Karlsruher Institut für Technologie

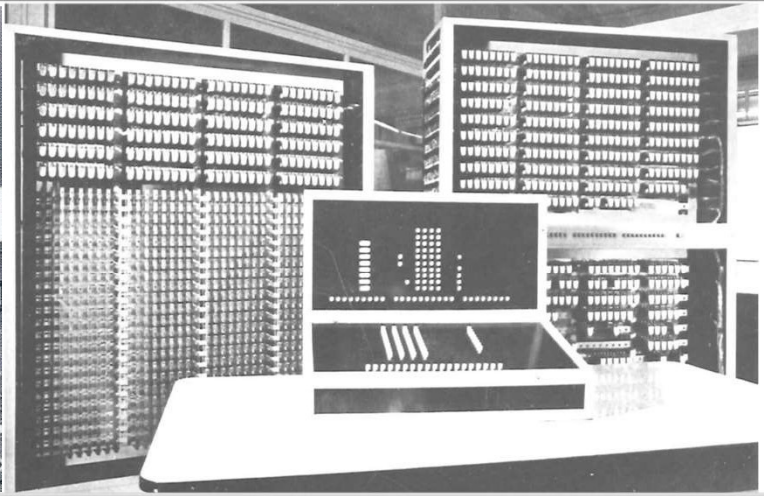
Institut für Technik der Informationsverarbeitung

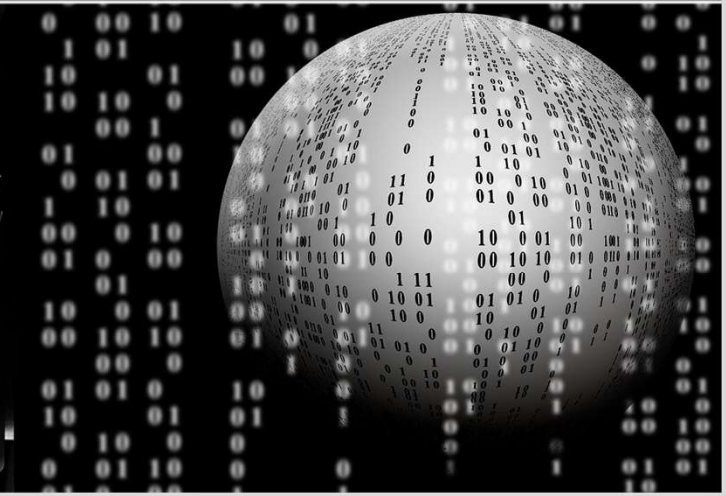
Institutsleitung
Prof. Dr.-Ing. J. Becker
Prof. Dr.-Ing. E. Sax
Prof. Dr. rer. nat. W. Stork

6.2 Big Data & Maschinelles Lernen (B)

Informationstechnik II

Prof. Dr.-Ing. Eric Sax

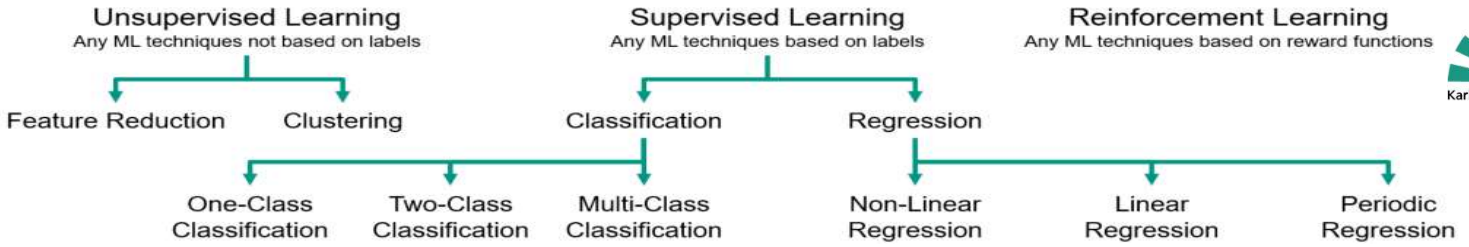


Inhalt IT2

6. Big Data & Maschinelles Lernen

- Definitionen, Verwendung (Charakteristik, Risiken, Chancen)
- Motivation und Anwendungsfälle
- Data Science Prozesse:
  - KDD
  - CRISP-DM
- CRISP-DM im Detail
  - Business Understanding
  - Data Understanding
  - Data Preparation
  - Modeling
  - Evaluation
- Infrastruktur für Big Data

Machinelles Lernen

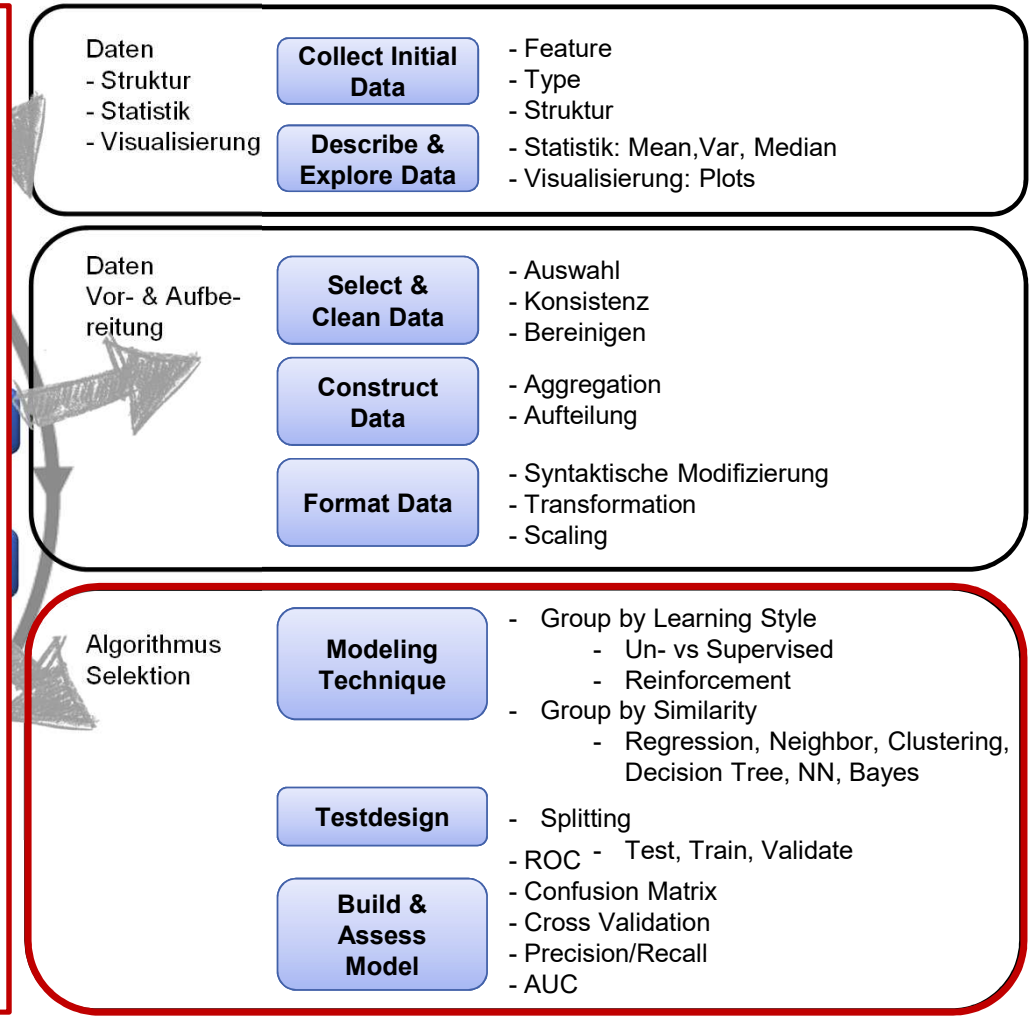


Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

CRISP-DM im Detail

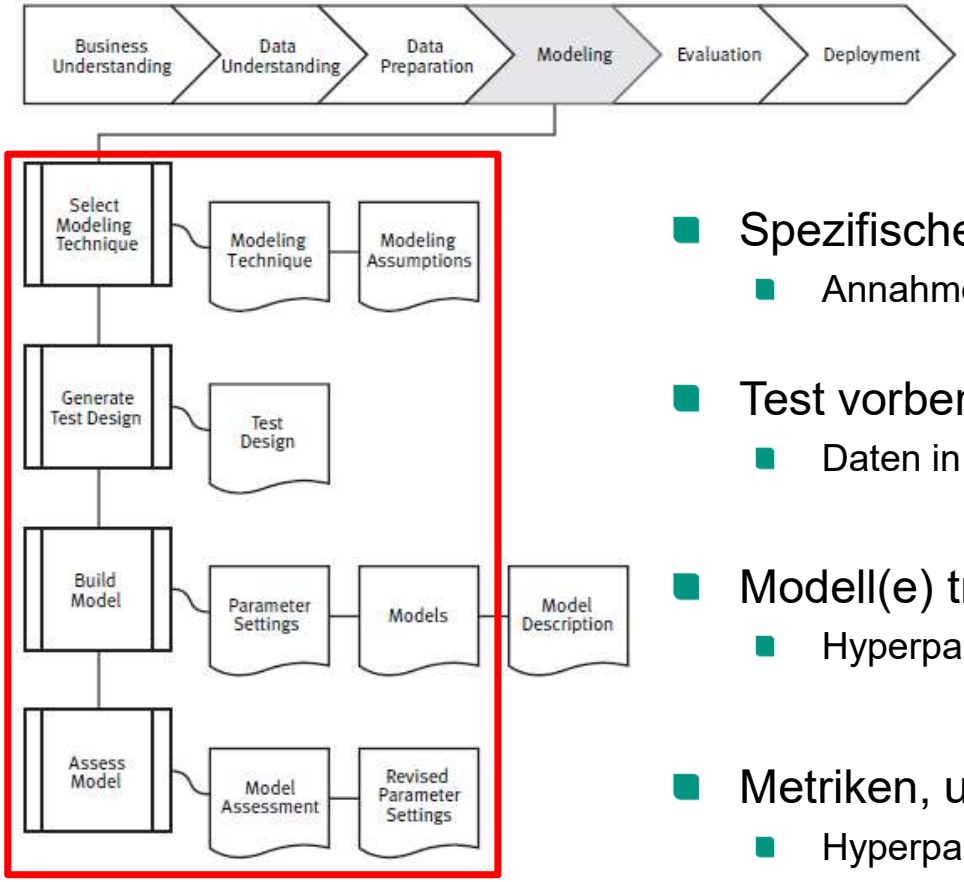
Modellierung / Algorithmus Auswahl

- Spezifische Algorithmen auswählen
  - Annahmen des Algorithmus festhalten (z.B. Gleichverteilung von Daten)
- Test vorbereiten:
  - Daten in Training, Test und Validierung aufteilen; Cross-Validierung
- Modell(e) trainieren
  - Hyperparameter anpassen (Thresholds, Struktur Neuronaler Netze)
- Metriken, um trainierte Modelle zu bewerten
  - Hyperparameter anpassen und bestes Modell auswählen



CRISP-DM im Detail: Modeling

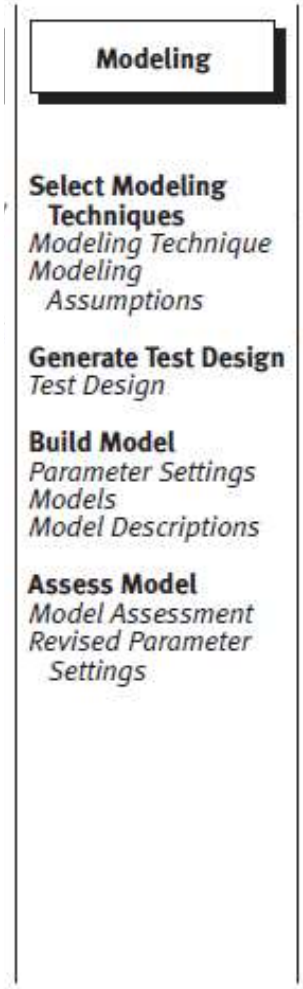
- Modeling = **A**lgorithmen-Auswahl, **A**ssessment, Parameter-**A**daption (**A³**)



- Spezifische Algorithmen auswählen
  - Annahmen des Algorithmus festhalten (z.B. Gleichverteilung von Daten)
- Test vorbereiten:
  - Daten in Training, Test und Validierung aufteilen; Cross-Validierung
- Modell(e) trainieren
  - Hyperparameter anpassen (Thresholds, Struktur Neuronaler Netze)
- Metriken, um trainierte Modelle zu bewerten
  - Hyperparameter anpassen und bestes Modell auswählen

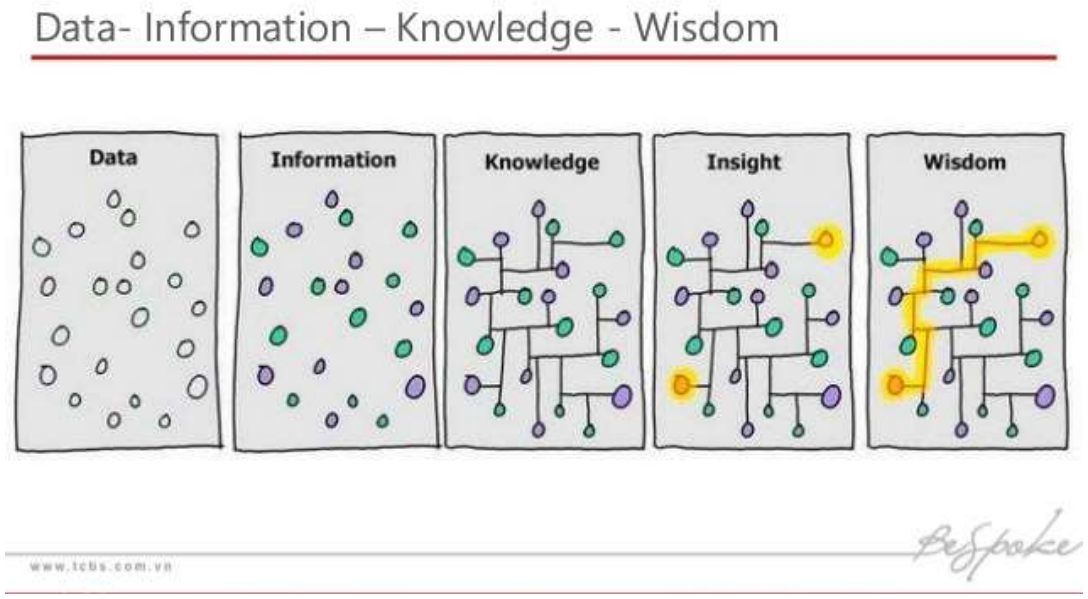
CRISP-DM im Detail: Data Modeling

- Auswahl der Modellierungsmethode
- Test vorbereiten
- Modell erstellen
- Modell bewerten



Worum geht es denn gleich nochmal?

