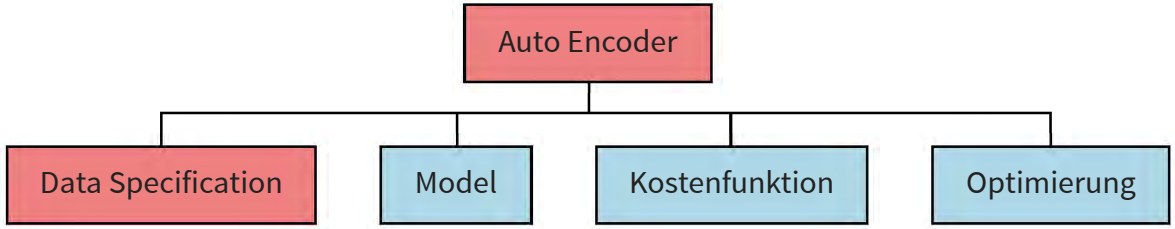


Auto Encoder ist eine Dimensionality Reduction.

AUTO ENCODER - MOTIVATION

- Nicht-lineare Dimensionality Reduction anhand von Daten lernen.
- Dazu verwenden wir ein Neural Network mit Bottleneck und Reconstruction Error.

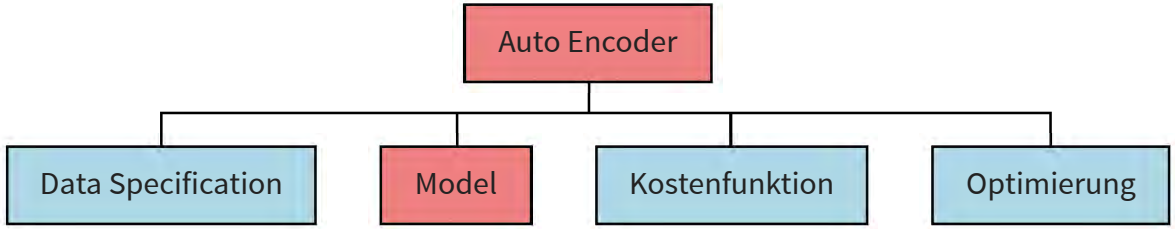
AUTO ENCODER



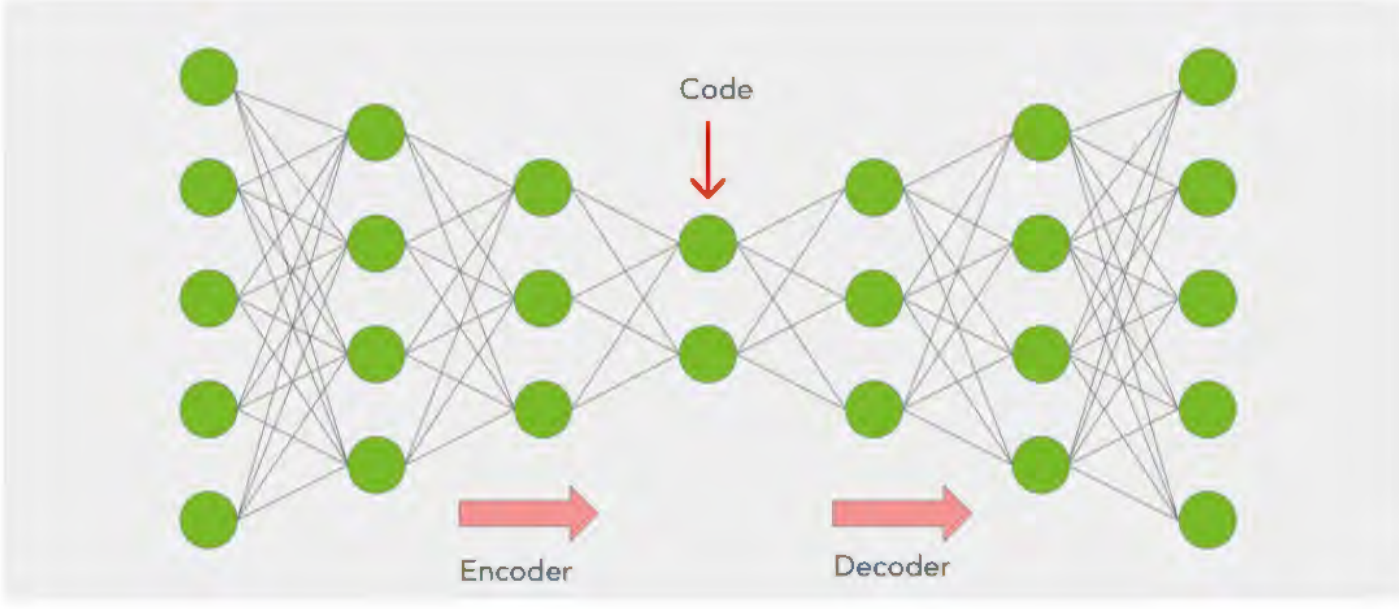
AUTO ENCODER - DATA SPECIFICATION

1. Welche Features haben wir, z.B. `pixel`
2. Kategorische Features müssen encoded werden.
3. Numerische Features müssen standardisiert werden.