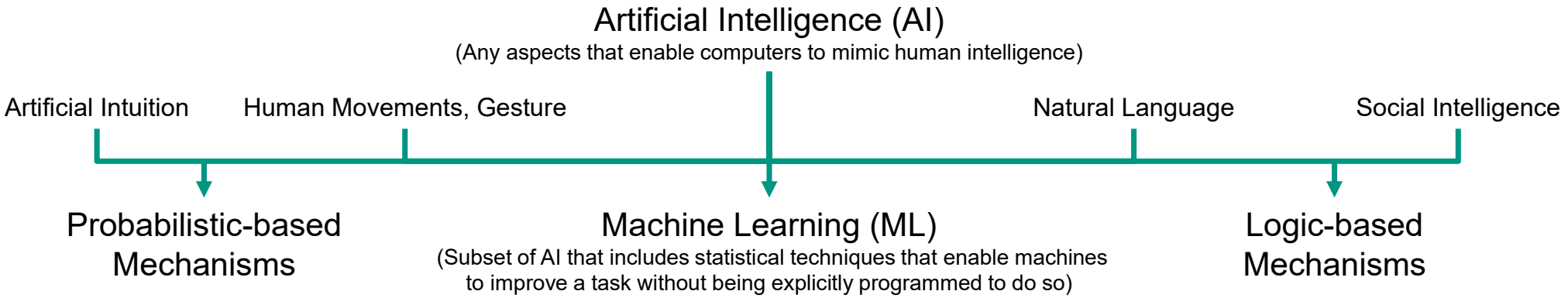
- Klassisch geben wir Kriterien vor, die zu einer Klassifikation bzw. Ausführung führen.
- Intelligente Verfahren lernen aus Erfahrung, mit Belohnung, ohne dass konkrete Beschreibungen a priori vorliegen.



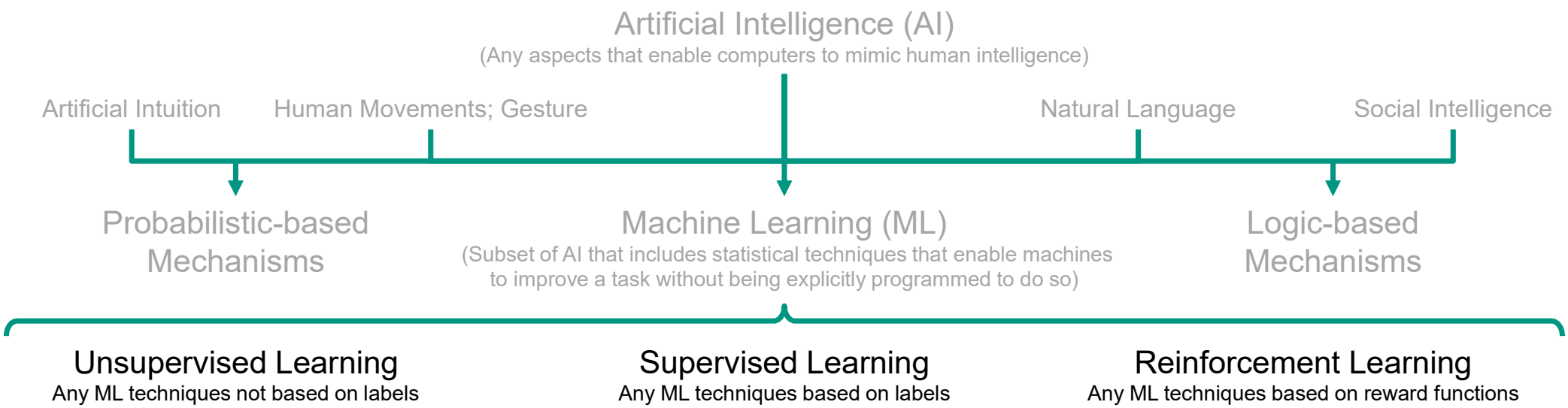
Worum geht es denn gleich nochmal?

- Klassisch geben wir Kriterien vor, die zu einer Klassifikation bzw. Ausführung führen.
- Intelligente Verfahren lernen aus Erfahrung, mit Belohnung, ohne dass konkrete Beschreibungen a priori vorliegen.

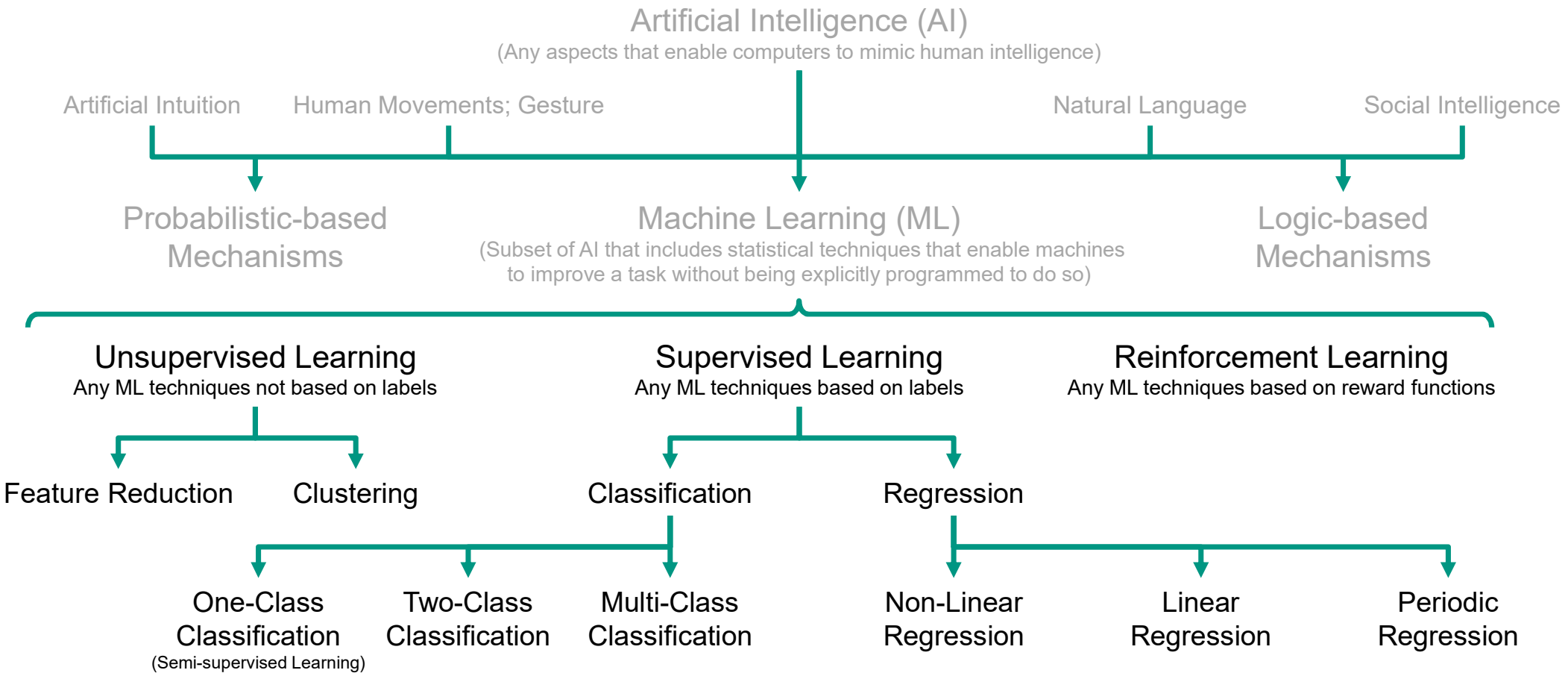
“Artificial Intelligence” und “Machine Learning”



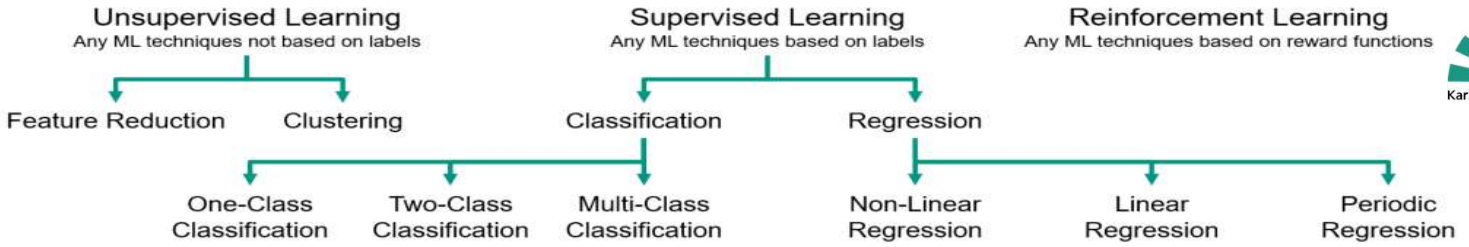
Lernmethoden



Spezielle Verfahren

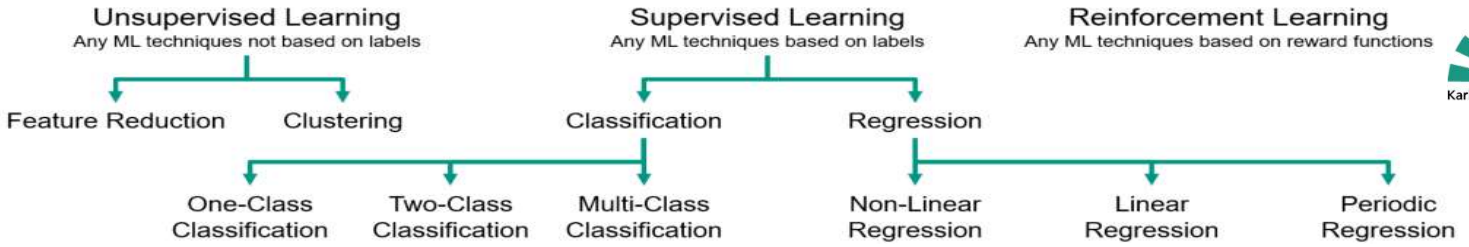


Machinelles Lernen



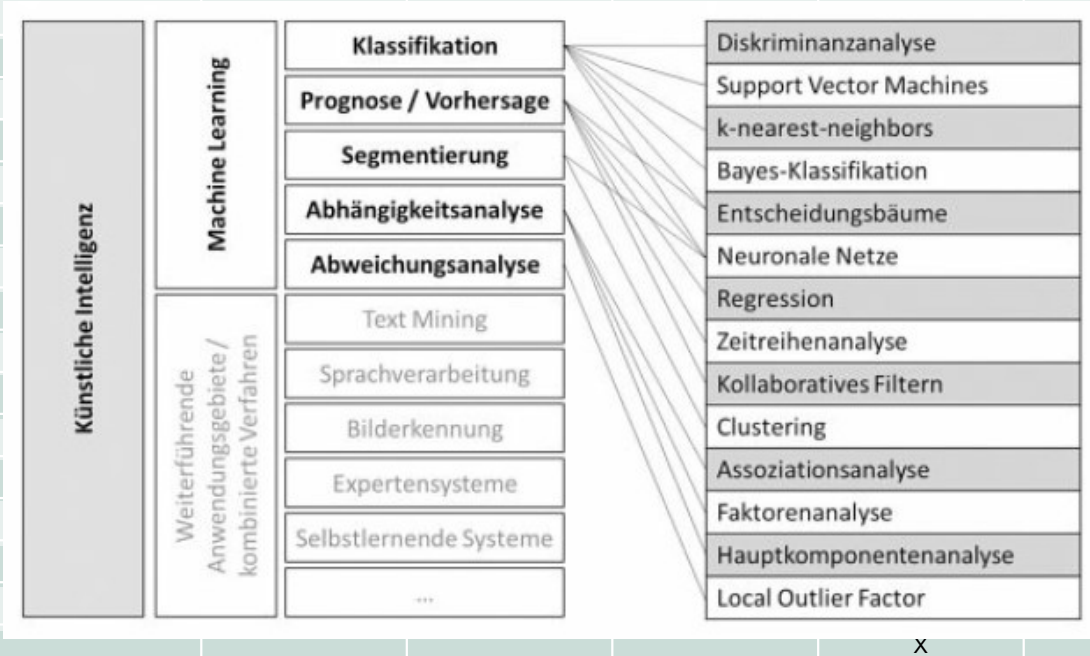
Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Machinelles Lernen

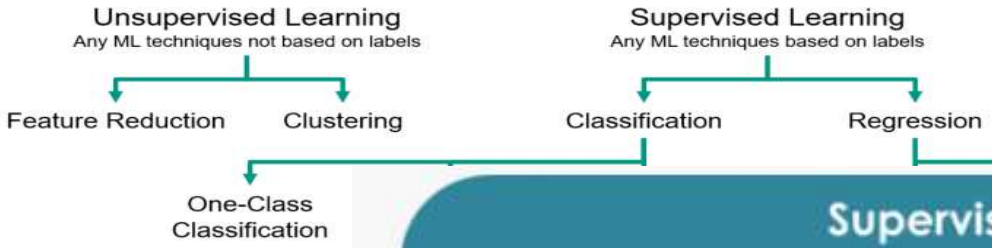


Algorithm	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X							
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM								
Isolation Forest									
	LODA								
(künstliche) Neuronale Netze	NN								X
Support Vector Machine	SVM								
Decision Tree									
Bayes-Klassifikation									
Random Forest									
Diskriminanzanalyse									
Logistic Regression									
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Andere Darstellungsformen

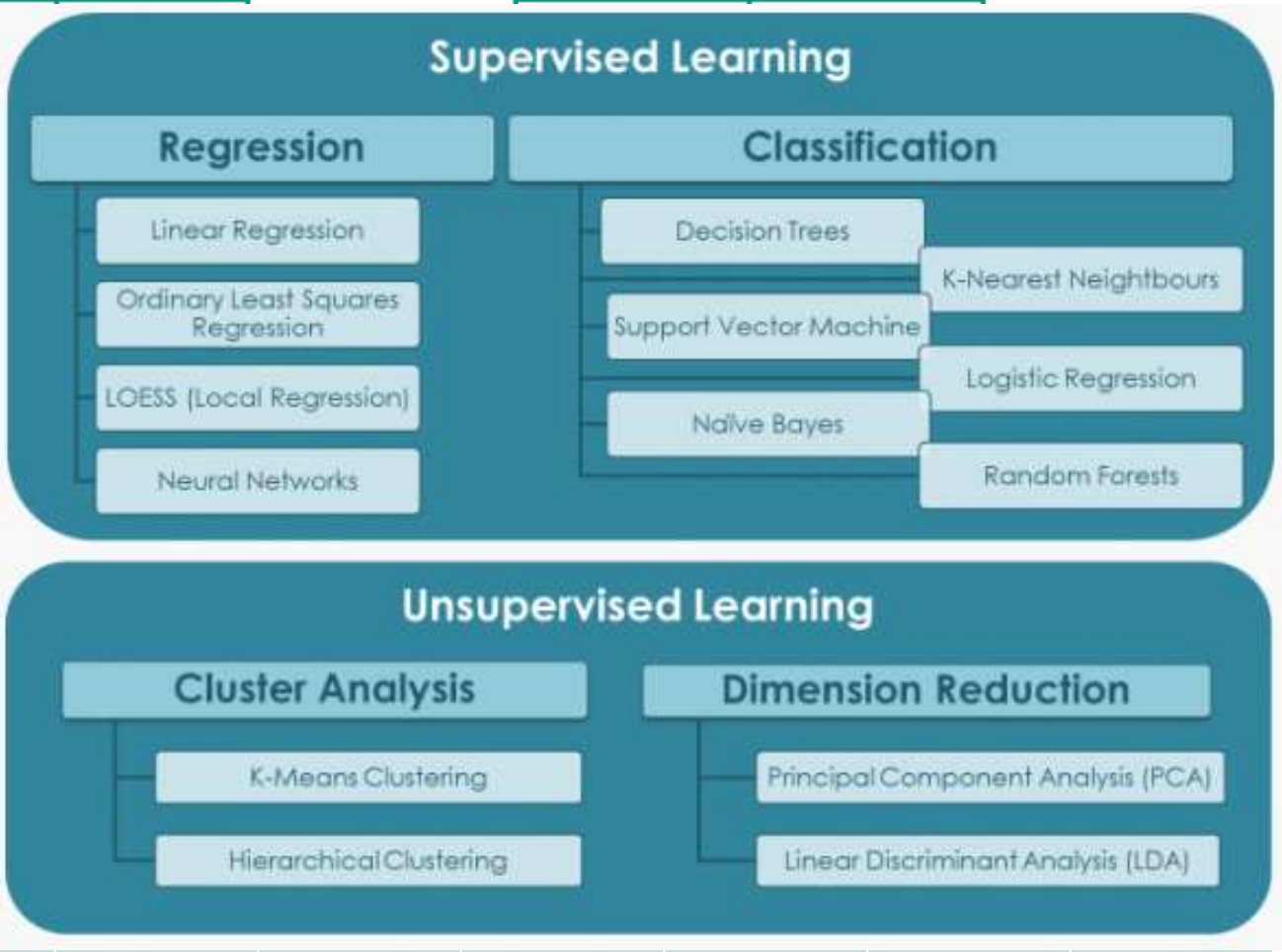


Machinelles Lernen

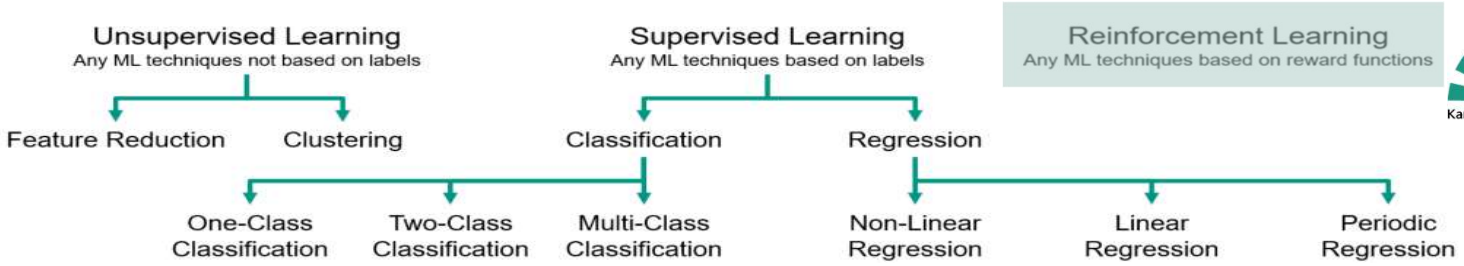


Algorithm	Abk.	Feature Reduction	Cluste
Faktorenanalyse		X	
Principal Component Analysis	PCA	X	
	K-means		X
hierarchical cluster analysis	HCA		X
DBSCAN			X
One Class Support Vector Machine	OCSVM		
Isolation Forest			
	LODA		
(künstliche) Neuronale Netze	NN		
Support Vector Machine	SVM		
Decision Tree			
Bayes-Klassifikation			
Random Forest			
Diskriminanzanalyse			
Logistic Regression			
Linear Regression			
Harmonic Regression			
Nächste-Nachbar-Klassifikation	K-NN	x	x

Andere Darstellungsformen



Machinelles Lernen



Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Reinforcement Learning („Bestärkendes Lernen“)

Vorstufen

- Beim Teach-In-Verfahren (kurz: Teachen) fährt der Programmierer den Roboter mit einer Steuerkonsole in die gewünschte Position.
  - Alle so erreichten Koordinaten (Punkte) werden in der Steuerung gespeichert.
  - Dieser Schritt wird solange wiederholt, bis der gesamte Arbeitszyklus einmal durchlaufen ist.
  - Der Programmablauf besteht darin, dass der Roboter alle gespeicherten Punkte autonom anfährt.
  - Für die Bewegung zwischen den einzelnen Punkten können Parameter eingegeben werden.
  - So sind die Geschwindigkeit und die Beschleunigung einstellbar, bei einigen Robotern ist auch eine Angabe der notwendigen Genauigkeit möglich.

Reinforcement Learning
Vorstufen

- Man unterscheidet bei der Fahrt des Roboters grundsätzlich zwei verschiedene Varianten:
  - P2P: Point-to-Point: Der Roboter fährt von Punkt n nach Punkt n+1, wobei der geometrisch günstigste Weg gewählt wird (s. Folie zuvor).
    - ► Diese Methode ist schnell, führt aber zu einem unbekannten Pfad zwischen den Punkten. Diese Fahrt ist in aller Regel keine Gerade.
  - CP: Continuous Path: Hier wird der Roboter angewiesen, nicht nur von einem Punkt zum anderen zu fahren, sondern auch noch einem vorgegebenen Pfad zu folgen.
    - ► Diese Fahrt ist i. d. R. langsamer als P2P, dafür wird aber ein bestimmter Pfad nur innerhalb geringer Toleranzen verlassen.
    - ► Diese Methode eignet sich für komplexe Fügevorgänge, bei denen es weniger auf Geschwindigkeit als auf Genauigkeit (z. B. wegen Hindernissen) ankommt.

Reinforcement Learning
Vorstufen

- Man unterscheidet bei der Fahrt des Roboters grundsätzlich zwei verschiedene Varianten:
  - P2P: Point-to-Point: Der Roboter fährt von Punkt n nach Punkt n+1, wobei der geometrisch günstigste Weg gewählt wird (s. Folie zuvor).
    - ► Diese Methode ist schnell, führt aber zu einem unbekannten Pfad zwischen den Punkten. Diese Fahrt ist in aller Regel keine Gerade.
  - CP: Continuous Path: Hier wird der Roboter angewiesen, nicht nur von einem Punkt zum anderen zu fahren, sondern auch noch einem vorgegebenen Pfad zu folgen.
    - ► Diese Fahrt ist i. d. R. langsamer als P2P, dafür wird aber ein bestimmter Pfad nur innerhalb geringer Toleranzen verlassen.
    - ► Diese Methode eignet sich für komplexe Fügevorgänge, bei denen es weniger auf Geschwindigkeit als auf Genauigkeit (z. B. wegen Hindernissen) ankommt.
- Playback-Verfahren
  - Der Programmierer fährt durch direktes Führen des Roboterarms die vorgesehene Bahn ab.
    - ► Der Roboter wiederholt genau diese Bewegungen.
    - ► Diese Methode wird häufig bei Lackierrobotern eingesetzt.
  - Playback-Verfahren mit 3D-Messarmen (mobile Koordinatenmeßsysteme KMG).
    - ► Während des Führens des KMG entlang der Bauteilkontur werden Koordinaten aufgezeichnet und später in ein Roboterprogramm konvertiert.

Reinforcement Learning
Vorstufen

- Playback-Verfahren
  - Der Programmierer fährt durch direktes Führen des Roboterarms die vorgesehene Bahn ab.
    - ► Der Roboter wiederholt genau diese Bewegungen.
    - ► Diese Methode wird häufig bei Lackierrobotern eingesetzt.
  - Playback-Verfahren mit 3D-Messarmen (mobile Koordinatenmeßsysteme KMG).
    - ► Während des Führens des KMG entlang der Bauteilkontur werden Koordinaten aufgezeichnet und später in ein Roboterprogramm konvertiert.

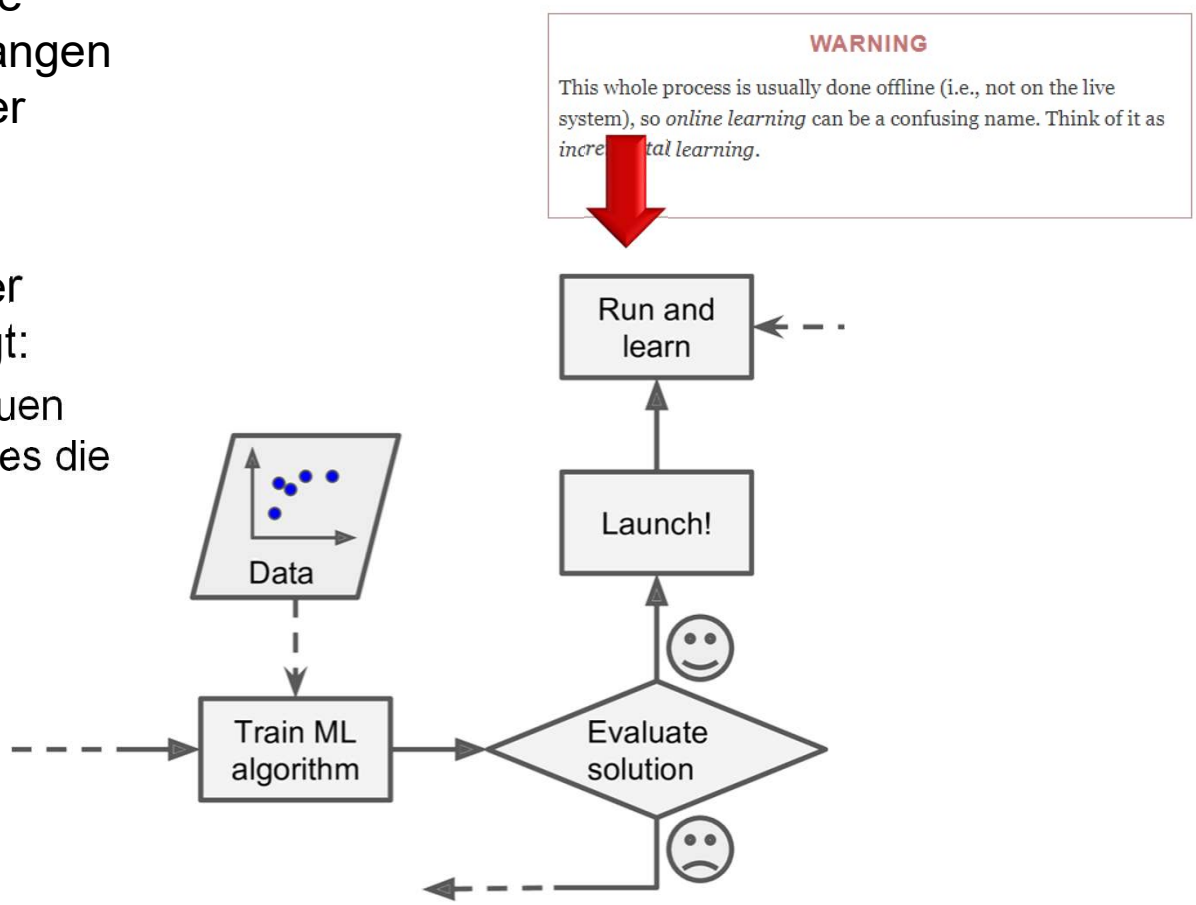
MiWuLa …

- „Automatisierter Schiffsverkehr“ …
  - … findet ohne menschliche Interaktion statt
  - … klinkt sich in die Automatisierungsphilosophie des MiWuLa ein
- Ausblick
  - Schiffsignale passend zum Manöver
  - Mehrere Schiffe mit Kollisionsschutz
    - Über „Karte“ oder „on board“
  - „Längsseits gehen“ von Polizeiboot, Lotse, Proviant etc.

Miniatur Wunderland
www.miniatur-wunderland.de

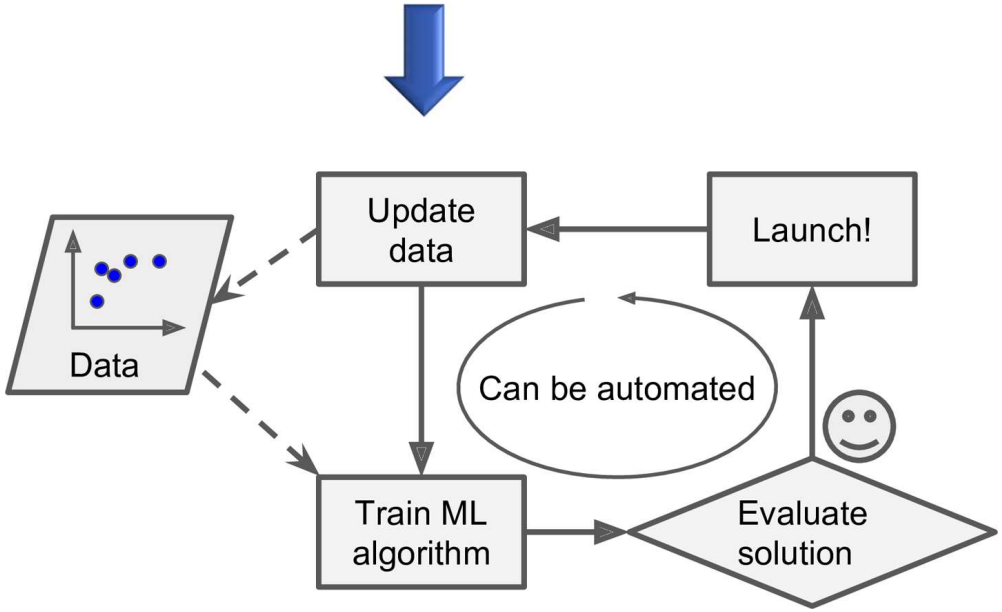
Incremental (online) Learning

- Online-Lernen ist ideal für Systeme, die Daten als kontinuierlichen Fluss empfangen (z.B. Aktienkurse) und sich schnell oder autonom an Veränderungen anpassen müssen.
- Es ist eine gute Option, wenn man über begrenzte Computerressourcen verfügt:
  - Sobald ein Online-Lernsystem von neuen Dateninstanzen erfahren hat, ergänzt es die bestehenden.



Batch (offline) Learning

- Beim Batch-Lernen ist das System nicht inkrementell lernfähig:
  - Es muss mit allen verfügbaren Daten trainiert werden.
  - Dies nimmt in der Regel viel Zeit und Rechenressourcen in Anspruch, so dass dies offline geschieht.
  - Zuerst wird das System trainiert, dann wird es in die Praxis eingeführt und läuft ohne Lernen weiter.
  - Es wendet an, was es gelernt hat.
  - Dies wird als Offline-Lernen bezeichnet.
- Wenn man möchte, dass ein Batch-Lernsystem über neue Daten (z.B. eine neue Art von Spam) informiert wird, muss man eine neue Version des Systems von Grund auf den gesamten Datensatz trainieren (nicht nur die neuen Daten, sondern auch die alten Daten).