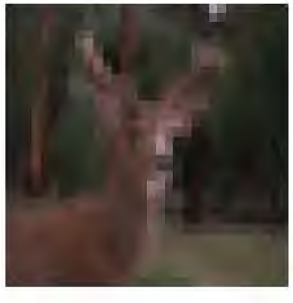
AUTO ENCODER



AUTO ENCODER - MODEL



INPUT SPACE
3072 FEATURES



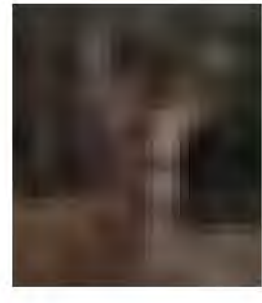
Encoder

LATENT SPACE
200 FEATURES

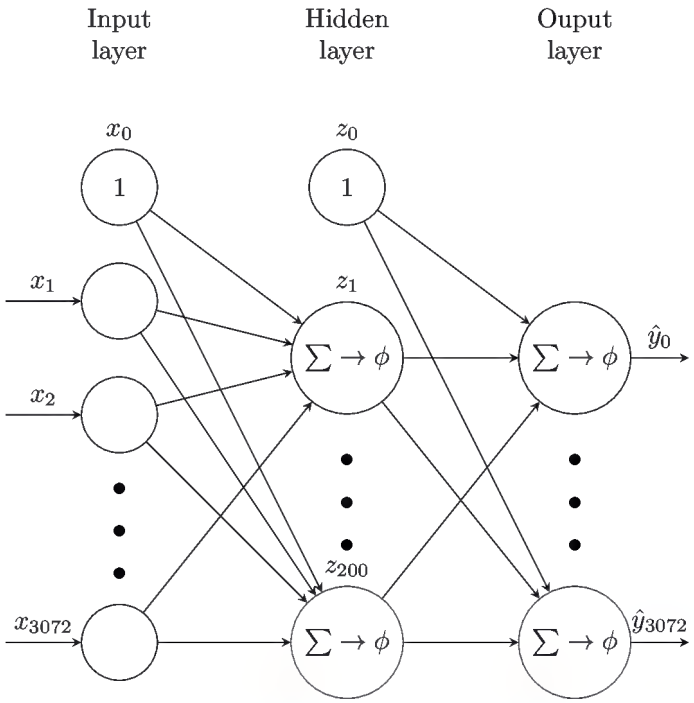
[-57.2,
-3.6, ...,
2.3, -3.4]

Decoder

RECONSTRUCT
INPUT SPACE
3072 FEATURES



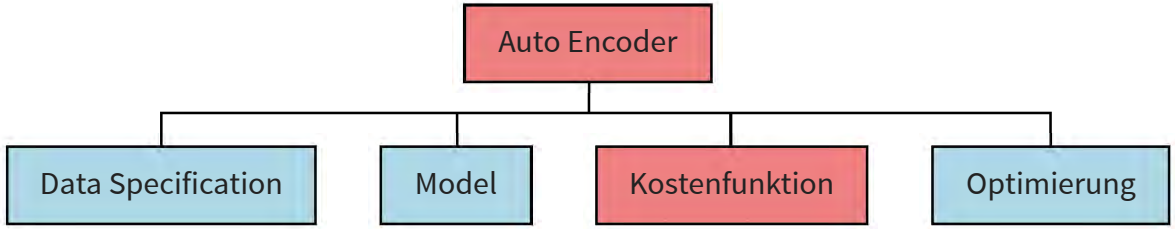
AUTO ENCODER - MODEL - BEISPIELSWEISE FÜR CIFAR



Hier Reduktion von 3072 auf 200 Dimensionen.

Genaue Architektur ist "Hyper Parameter"

AUTO ENCODER

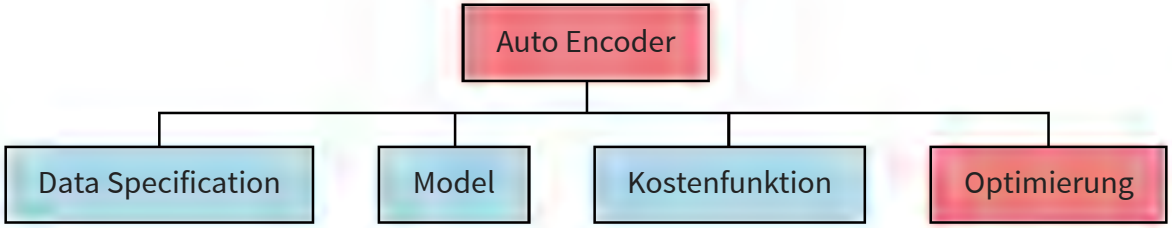


AUTO ENCODER - KOSTENFUNKTION

$$J(\beta) = \quad \|X - \hat{X}\|_2$$

Ander Kostenfunktionen möglich, z.B. L1 Distanz.