Reinforcement Learning

„Bestärkendes“ Lernen

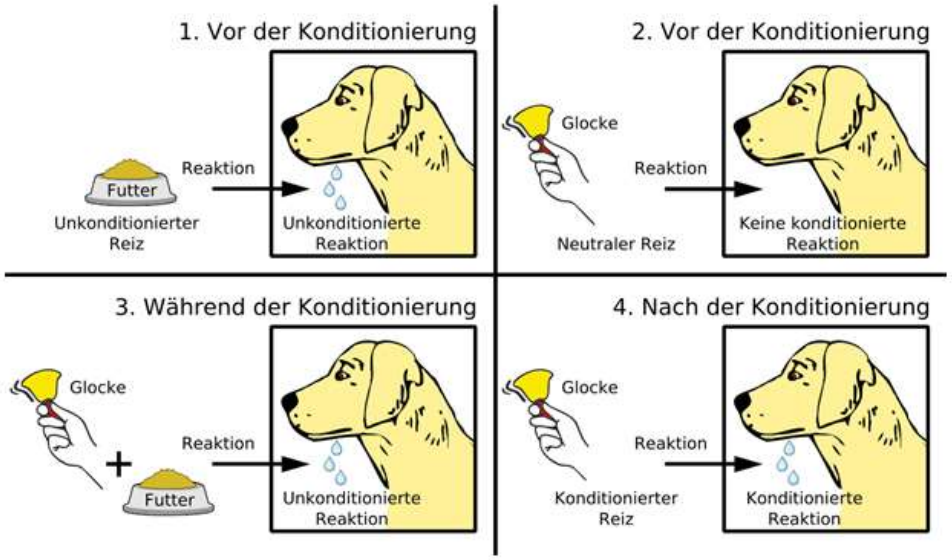
- Ein lernendes System (Agent) kann die Umgebung beobachten, Aktionen auswählen und durchführen und dafür Belohnungen erhalten (Reward).
  - Der Agent wählt situativ eine Aktion aus einer Menge von verfügbaren Aktionen, wodurch er in einen Folgezustand gelangt und eine Belohnung erhält.
  - Ziel des Agenten ist es somit, den zukünftig erwarteten Gewinn zu maximieren.
- Es muss dann von selbst lernen, was die beste Strategie, eine so genannte Politik (Policy) ist, um im Laufe der Zeit die meiste Belohnung zu erhalten.
- Eine Richtlinie definiert, welche Aktion der Agent wählen soll, wenn er sich in einer bestimmten Situation befindet.

- Remark: *Unterschied Zustand - Situation*
  - *Ein Zustand eines Systems ist die Abspeicherung aller Variablen, die das System hat.*
    - *Aus diesen Variablen lässt sich der Zustand komplett rekonstruieren.*
  - *Bei den meisten Systemen kommen wir jedoch an alle Variablen gar nicht ran, weshalb die unvollständige Variablenmenge nur eine Situation ist.*

Lernen

Bedingter Reflex nach Pawlow

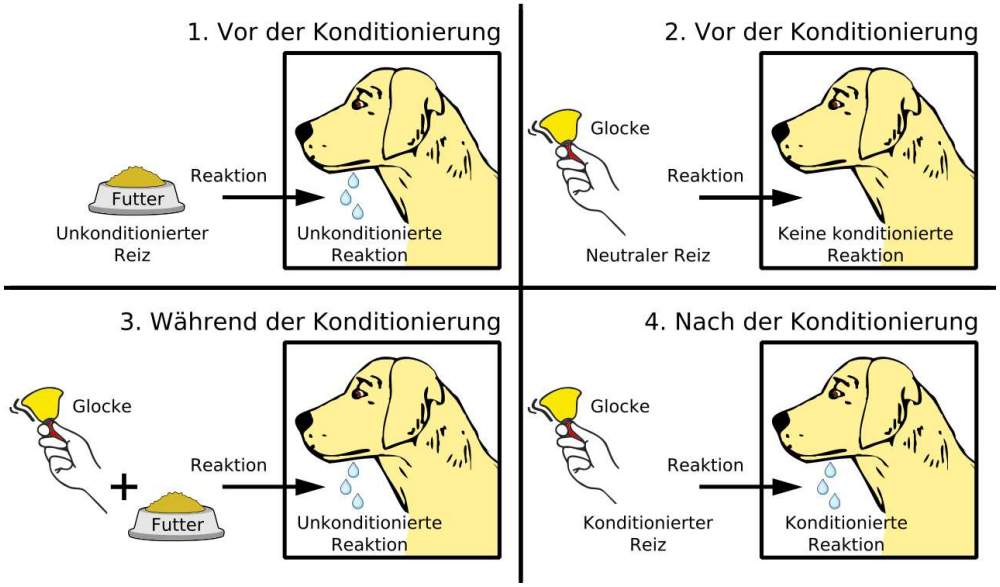
- Um diese Hypothese zu prüfen, gestaltete er 1905[1] ein aussagekräftiges Experiment:
  - Auf die Darbietung von Futter, einem unbedingten Reiz, folgt Speichelfluss (unbedingte Reaktion), auf das Ertönen eines Glockentons (neutraler Reiz) nichts.
  - Wenn aber der Glockenton wiederholt in engem zeitlichem Zusammenhang mit dem Anbieten von Futter erklingt, reagieren die Hunde schließlich auf den Ton allein mit Speichelfluss.
  - Dieses Phänomen bezeichnete Pawlow als Konditionierung.



- **Klassische Konditionierung** ist eine von dem russischen Physiologen Iwan Petrowitsch Pawlow begründete behavioristische Lerntheorie, die besagt, dass einer natürlichen, meist angeborenen, sogenannten *unbedingten* Reaktion durch Lernen eine neue, *bedingte* Reaktion hinzugefügt werden kann.

Reinforcement Learning
„Konditionierung“

- Die Begriff Reinforcement wurde im Jahre 1903 von Pavlov eingeführt, um die Stärkung der Assoziation zwischen einer unbedingten und einen konditionierten Reiz, der sich ergibt, wenn die beiden zusammen dargestellt werden, zu beschreiben.
- Bei der Konditionierung bezeichnet man ein Ereignis, das die Wahrscheinlichkeit erhöht, dass ein bestimmtes Verhalten gezeigt wird, als Verstärkung.



Reinforcement Learning

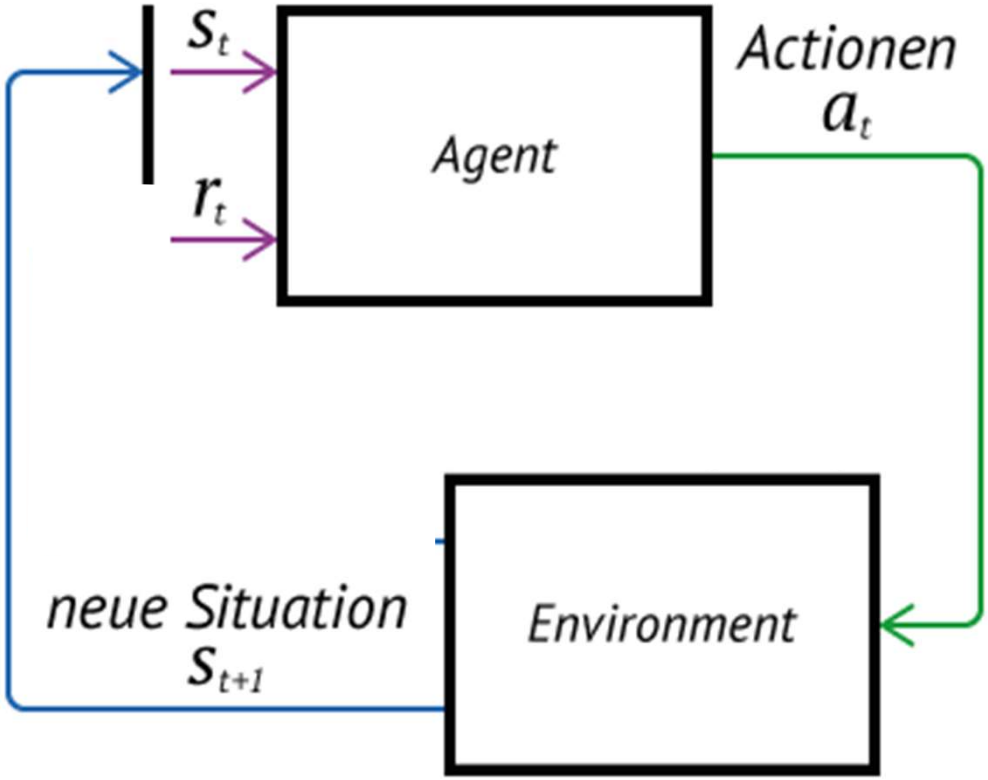
„Bestärkendes“ Lernen

- Reinforcement Learning wird auf einen Agenten angewandt, der auf ein dynamisches Environment (System) Einfluss ausübt und in diesem Environment bestimmte Aufgaben ausüben muss.

Quelle: http://www.informatikseite.de/neuro/node67.php

Reinforcement Learning

„Bestärkendes“ Lernen

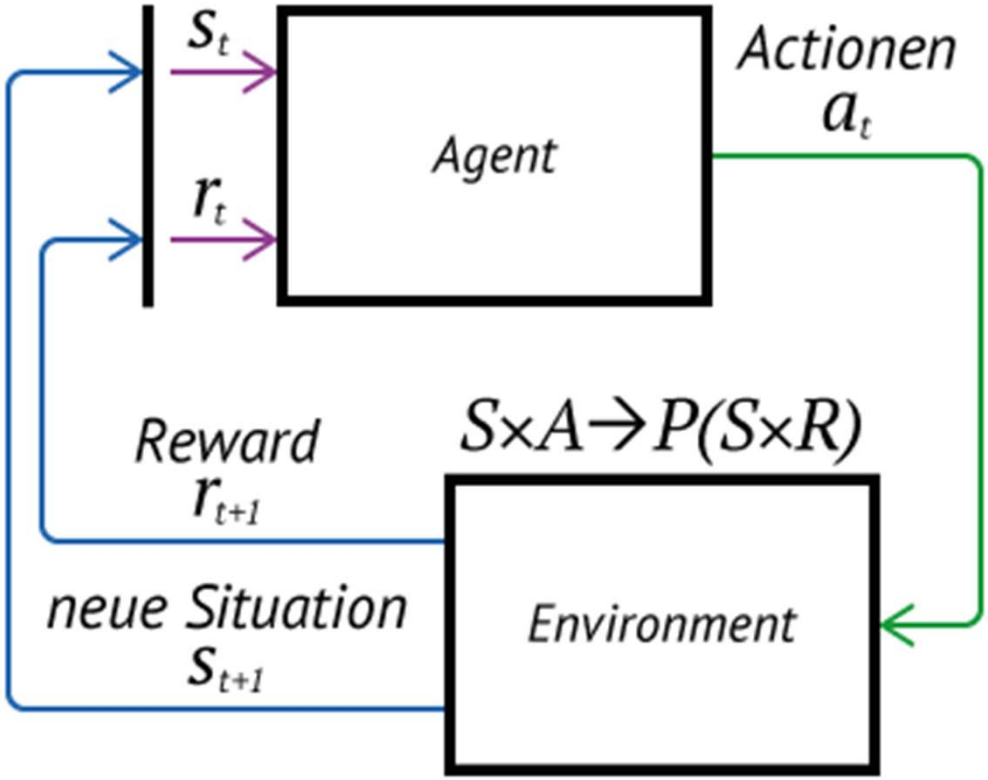


- Reinforcement Learning wird auf einen Agenten angewandt, der auf ein dynamisches Environment (System) Einfluss ausübt und in diesem Environment bestimmte Aufgaben ausüben muss.
- Der Agent bildet also die momentane Situation s $\in$ S des Environments auf eine Aktion a $\in$ A ab.

Quelle: http://www.informatikseite.de/neuro/node67.php

Reinforcement Learning

„Bestärkendes“ Lernen



- Reinforcement Learning wird auf einen Agenten angewandt, der auf ein dynamisches Environment (System) Einfluss ausübt und in diesem Environment bestimmte Aufgaben ausüben muss.
- Der Agent bildet also die momentane Situation s $\in$ S des Environments auf eine Aktion a $\in$ A ab.
- Für jede Aktion bekommt der Agent einen Reward r $\in$ R, der seine Aktion bewertet, aber nicht sagt, was er falsch gemacht hat oder *was* er gut gemacht hat.
- Beim Reinforcement Learning ist es das Ziel einen möglichst hohen Reward über eine lange Zeit zu erwirtschaften.

Quelle: http://www.informatikseite.de/neuro/node67.php

<weiterführende Information>

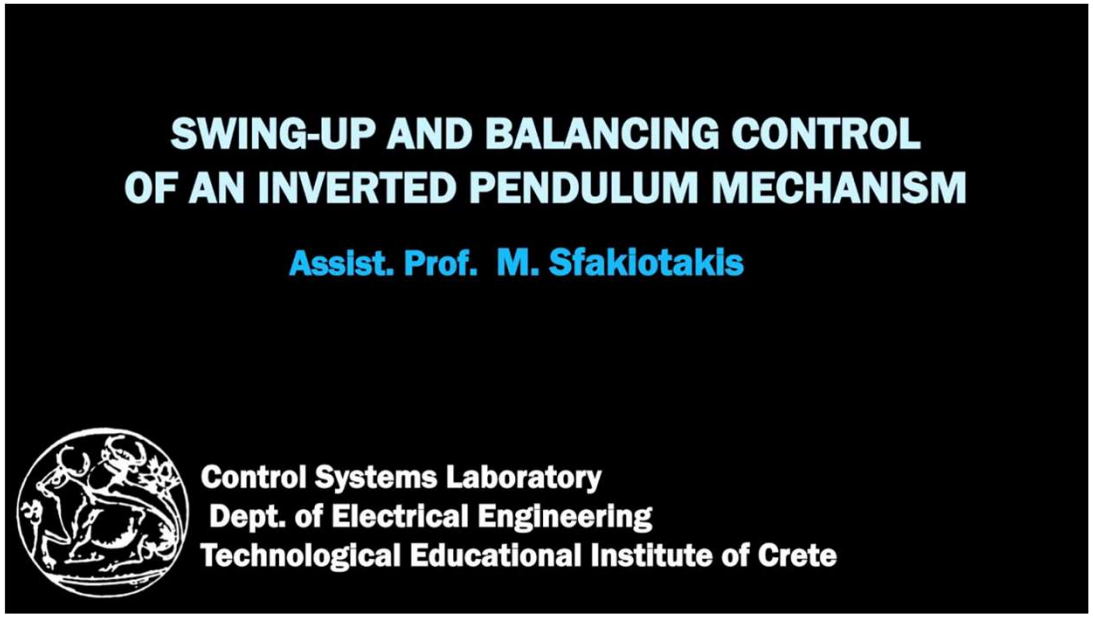
Artificial Intelligence: What's The Difference Between Deep Learning And Reinforcement Learning?

- The various cutting-edge technologies that are under the umbrella of artificial intelligence are getting a lot of attention lately. As the amount of data we generate continues to grow to mind-boggling levels, our AI maturity and the potential problems AI can help solve grows right along with it. This data and the amazing computing power that's now available for a reasonable cost is what fuels the tremendous growth in AI technologies and makes deep learning and reinforcement learning possible. With the rapid changes in the AI industry, it can be challenging to keep up with the latest cutting-edge technologies.
- Both deep learning and reinforcement learning are machine learning functions, which in turn are part of a wider set of artificial intelligence tools. What makes deep learning and reinforcement learning functions interesting is they enable a computer to develop rules on its own to solve problems. This ability to learn is nothing new for computers – but until recently we didn't have the data or computing power to make it an everyday tool.
- What is deep learning?
  - Deep learning is essentially an autonomous, self-teaching system in which you use existing data to train algorithms to find patterns and then use that to make predictions about new data. For example, you might train a deep learning algorithm to recognize cats on a photograph. You would do that by feeding it millions of images that either contains cats or not. The program will then establish patterns by classifying and clustering the image data (e.g. edges, shapes, colors, distances between the shapes, etc.). Those patterns will then inform a predictive model that is able to look at a new set of images and predict whether they contain cats or not, based on the model it has created using the training data.
  - Deep learning algorithms do this via various layers of artificial neural networks which mimic the network of neurons in our brain. This allows the algorithm to perform various cycles to narrow down patterns and improve the predictions with each cycle.
  - A great example of deep learning in practice is Apple's Face ID. When setting up your phone you train the algorithm by scanning your face. Each time you log on using e.g. Face ID, the TrueDepth camera captures thousands of data points which create a depth map of your face and the phone's inbuilt neural engine will perform the analysis to predict whether it is you or not.
- What is reinforcement learning?
  - Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error. It performs actions with the aim of maximizing rewards, or in other words, it is learning by doing in order to achieve the best outcomes. This is similar to how we learn things like riding a bike where in the beginning we fall off a lot and make too heavy and often erratic moves, but over time we use the feedback of what worked and what didn't to fine-tune our actions and learn how to ride a bike. The same is true when computers use reinforcement learning, they try different actions, learn from the feedback whether that action delivered a better result, and then reinforce the actions that worked, i.e. reworking and modifying its algorithms autonomously over many iterations until it makes decisions that deliver the best result.
  - A good example of using reinforcement learning is a robot learning how to walk. The robot first tries a large step forward and falls. The outcome of a fall with that big step is a data point the reinforcement learning system responds to. Since the feedback was negative, a fall, the system adjusts the action to try a smaller step. The robot is able to move forward. This is an example of reinforcement learning in action.
  - One of the most fascinating examples of reinforcement learning in action I have seen was when Google's Deep Mind applied the tool to classic Atari computer games such as Break Out. The goal (or reward) was to maximize the score and the actions were to move the bar at the bottom of the screen to bounce the playing ball back up to break the bricks at the top of the screen. You can watch the video here which shows how, in the beginning, the algorithm is making lots of mistakes but quickly improves to a stage where it would beat even the best human players.
- Difference between deep learning and reinforcement learning
  - Deep learning and reinforcement learning are both systems that learn autonomously. The difference between them is that deep learning is learning from a training set and then applying that learning to a new data set, while reinforcement learning is dynamically learning by adjusting actions based in continuous feedback to maximize a reward.
  - Deep learning and reinforcement learning aren't mutually exclusive. In fact, you might use deep learning in a reinforcement learning system, which is referred to as deep reinforcement learning and will be a topic I cover in another post.
- Bernard Marr Contributor, Artificial Intelligence in Practice: How 50 Companies Used AI & Machine Learning,

„Polebalancer“

Ziel das Pole Balancers ist es, den Stab gerade zu halten.

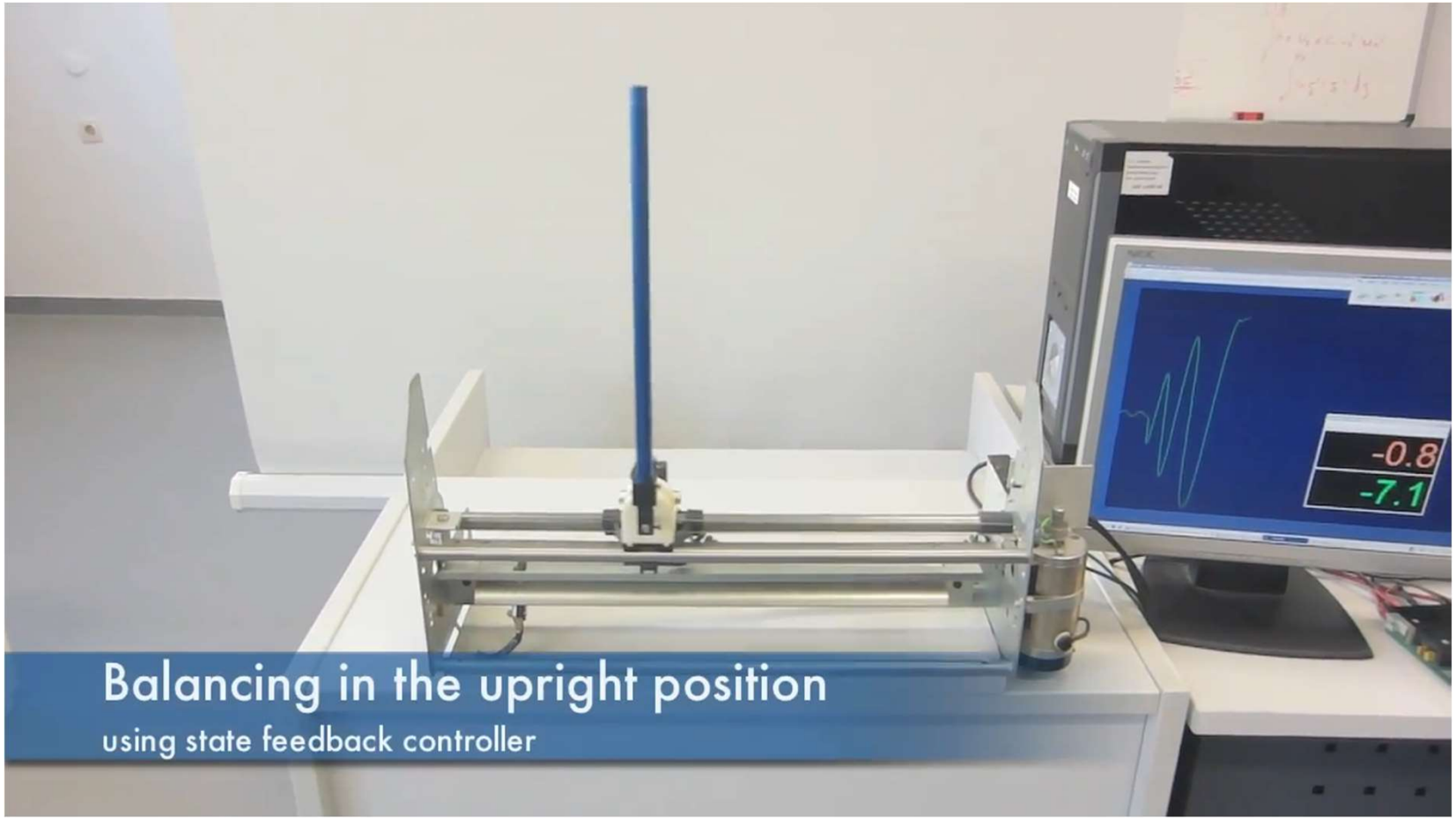
- Reinforcment Learning mit einem „pure negativ reward“



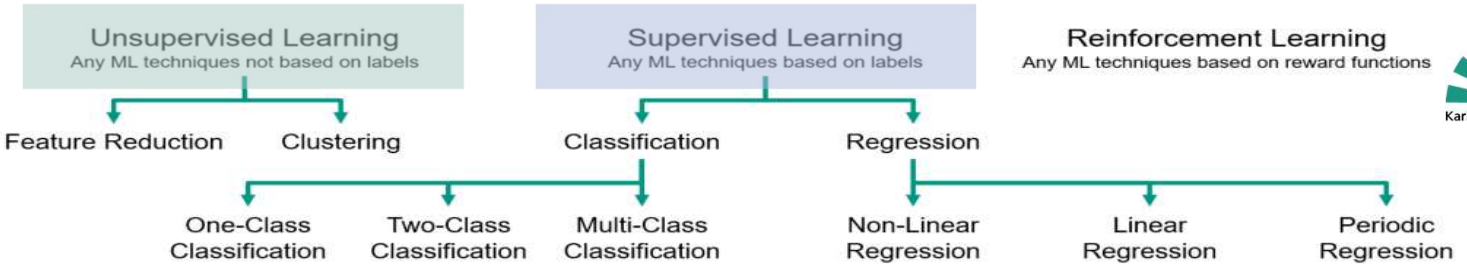
- Der Polebalancer besteht aus einem Wagen, der nach links und rechts per Bang-Bang-Control beschleunigt werden kann. Dabei muss er immer in eine Richtung beschleunigt werden. Er kann nicht einfach stehen gelassen werden.
- Der Wagen hat eine Position auf der Schiene mit einer Geschwindigkeit. .
- An den Seiten der Schiene, auf der der Wagen steht, befinden sich Begrenzungen.
- Auf dem Wagen befindet sich ein Pole an einem Gelenk. Dieser hat einen Winkel zur Senkrechten und eine Geschwindigkeit .
- Ziel das Pole Balancers ist es, den Stab gerade zu halten. Wir können dies mit Reinforcment Learning mit einem pure negativ reward tun.

„Polebalancer“

Ziel das Pole Balancers ist es, den Stab gerade zu halten.



Machinelles Lernen



Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Eine Anmerkung zu den Begriffen

Überwachtes und unüberwachtes Lernen

- Im übertragenen Sinn »überwacht« ein Lehrer den Lernenden, indem er ihm vorsichtige Hinweise auf das Lernziel sowie einige Beispiele gibt.
- Beim »unüberwachten« Lernen könnten die selben Beispiele vorhanden sein, es gibt jedoch keine Hinweise auf das Lernziel. Der Lernende wird nicht über den Zweck des Lernens informiert, sondern muss seine eigenen Schlüsse ziehen, was die Beispiele gemeinsam haben (eine Art Autodidakt)

- Der Unterschied zwischen den beiden Fragen ist zwar nicht groß, aber bedeutsam.
  - Wenn ein Ziel angegeben ist, kann die Aufgabe als überwachter Task formuliert werden.
  - Überwachte Tasks erfordern andere Verfahren als unüberwachte und liefern oft sehr viel nützlichere Ergebnisse.
  - Einem überwachten Verfahren wird ein bestimmter Zweck der Gruppierung vorgegeben – die Vorhersage des Ziels.

Überwachtes vs. Unüberwachtes Lernen

- Beim überwachten Lernen liegen Daten vor, die schon ein Ergebnis oder „Label“ enthalten.
- Beim überwachten Lernen wird die vom Netz erzeugte Ausgabe betrachtet und ihre Abweichung von der gewünschten Ausgabe gemessen.
  - Danach werden die Gewichte entsprechend der Größe der Abweichung angepasst.
- Remark: *Das bestärkende Lernen (Reinforcement Learning) kann als überwachtes Lernen ohne Informationen über die Höhe der Abweichung vom gewünschten Wert interpretiert werden.*

Überwachtes vs. Unüberwachtes Lernen

- Beim überwachten Lernen liegen Daten vor, die schon ein Ergebnis oder „Label“ enthalten.
- Beim überwachten Lernen wird die vom Netz erzeugte Ausgabe betrachtet und ihre Abweichung von der gewünschten Ausgabe gemessen.
  - Danach werden die Gewichte entsprechend der Größe der Abweichung angepasst.
- Remark: *Das bestärkende Lernen (Reinforcement Learning) kann als überwachtes Lernen ohne Informationen über die Höhe der Abweichung vom gewünschten Wert interpretiert werden.*

- Das unüberwachte Lernen setzt keinen Lehrer voraus, d.h. für die Eingabewerte der Trainingsdaten ist die richtige Ausgabe nicht bekannt.
- Das Netz muss sich selber organisieren, d. h. die Gewichtungen anpassen, um Muster zu erkennen und z.B. Gruppierungen (Cluster) der Datensätze vorzunehmen.
- Beim unüberwachten Lernen handelt es sich um das allgemeine Verstehen der vorliegenden Daten und die Entdeckung versteckter Strukturen.
  - ► Die Daten sind also nicht durch „Labels“ gekennzeichnet.

Überwachtes vs. Unüberwachtes Lernen

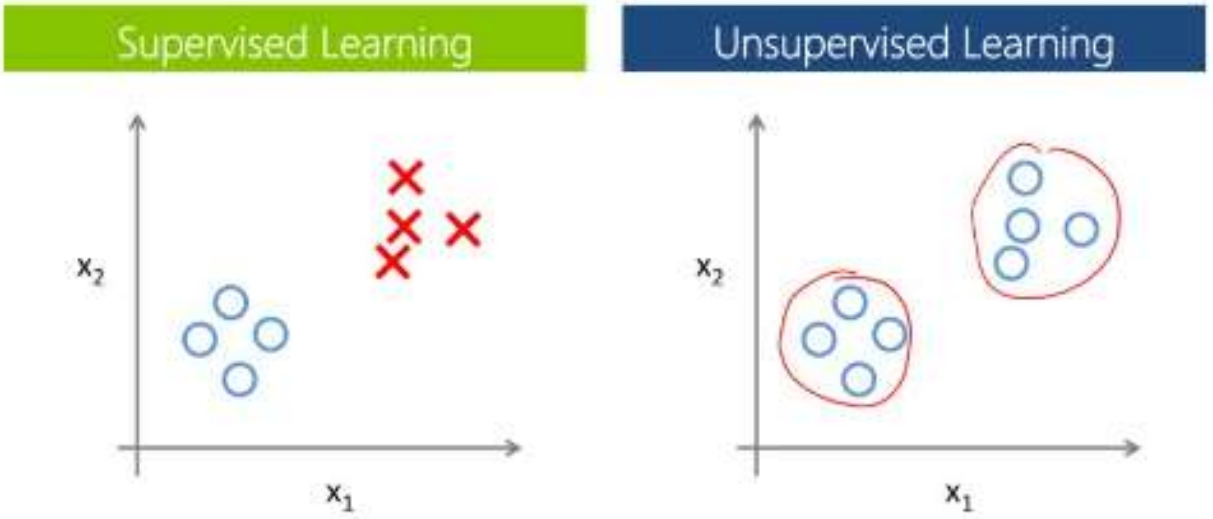
- Beim überwachten Lernen liegen Daten vor, die schon ein Ergebnis oder „Label“ enthalten.
- Beim überwachten Lernen wird die vom Netz erzeugte Ausgabe betrachtet und ihre Abweichung von der gewünschten Ausgabe gemessen.
  - Danach werden die Gewichte entsprechend der Größe der Abweichung angepasst.
- Remark: *Das bestärkende Lernen (Reinforcement Learning) kann als überwachtes Lernen ohne Informationen über die Höhe der Abweichung vom gewünschten Wert interpretiert werden.*

- Das unüberwachte Lernen setzt keinen Lehrer voraus, d.h. für die Eingabewerte der Trainingsdaten ist die richtige Ausgabe nicht bekannt.
- Das Netz muss sich selber organisieren, d. h. die Gewichtungen anpassen, um Muster zu erkennen und z.B. Gruppierungen (Cluster) der Datensätze vorzunehmen.
- Beim unüberwachten Lernen handelt es sich um das allgemeine Verstehen der vorliegenden Daten und die Entdeckung versteckter Strukturen.
  - ► Die Daten sind also nicht durch „Labels“ gekennzeichnet.