AUTO ENCODER

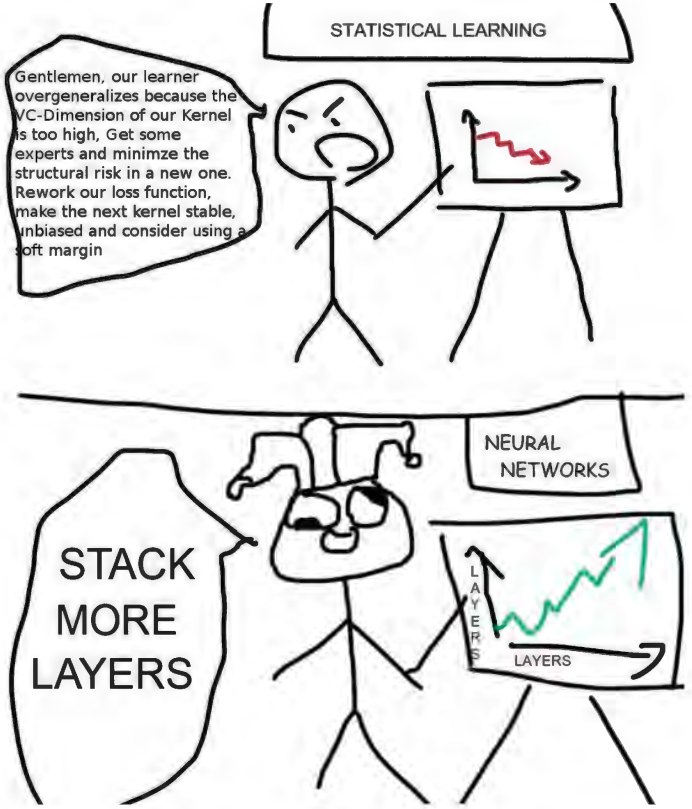


Batch Gradient Descent

AUTO ENCODER VS. PCA

Wenn Encoder und Decoder linear sind (z.B. lineare Aktivierungsfunktion), dann lernt der Auto Encoder den gleichen Latent Space wie PCA.

DEEP LEARNING IST EIN ~~SPIELPLATZ~~ FRAMEWORK!



Aufzeigen, wie unterschiedlich und problemspezifisch Architekturen in der Praxis sind.

NEURAL NETWORK - LENET (1998)