- Die Frage nach überwachtem oder unüberwachtem Lernen ist also weniger „dramatisch“ als es die Begriffe vermuten lassen.
- Es geht nicht um eine echte „Überwachung“ des Lernprozesses, sondern lediglich darum, ob die Daten ge-labelt sind.

Überwachtes vs. Unüberwachtes Lernen

	Deutsch	Englisch	Beschreibung
Teacher	Überwachtes Lernen	Supervised	Es gibt einen desired Output, welchen das Netz lernen soll.
Reward	Bestärkendes Lernen	Reinforcment Learning	Es gibt nur eine Belohnung, die sagt, ob die Aktion gut oder schlecht war.
-	Unüberwachtes Lernen	Unsupervised	Es gibt gar keinen Output mehr. Wir können die Muster nur zu Clustern zusammenfassen.



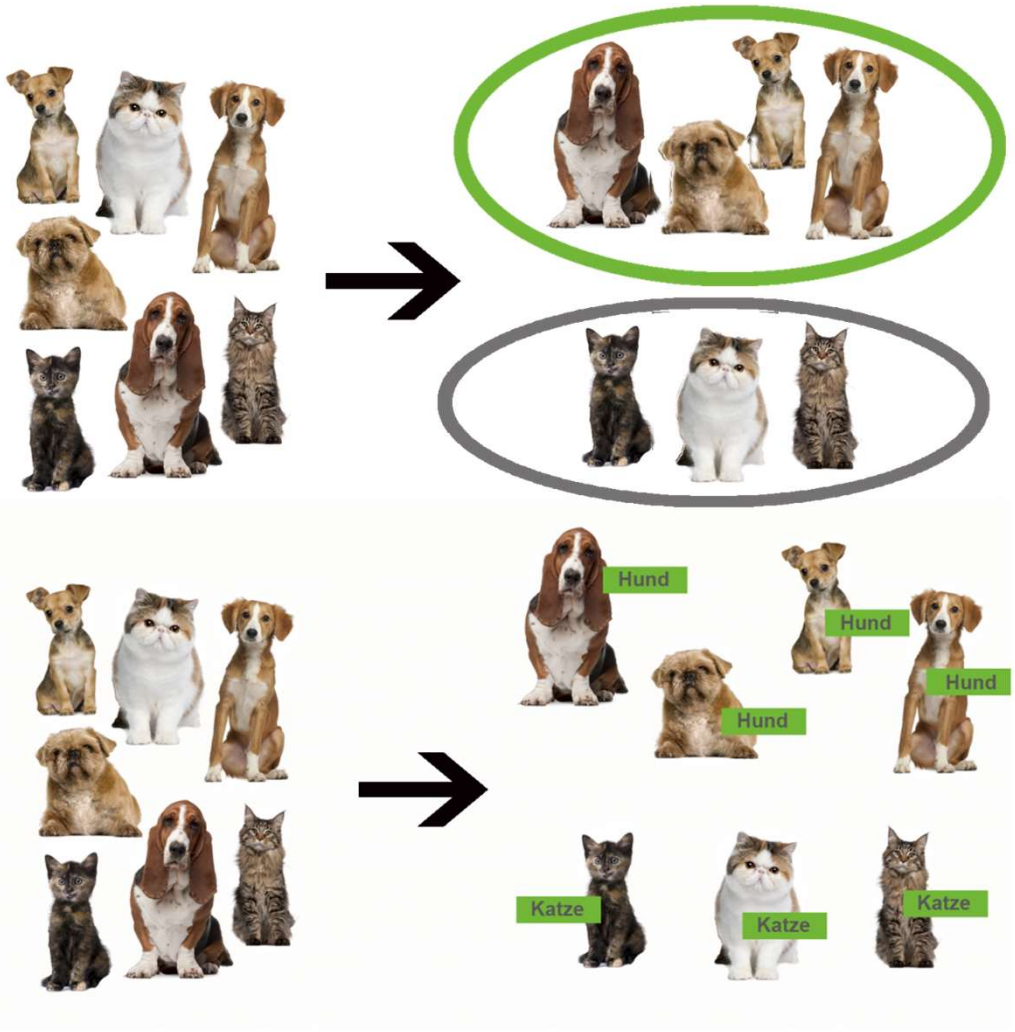
Überwachtes vs. Unüberwachtes Lernen

	Deutsch	Englisch	Beschreibung
Teacher	Überwachtes Lernen	Supervised	Es gibt einen desired Output, welchen das Netz lernen soll.
Reward	Bestärkendes Lernen	Reinforcment Learning	Es gibt nur eine Belohnung, die sagt, ob die Aktion gut oder schlecht war.
-	Unüberwachtes Lernen	Unsupervised	Es gibt gar keinen Output mehr. Wir können die Muster nur zu Clustern zusammenfassen.

Überwachtes vs. Unüberwachtes Lernen

Beispiel: Hunde und Katzen

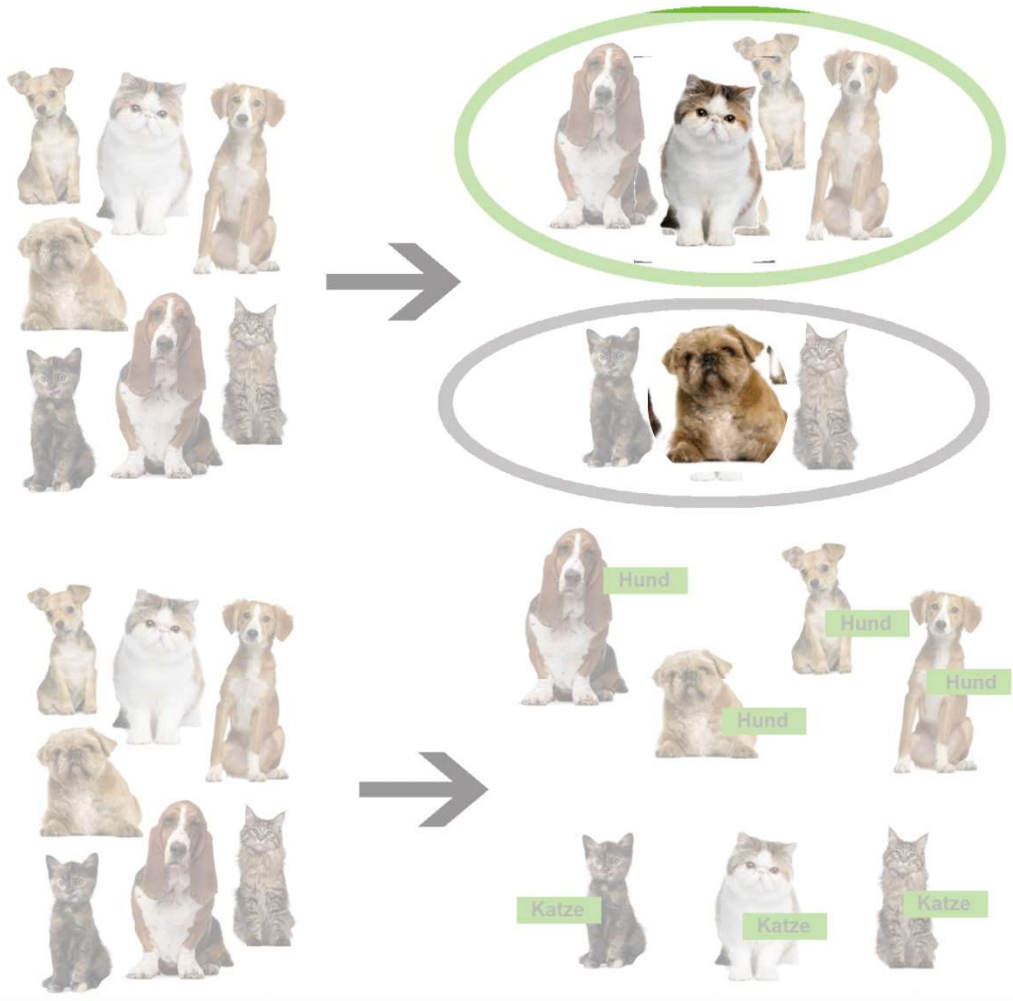
- Unüberwachtes Lernen:
  - Das System weiß nicht, was es erkennen soll.
  - Wenn es Bilder von Tieren verarbeitet, teilt das System alles was aussieht wie eine Katze oder alles was aussieht wie ein Hund in entsprechende Gruppen ein, ohne diese jedoch so zu benennen, da nicht definiert ist, was eine Katze und was ein Hund ist.
  - Diese Methode wird angewandt, wenn Daten noch unbekannt sind und entsprechend keine Vorgaben vorhanden sind.
- Überwachtes Lernen:
  - Es gibt Trainings-Daten, bei denen wir die Eingangs-Parameter sowie das Ergebnis kennen.
  - Aus den Trainings-Daten werden Modelle von Hunden und Katzen erstellt, die zusammen mit den Machine Learning Algorithmen das Ergebnis liefern.
  - Da die Modelle erstellt wurden, kann man „unbekannte“ Bilder liefern und das System berechnet dafür das Ergebnis (Prediction).



Überwachtes vs. Unüberwachtes Lernen

Beispiel: Hunde und Katzen

- Unüberwachtes Lernen:
  - Das System weiß nicht, was es erkennen soll.
  - Wenn es Bilder von Tieren verarbeitet, teilt das System alles was aussieht wie eine Katze oder alles was aussieht wie ein Hund in entsprechende Gruppen ein, ohne diese jedoch so zu benennen, da nicht definiert ist, was eine Katze und was ein Hund ist.
  - Diese Methode wird angewandt, wenn Daten noch unbekannt sind und entsprechend keine Vorgaben vorhanden sind.
- Überwachtes Lernen:
  - Es gibt Trainings-Daten, bei denen wir die Eingangs-Parameter sowie das Ergebnis kennen.
  - Aus den Trainings-Daten werden Modelle von Hunden und Katzen erstellt, die zusammen mit den Machine Learning Algorithmen das Ergebnis liefern.
  - Da die Modelle erstellt wurden, kann man „unbekannte“ Bilder liefern und das System berechnet dafür das Ergebnis (Prediction).



<weiterführende Information>

Überwachtes vs unüberwachtes maschinelles Lernen

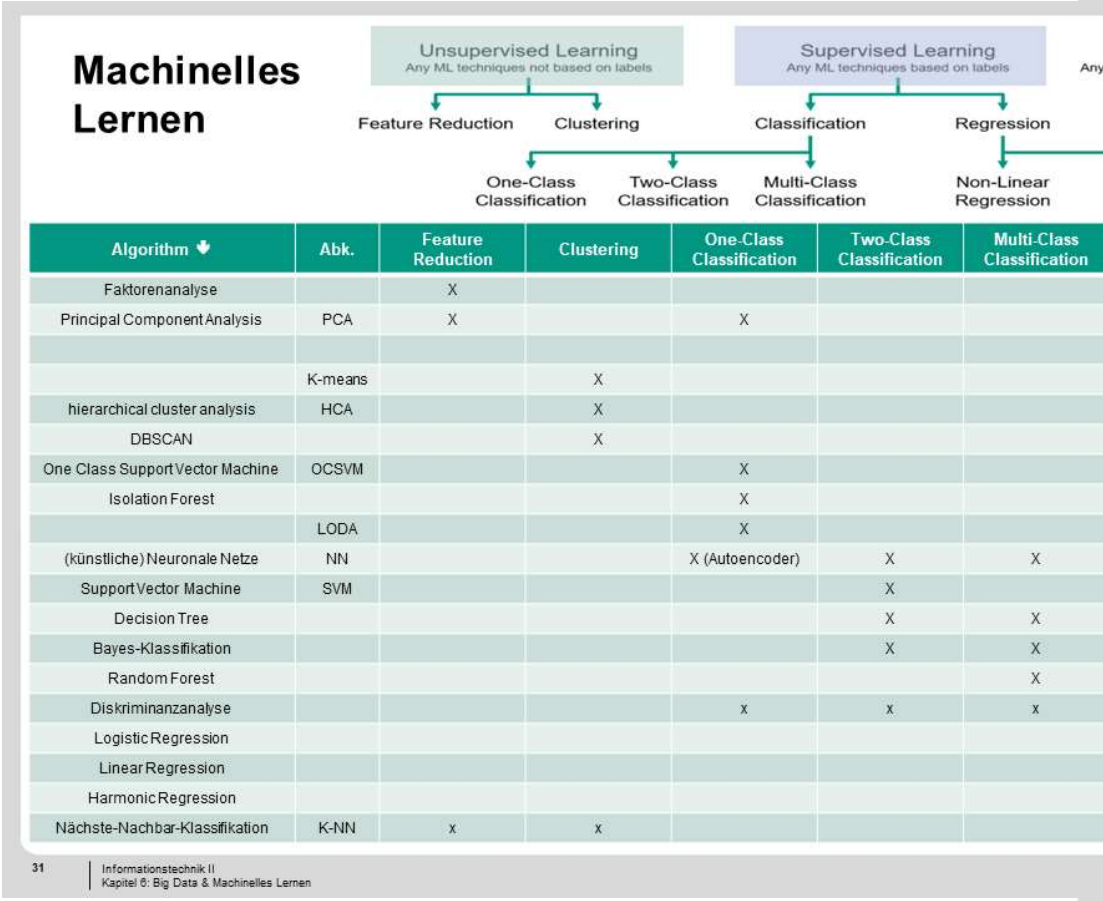
- Während überwachte Lernverfahren über eine Trainingsphase regelrecht auf ein(!) Problem abgerichtet werden und dann produktiv als Assistenzsystem (bis hin zum Automated Decision Making) funktionieren, sind unüberwachte maschinelle Lernverfahren eine Methodik, um unübersichtlich viele Zeilen und Spalten von folglich sehr großen Datenbeständen für den Menschen leichter interpretierbar machen zu können.

- Trainiere einen Algorithmus mit überwachtem maschinellen Lernen
  - Wenn ein Modell anhand von mit dem Ergebnis (z. B. Klassifikationsgruppe) gekennzeichneter Trainingsdaten erlernt werden soll, handelt es sich um überwachtes Lernen.
  - Die richtige Antwort muss während der Trainingsphase also vorliegen und der Algorithmus muss die Lücke zwischen dem Input (Eingabewerte) und dem Output (das vorgeschriebene Ergebnis) füllen.
  - Die Überwachung bezieht sich dabei nur auf die Trainingsdaten!
  - Im produktiven Lauf wird grundsätzlich nicht überwacht
    ► Das Lernen könnte sich auf neue Daten in eine ganz andere Richtung entwickeln, als dies mit den Trainingsdaten der Fall war.

„Clusterbildung“ als Repräsentant des unüberwachten Lernens

Clusterbildung hat das Ziel nicht annotierte Trainingsdaten zu gruppieren.

- Im Unterschied zur Klassifikation ist damit die Zuordnung der Trainingsdaten zu den gesuchten Klassen nicht vorhanden.
- Die Cluster, in welche die Eingangswerte gruppiert werden, sind zu Beginn unbekannt.
- Durch die nicht annotierten Trainingsdaten ist Clusterbildung eine Problemstellung für unüberwachtes Lernen und ein wichtiger Bestandteil
  bei der Wissensentdeckung in großen Datenmengen.



Algorithm	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification
Faktorenanalyse		X				
Principal Component Analysis	PCA	X		X		
	K-means		X			
hierarchical cluster analysis	HCA		X			
DBSCAN			X			
One Class Support Vector Machine	OCSVM			X		
Isolation Forest				X		
	LODA			X		
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X
Support Vector Machine	SVM				X	
Decision Tree					X	X
Bayes-Klassifikation					X	X
Random Forest						X
Diskriminanzanalyse				x	x	x
Logistic Regression						
Linear Regression						
Harmonic Regression						
Nächste-Nachbar-Klassifikation	K-NN	x	x			

31 | Informationstechnik II
Kapitel 6: Big Data & Machinelles Lernen

„Clusterbildung“ als Repräsentant des unüberwachten Lernens

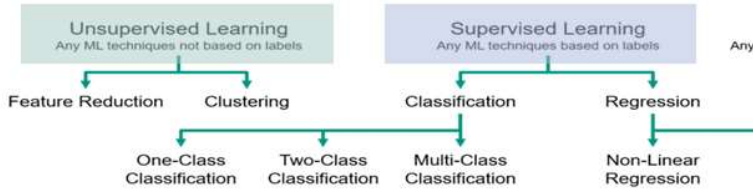
Clusterbildung hat das Ziel nicht annotierte Trainingsdaten zu gruppieren.

- Im Unterschied zur Klassifikation ist damit die Zuordnung der Trainingsdaten zu den gesuchten Klassen nicht vorhanden.
- Die Cluster, in welche die Eingangswerte gruppiert werden, sind zu Beginn unbekannt.
- Durch die nicht annotierten Trainingsdaten ist Clusterbildung eine Problemstellung für unüberwachtes Lernen und ein wichtiger Bestandteil bei der Wissensentdeckung in großen Datenmengen.

„Klassifikation“ als Repräsentant des überwachten Lernens

- Bei der Klassifikation ist die Aufgabe den gegebenen Eingangswerten einen bestimmten diskreten Ausgangswert zuzuordnen.
  - ► Damit findet eine Klassifizierung der Eingangswerte statt (s. Beispiel „Panzer am Waldrand“).
- Entsprechende Aufgaben werden meistens durch überwachte Lernverfahren adressiert.
  - ► Mittels annotierter Trainingsdaten wird ein Klassifikator gelernt, der die Eingangswerte möglichst präzise in die vorgegebenen Klassen einordnet.
  - ► Ist das Training erfolgreich abgeschlossen, klassifiziert das Computerprogramm in der Testphase auch neue, bis dahin unbekannte Datensätze richtig.

Machinelles Lernen



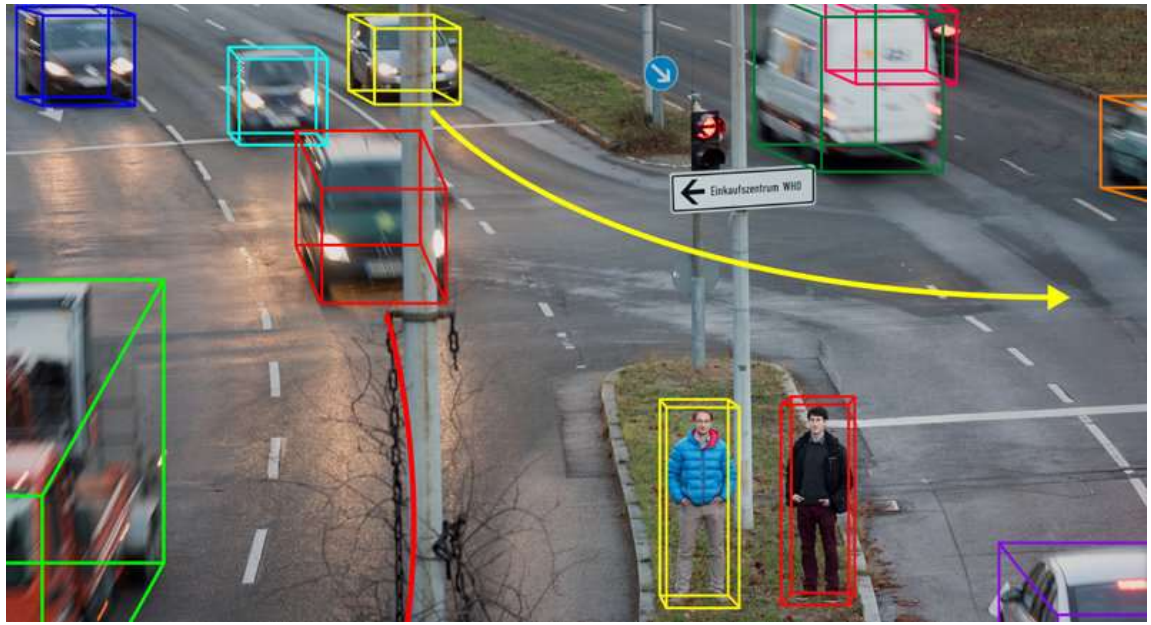
Algorithm	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification
Faktorenanalyse		X				
Principal Component Analysis	PCA	X		X		
	K-means		X			
hierarchical cluster analysis	HCA		X			
DBSCAN			X			
One Class Support Vector Machine	OCSVM			X		
Isolation Forest				X		
	LODA			X		
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X
Support Vector Machine	SVM				X	
Decision Tree					X	X
Bayes-Klassifikation					X	X
Random Forest						X
Diskriminanzanalyse				x	x	x
Logistic Regression						
Linear Regression						
Harmonic Regression						
Nächste-Nachbar-Klassifikation	K-NN	x	x			

31 | Informationstechnik II
Kapitel 6: Big Data & Machinelles Lernen

„Klassifikation“ als Repräsentant des überwachten Lernens

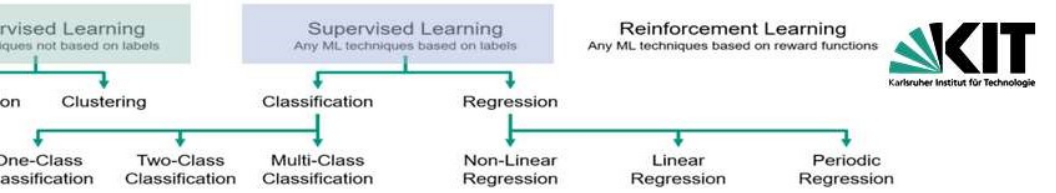
- Ein bekanntes Beispiel ist die Objekterkennung in Bilddaten und die Zuordnung eines gefundenen Objekts zu einer bestimmten Klasse.

- So müssen in modernen Fahrzeugen Objekte in Echtzeit erkannt und klassifiziert werden, beispielsweise Fahrzeuge, Verkehrsschilder, Fußgänger, Hindernisse … (s. Beispiel „Fahrt über den KIT-Campus“)
- Die Objekterkennung wird dafür mittels ausgesuchter, annotierter Datensätze trainiert.



„Regression“ als Repräsentant des überwachten Lernens

Ziel: Approximation

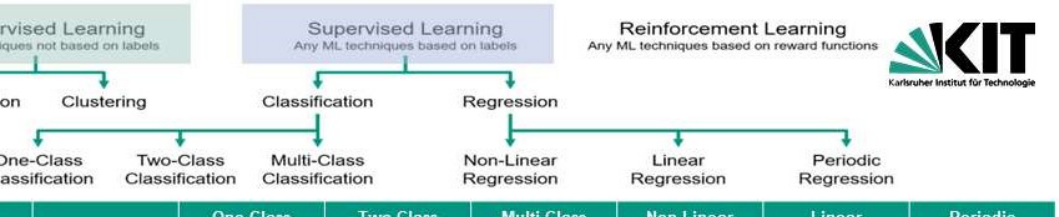


	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
		X					
	X						
	X						
	X						
		X					
		X					
		X					
		X (Autoencoder)	X	X	X	X	X
			X				
			X	X			
			X	X			
				X			
		x	x	x			
					X	X	
						X	
							X
	x						

Institut für Technik der Informationsverarbeitung (ITIV)
Prof. Dr.-Ing. Eric Sax, © 2020

- Bei den beiden bisher vorgestellten Problemstellungen ist das Ziel die Zuordnung von Eingangswerten zu einem diskreten Ausgangswert – einer Klasse oder einem Cluster.
  - ► Im Gegensatz dazu wird bei der Regression eine Funktion approximiert, um beliebigen Eingangswerten einen kontinuierlichen Ausgangswert zuordnen zu können.
  - ► Für die Approximation einer Funktion sind üblicherweise Stützstellen mit den dazugehörigen Stützwerten notwendig.
  - ► Daher ist die Regression eine Aufgabe für überwachtes Lernen.
  - ► Liegt beispielsweise das Wissen vor, dass Niederschlag, Temperatur und Treibstoffverbrauch zusammenhängen, kann durch eine Regression der veränderte Treibstoffverbrauch in Abhängigkeit von Niederschlag und Temperatur vorhergesagt werden.

„Regression“ als Repräsentant des überwachten Lernens

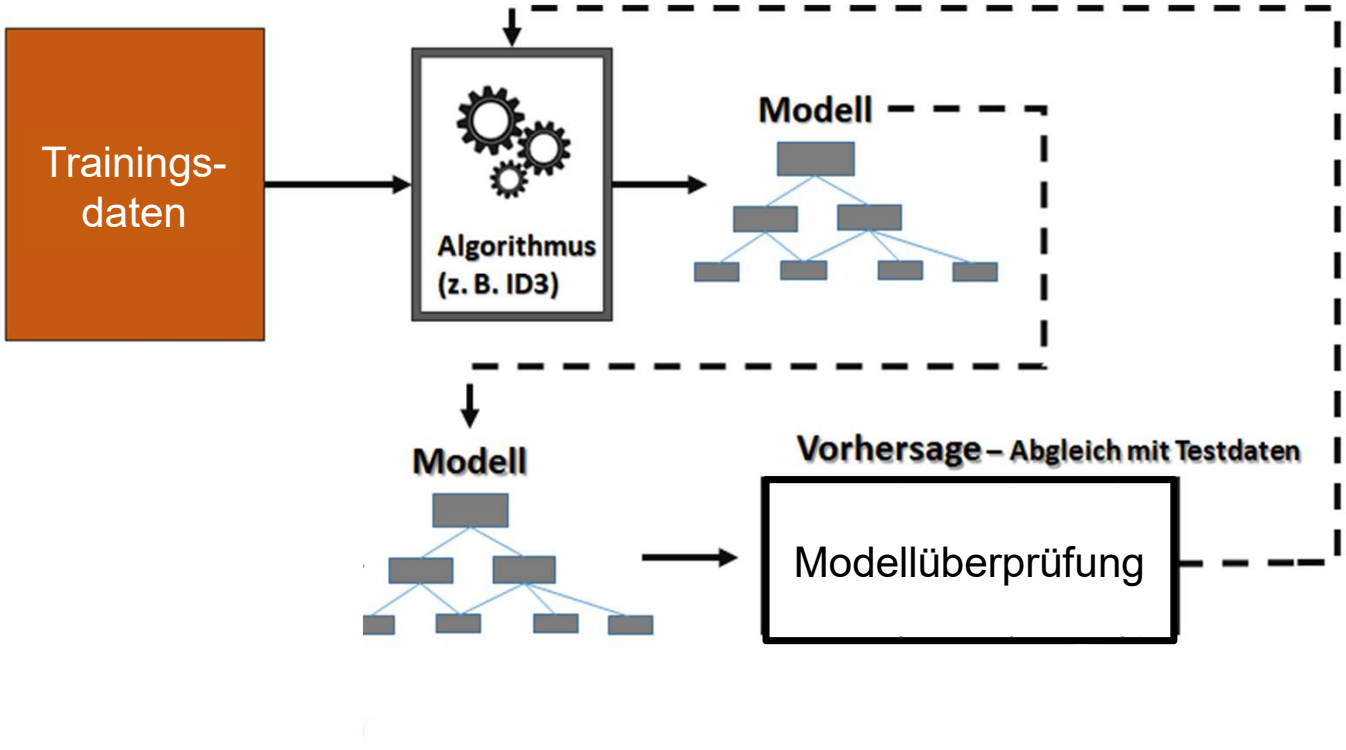


	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
		X					
	X						
	X						
	X						
		X					
		X					
		X					
		X (Autoencoder)	X	X	X	X	X
			X				
			X	X			
			X	X			
				X			
		x	x	x			
					X	X	
						X	
							X
	x						

Institut für Technik der Informationsverarbeitung (ITIV)
Prof. Dr.-Ing. Eric Sax, © 2020

- Die Problemstellungen können einzeln oder in Kombination auftreten, je nach zur Verfügung stehender Datensätze.
- Im Unterschied zu Clusterbildung wird ein Zusammenhang entdeckt.
- Die Regression nutzt vorhandenes, erfasstes Wissen, um eine Vorhersage zu treffen, die bisher nicht möglich war.