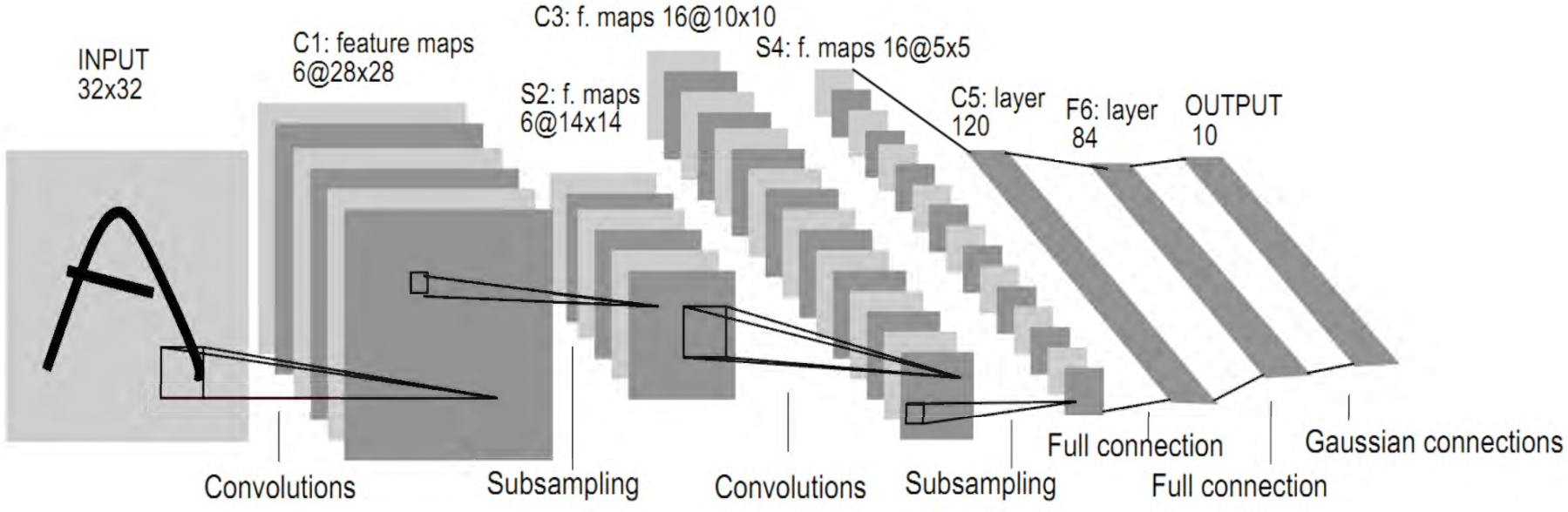


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

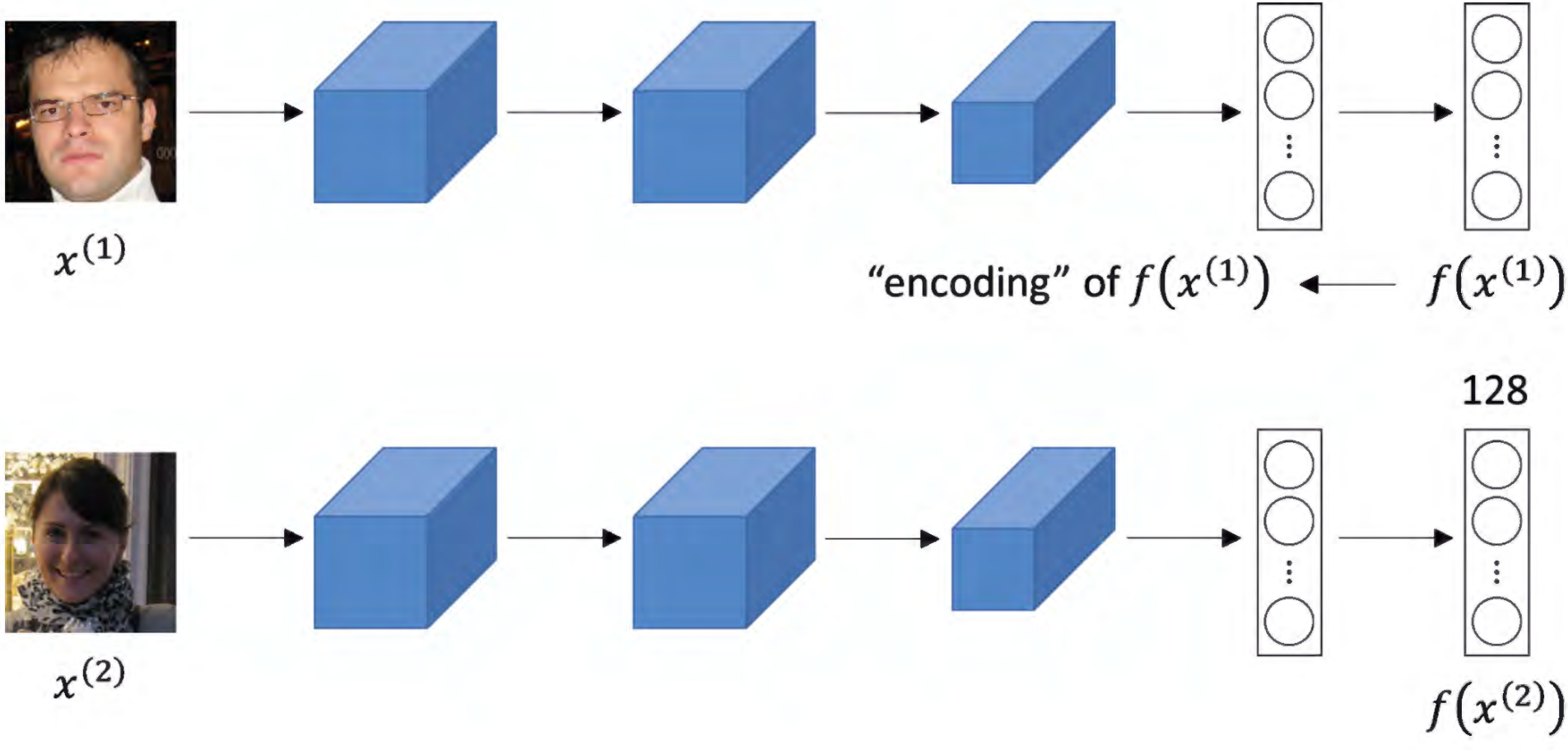
Paper: gradient-based learning applied to document recognition

3D visualization

NEURAL NETWORK - LENET (1998)

- 7 Layers
- 60'000 lernbare Parameter
- Sigmoid statt ReLU (ReLU noch nicht entdeckt)
- Dataset: MNIST
  - 60'000 train data
  - 10' '000 test data

NEURAL NETWORK - SIAMESE (2015)

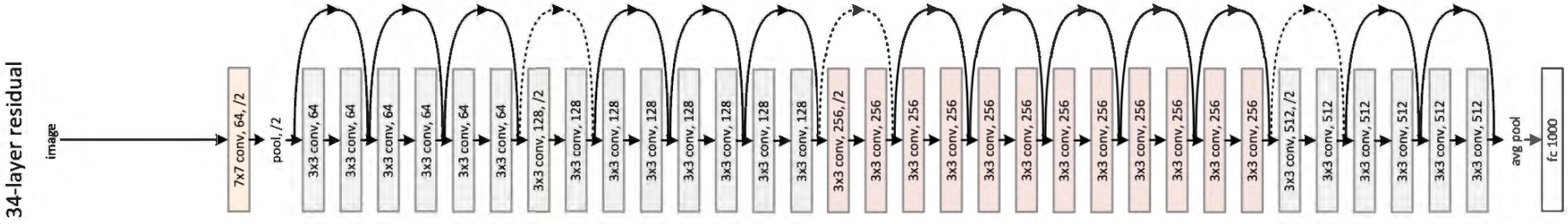


Paper: Siamese Neural Networks for One-shot Image Recognition

NEURAL NETWORK - SIAMESE (2015)

- Lerne Ähnlichkeit von Inputs (z.B. Bilder)
- Erlaube Generalisierung auf neue Klassen (one-shot learning)
- Wird oft für Face Detection verwendet (z.B. Facenet)

NEURAL NETWORK - RESNET (2015)



Paper: Deep Residual Learning for Image Recognition

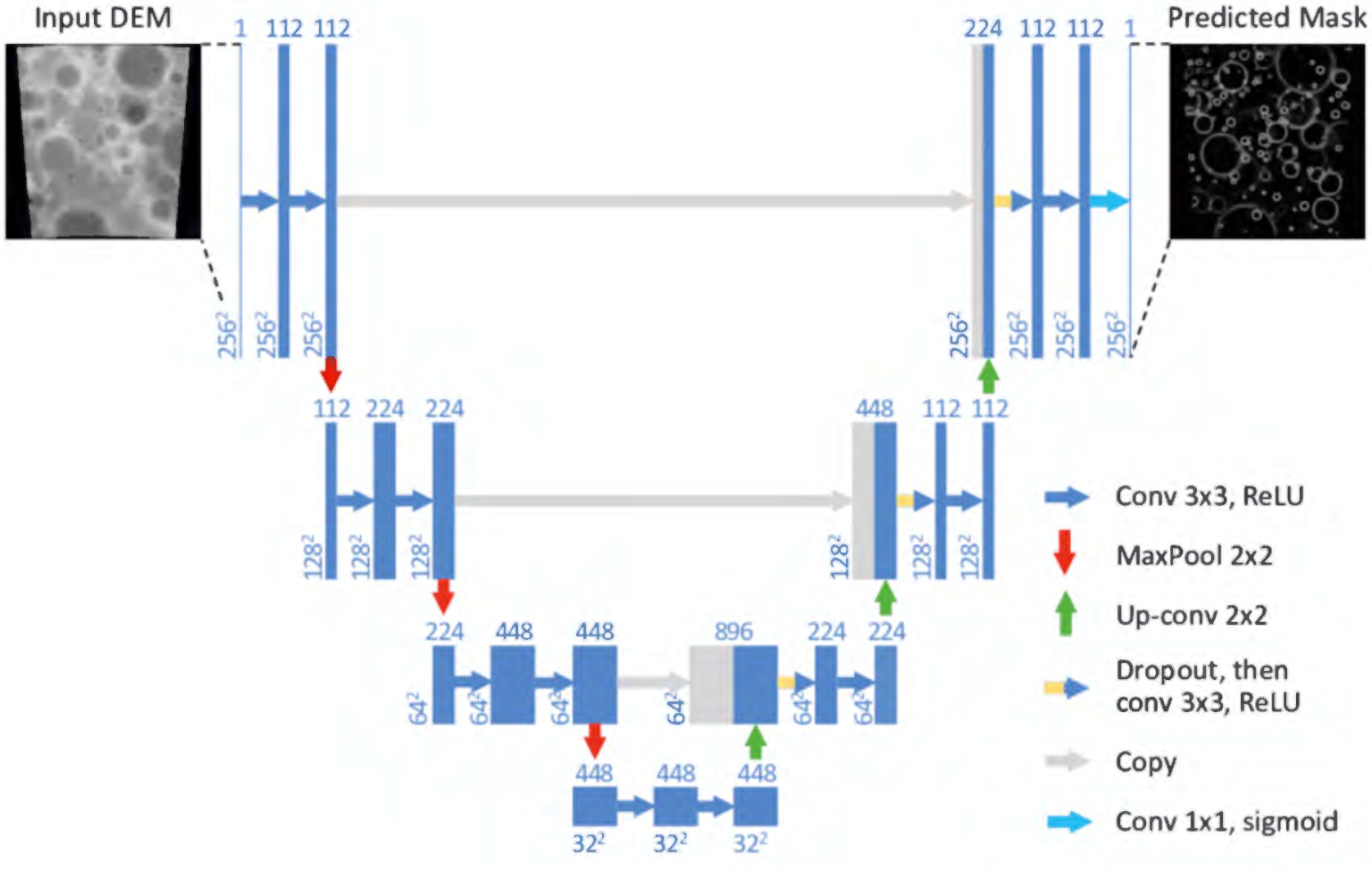
3D visualization (50 layers)

NEURAL NETWORK - RESNET (2015)

Skip-Connections machen tiefe Netzwerke trainierbar

Netzwerk mit 1202 Layers trainiert

NEURAL NETWORK - U-NET (2015)

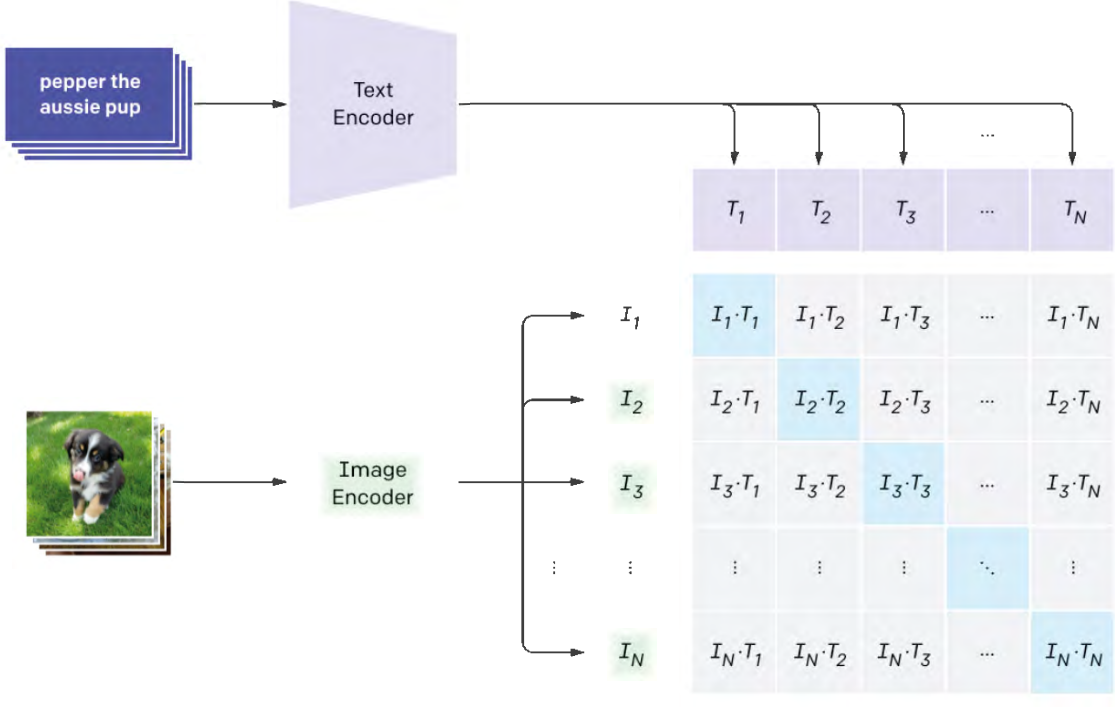


NEURAL NETWORK - U-NET (2015)

- Löst Verluste von räumlichen Informationen (nötig für z.B. Bildsegmentierung)
- Encoder, Decoder mit Skip-Connections
- Skip Connections helfen, räumliche Informationen zu erhalten
- Bildsegmentierung revolutioniert

NEURAL NETWORK - CLIP (2021)

1. Contrastive pre-training

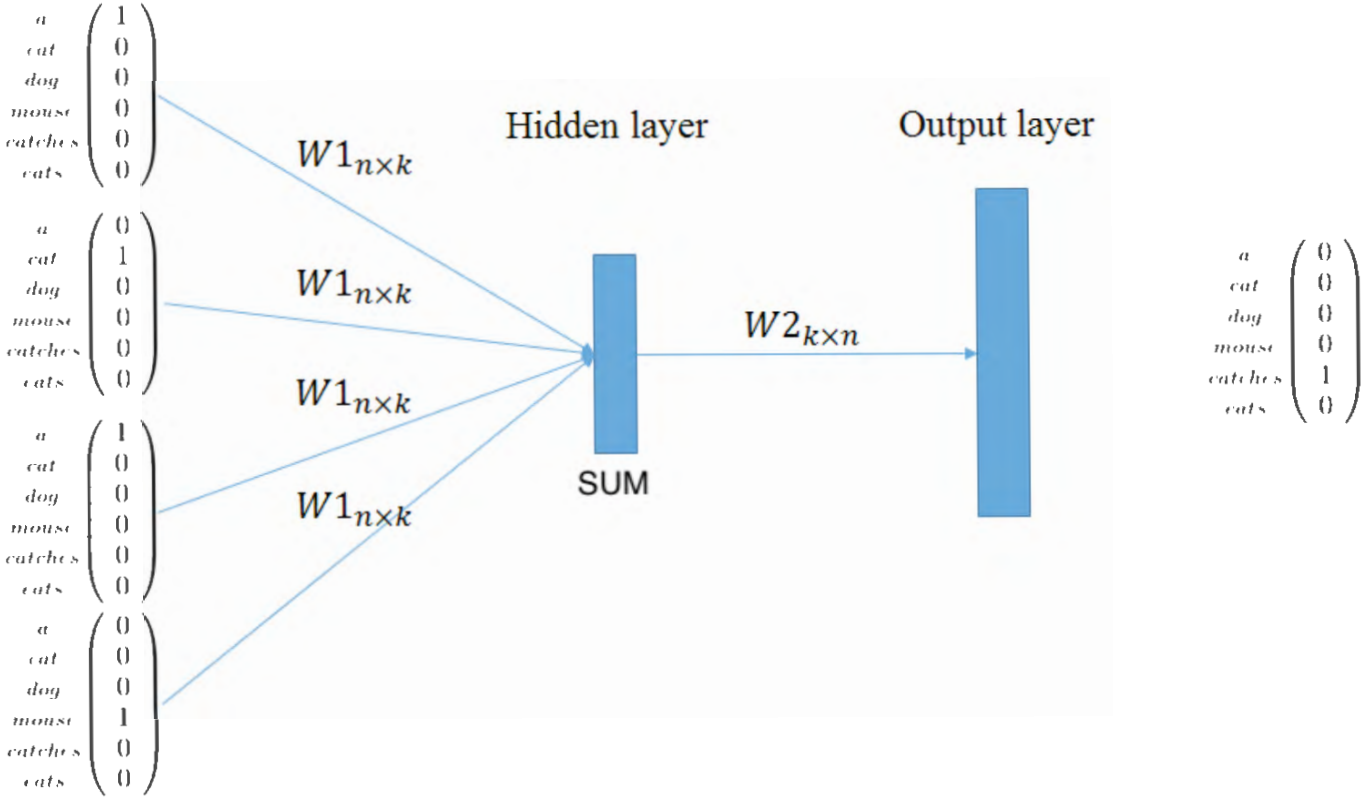


NEURAL NETWORK - CLIP (2021)

- Gegeben ein Bild, welcher von 32'768 Texten gehört zum Bild.
- Trainiert auf 400 Millionen Bildern und Texten
- Auf 256 GPUs für 2 Wochen trainiert
- Text-Encoder und Bild-Encoder (Transformer Architektur)
- Gemeinsamer Latent-Space für Bilder und Texte, erlaubt zero-shot learning

NEURAL NETWORK - WORD2VEC (2013) - CBOW

Task: `a cat ? a mouse`



Paper: Efficient Estimation of Word Representations in Vector Space

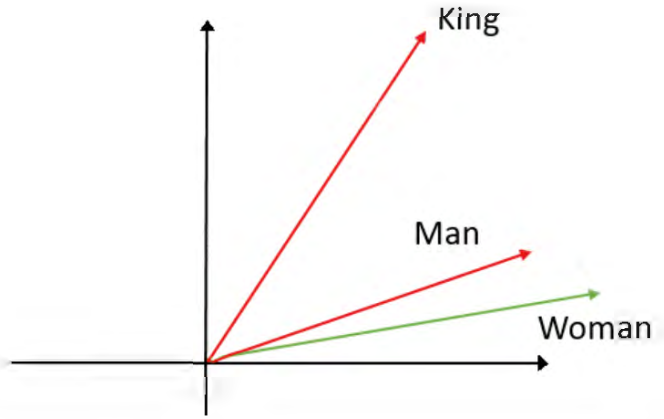
NEURAL NETWORK - WORD2VEC (2013)

- 1,6 Milliarden Wörter im Train Set (Google News)
- 1 Million häufigste Wörter (1 Million Dimensionen)
- Ziel: Lernen von Word-Embeddings
  - Reduktion auf l Dimensionen

NEURAL NETWORK - WORD2VEC - WHAT IS LEARNED?

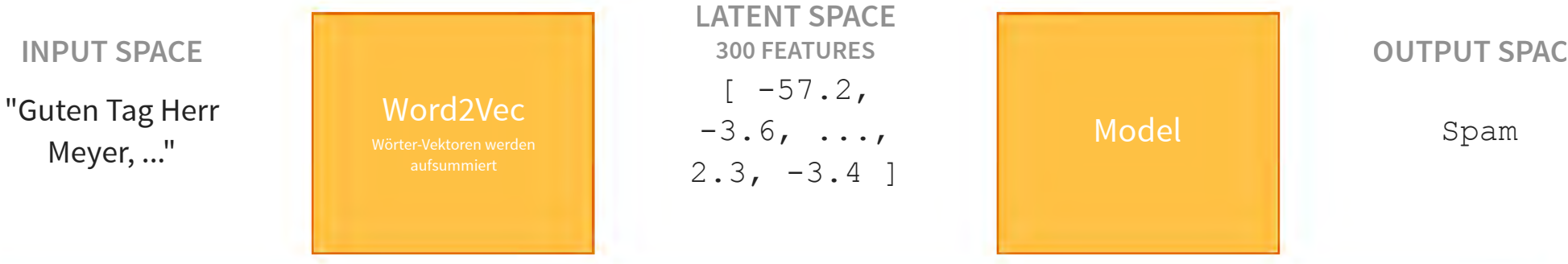
- Synonyme werden ähnlich encoded
- Bedeutung der Wörter:

```
King        - Man     + Woman    ~ Queen
```




```
Germany     - Berlin   + Paris    ~ France
Germany     - Berlin   + London   ~ England
easiest     - easy     + lucky    ~ luckiest
mice        - mouse    + dollar   ~ dollars
impossibly - possibly + ethical ~ unethical
```

WORD2VEC ALS FEATURE PREPROCESSING



Der Latent Space beschreiben präziser den Text.
Model hat dadurch ein leichteres Spiel.

Word2Vec kann hier als sinnvolles Text-Encoding verstanden werden
Anstatt z.B. One Hot Encoding, wo alle Wörter gleich weit entfernt sind.

Transfer Learning: Vortrainiertes Wissen von anderem Task wiederverwenden

Word2Vec ist veraltet (2013), bessere Encodings existieren (z.B. BERT).

DATA SCIENCE PITFALLS - BIAS IN MACHINE LEARNING

- Pre-existing
  - Historischer (oder aktueller) Bias in den Daten
  - Bias in den Algorithmen, Problemstellung
- Technical, z.B.
  - Limitierte Rechenzeit
  - Design: erste sichtbare Resultate vs. später sichtbare Resultate
  - Verzerrte Zufallszahlengenerierung
- Emergent
  - Unpredictable correlations z.B. Muster in den Daten lässt auf Demographie zurückführen (indirekter Pre-existing bias)
  - Feedback loops: Algorithmus beeinflusst Welt, Welt beeinflusst Algorithmus

PRE-EXISTING: AMAZON BEWERBUNGEN

Goal: Vorfiltern der Bewerbungen für einen Job

- Gelernt: Pre-Existing Bias gegen Frauen in Tech-Berufen
- Action: Entfernen des Geschlechts als Feature
- Gelernt: Unpredictable correlations ”all-women highschool", "women's chess club" negative features

Amazon hat das Projekt eingestampft

EMERGENT: TWITTER CROPPING ALGORITHM

Goal: "crop an image to an easily-viewable size"

Sharing learnings about our image cropping algorithm

FEEDBACK-LOOP: SOCIAL MEDIA

Idee: Benutzer klicken, was sie mögen, also zeige Inhalte die Benutzer klickt!

- Gelernt: Benutzer klicken auf extreme Inhalte und Inhalte, denen der Benutzer zustimmt.
- Anpassung der Publishers: Inhalte werden extremer, Clickbaits werden standard
- Anpassung der User: Einseitige Berichterstattung führt zu Bubbles, andere Seite wird negative gesehen (Spaltung der Gesellschaft)

PRE-EXISTING: BIAS IN WORD2VEC

WAS WURDE AUCH GELERNT?

- ```
  Computer Programmer - Man + Woman ~ Homemaker
  surgeon - he + she ~ nurse
  football - he + she ~ volleyball
  burly - he + she ~ blond
  ```
- "John" ist näher an "Programmierer" als "Mary"

Word Embeddings lernen Pre-Existing Bias in Daten

Paper: Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings
  ```

WAS TUN GEGEN BIAS IN MACHINE LEARNING

- Bias aus Dataset entfernen ist schwierig (wegen unpredictable correlations)
- Gewisse Probleme sollten allenfalls (noch) nicht mit Machine Learning automatisiert werden
- Bias in Machine Learning ist ein aktives Forschungsgebiet
- Regularisierung von Machine Learning Systemen wird kommen (ähnlich dem drei Phasen-Test für Medikamente nur in einer anderen Form)

Bewusstsein für diesen Effekt schaffen

DATA SCIENCE - ZUKUNFT

HEUTE

- Modelle auf Train-Set trainieren etc.

ZUKUNFT / HEUTE

- Foundation models (BERT, GPT-4, SAM) mit few/zero shot-learning oder Teil eines eigenen Modells mittels Transfer Learning
- AutoML: "Entwickler mit geringem ML-Fachwissen qualitativ hochwertige Modelle trainieren"

Zukunft ungewiss

ÜBUNGSZEIT (60 MINUTEN)

`exercise/neural_network.ipynb`

PRÜFUNGSFRAGEN - BEISPIELE

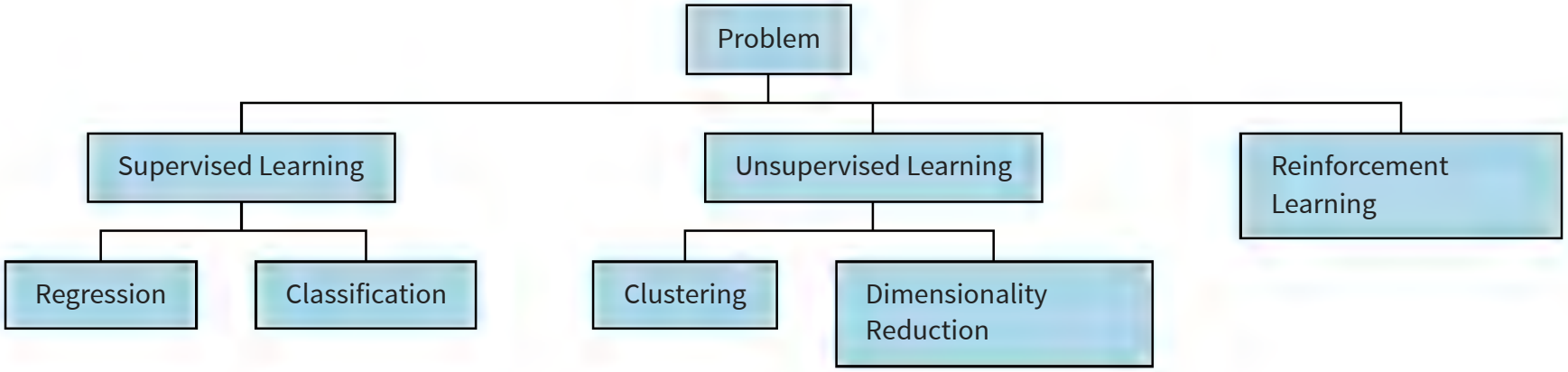
example-exam-questions.ipynb

TIPPS FÜRS LERNEN

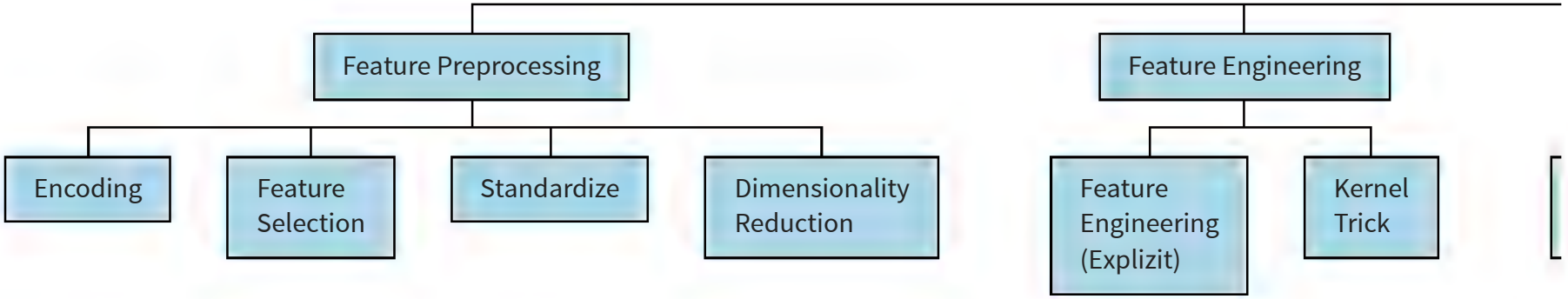
- CTRL+SHIFT+F für durchsuchen der Folien
- Hyperlinks in Folien

- Videos auf YouTube (Testlauf)
- Andere Quellen beziehen
  - sklearn User Guide
  - Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

GELERNTE MACHINE LEARNING PROBLEME



GELERNTE MACHINE LEARNING KONZEPTE



EVALUATION