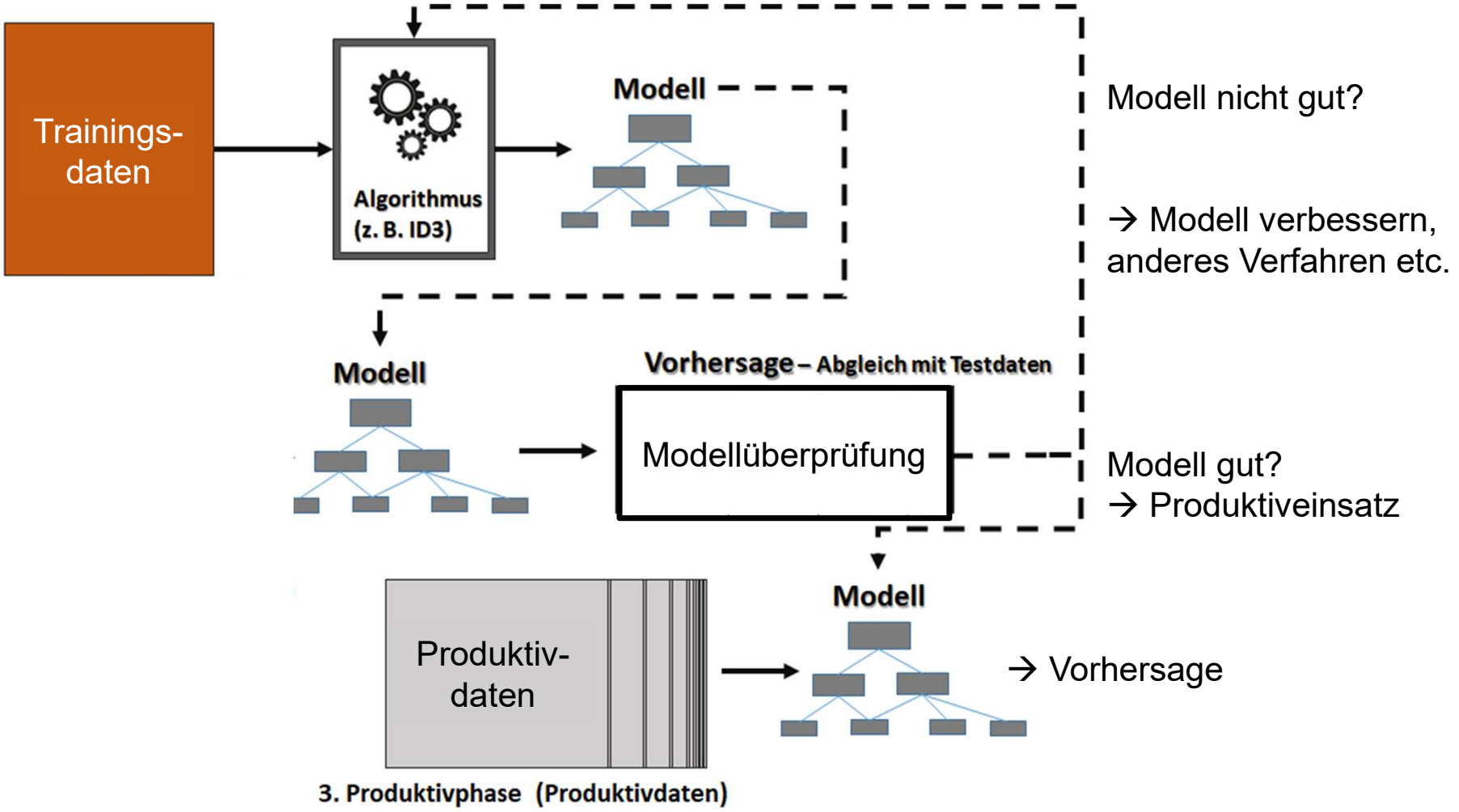
Grundsätzliches Vorgehen beim überwachten Lernen



Modell nicht gut?

→ Modell verbessern, anderes Verfahren etc.

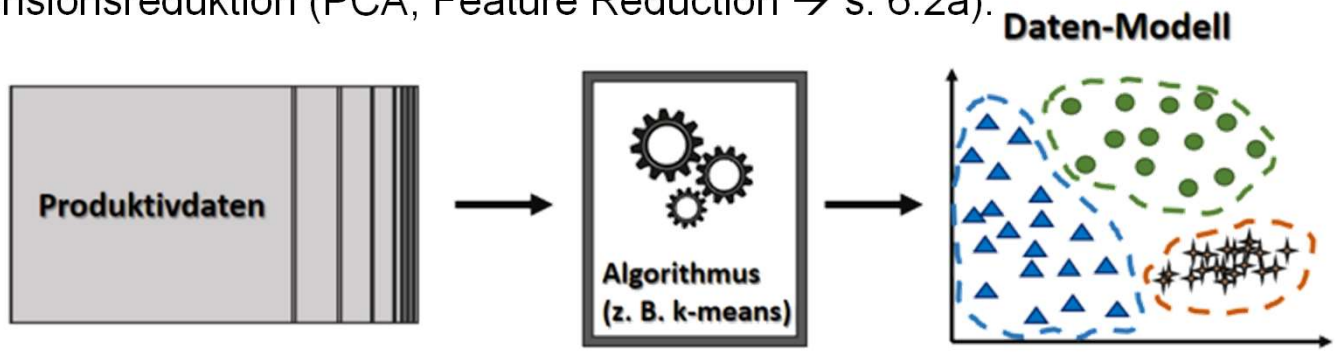
Grundsätzliches Vorgehen beim überwachten Lernen



Vorgehen beim unüberwachten Lernen

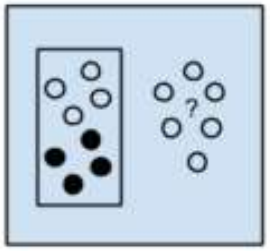
Mit unüberwachtem Lernen verborgene Strukturen identifizieren

- Beim unüberwachten Lernen geht es um nicht gekennzeichnete Daten (*unlabeled data*).
- Die möglichen Antworten/Ergebnisse sind gänzlich unbekannt.
- Folglich können wir den Algorithmus nicht trainieren, indem wir ihm die Ergebnisse, auf die er kommen soll, im Rahmen einer Trainingsphase vorgeben (vgl. überwachtes Lernen), sondern wir nutzen Algorithmen, die die Struktur der Daten erkunden und für uns Menschen sinnvolle Informationen aus Ihnen bilden.
  - ► oder auch nicht → häufig bleibt es beim Versuch → der Erfolg ist nicht garantiert!
- Zwei wesentliche Kategorien des unüberwachten Lernens
  - zum einem das Clustering, welches im Grunde ein unüberwachtes Klassifikationsverfahren darstellt
  - zum anderen die Dimensionsreduktion (PCA, Feature Reduction → s. 6.2a).



Unterschiedliche Arten von Lernen

Zusammenfassung



Supervised Learning

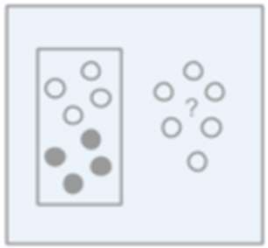
- Ge-labelte Daten
- Lernen/Vorhersagen von Output aus Input-Daten
- Herausforderung:
  - extrapolieren
  - generalisieren
- Beispiele
  - Klassifizierung
  - Regression

Unsupervised Learning

- Un-gelabelte Daten
- Finden von versteckten, Strukturen in Daten
- Herausforderung:
  - Subjektiver als SL
  - Validierung
- Beispiele
  - Clustering
  - Dimensionsreduktion

Unterschiedliche Arten von Lernen

Zusammenfassung

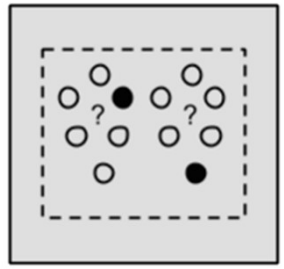


Supervised Learning

- Ge-labelte Daten
- Lernen/Vorhersagen von Output aus Input-Daten
- Herausforderung:
  - extrapolieren
  - generalisieren
- Beispiele
  - Klassifizierung
  - Regression

Unsupervised Learning

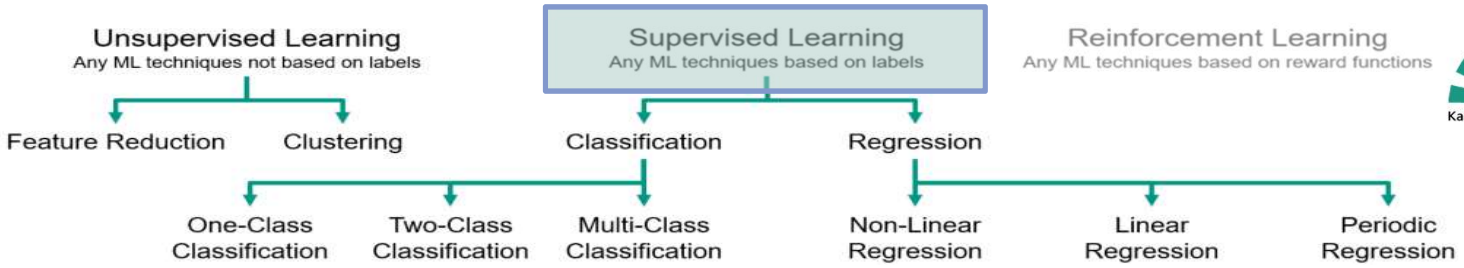
- Un-gelabelte Daten
- Finden von versteckten, Strukturen in Daten
- Herausforderung:
  - Subjektiver als SL
  - Validierung
- Beispiele
  - Clustering
  - Dimensionsreduktion



Semi-Supervised Learning

- nur ein Teil der Daten ist ge-labelt
- Nur für einen geringen Teil der Daten steht eine Annotation zur Verfügung
- geringerer Label-Aufwand im Vergleich zu Supervised Learning
- Bessere Leistung als bei Unsupervised Lernen

Machinelles Lernen



Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Klassifikation vs. Regression

Beispiel

- Folgende Situation
  - Eine Versicherung bietet Leistungspaket D1 und D2 an.
  - Nun wird eine Werbekampagne mit Anreiz A konzipiert.

- Betrachtet man die folgenden drei ähnlichen Fragen:
  - *Wird der Kunde auf ein Angebot für das Paket D1 eingehen, wenn er den Anreiz A erhält?*
    - Hierbei handelt es sich um eine Klassifizierung, weil das Ziel binär ist (entweder geht der Kunde auf das Angebot ein oder nicht).
  - *Welches Dienstpaket (D1, D2 oder keines) wird der Kunde wahrscheinlich kaufen, wenn er den Anreiz A erhält?*
    - Hierbei handelt es sich ebenfalls um eine Klassifizierung, allerdings mit einer Zielvariablen, die 3 Werte annehmen kann.
  - *Wie häufig wird dieser Kunde das Paket bestellen/nutzen?*
    - Dies ist ein klassisches Beispiel für eine Regression, denn die Zielvariable ist numerisch.

Klassifikation vs. Regression

Beide gehören den überwachten Verfahren an.

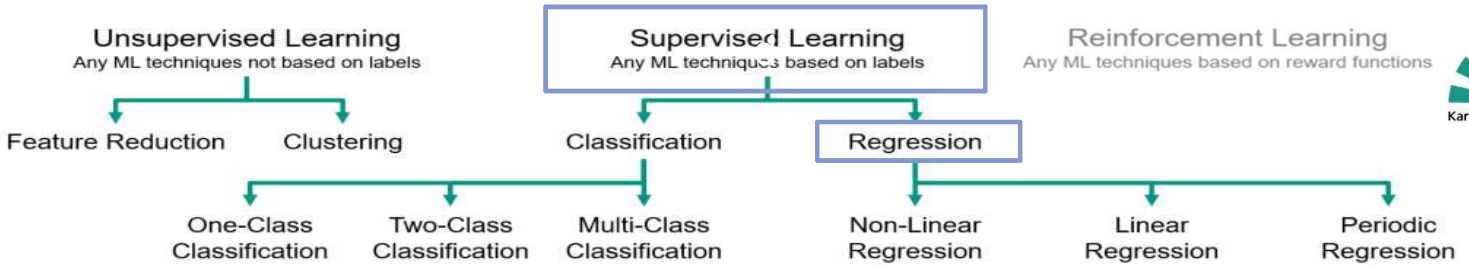
- Klassifizierung versucht, für ein zur Grundgesamtheit gehörendes Individuum vorherzusagen, zu welchen (einigen wenigen) Klassen dieses Individuum gehört.
  - Für gewöhnlich schließt die Zugehörigkeit zu einer Klasse die Zugehörigkeit zu einer anderen aus.
- Eine Frage, die der Klassifizierung dient, ist bspw.:
  - *»Welche Kunden werden ein bestimmtes Angebot wahrscheinlich annehmen?«*
    → In diesem Beispiel könnte man die beiden Klassen als „wird annehmen“ und „wird nicht annehmen“ bezeichnen.
- Die Klassifizierung liefert ein Modell, das bei Auftauchen eines neuen Individuums ermittelt, zu welcher Klasse dieses Individuum gehört.

- Regression versucht, für jedes Individuum den numerischen Wert einer Variable abzuschätzen oder vorherzusagen.
  - *»Wie oft wird ein bestimmter Kunde den Dienst nutzen?«*
    → ist ein Beispiel für eine Frage, die durch eine Regression beantwortet werden kann.
- Die hier vorherzusagende Eigenschaft (die Variable) ist die Dienstnutzung.
- Man könnte dafür ein Modell entwickeln, indem man die Dienstnutzung vergleichbarer Individuen der Grundgesamtheit betrachtet.
- Eine Regression liefert ein Modell, das den Wert einer Variable für ein bestimmtes Individuum abschätzt.

Anmerkungen

- Bei der Regression ist die Zielvariable numerisch, bei der Klassifizierung hingegen geht es um die Zuordnung zu einem (oft binären) kategorialen Ziel.
  - Zwanglos könnte man sagen, dass eine Klassifizierung vorhersagt, ob etwas der Fall ist, eine Regression hingegen sagt voraus, in welchem Ausmaß dies eintritt.
- Scoring ist sehr ähnlich.
  - Ein auf ein Individuum angewendetes Scoring-Modell ermittelt statt einer Klassenvorhersage eine Punktzahl, die die Wahrscheinlichkeit dafür angibt, dass ein Individuum zu einer bestimmten Klasse gehört (s. Beispiel „Fahrt über den Campus“, Prozente am Objekt).
- Im Szenario der Angebotsannahme würde ein Scoring-Modell die einzelnen Kunden auswerten und eine Punktzahl liefern, die angibt, wie wahrscheinlich es ist, dass ein Kunde auf das Angebot eingeht.

Machinelles Lernen



Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Regression
Begriff

- Re·gres·si·on
  - Langsamer Rückgang; rückläufige Tendenz
  - Gegenteil: Progression
- Die Regression kann als die „Mutter aller Verfahren“ bezeichnet werden.
  - ► Das liegt einerseits daran, dass sie in ihrer einfacheren Ausprägung als lineare Regression gut verständlich und damit nachvollziehbar ist, aber auch daran, dass die im Verfahren eingesetzten Methoden in vielen anderen Verfahren eine analoge Anwendung finden.
  - ► Als Prognoseverfahren ist es sicher das am häufigsten verwendete Schätzmodell in der Praxis.
- Folglich
  - ► Regressions-Analysen sind statistische Analyseverfahren, die zum Ziel haben, Beziehungen zwischen einer abhängigen und einer oder mehreren unabhängigen Variablen zu modellieren.
  - ► Sie werden insbesondere verwendet, wenn Zusammenhänge quantitativ zu beschreiben oder Werte der abhängigen Variablen zu prognostizieren sind.

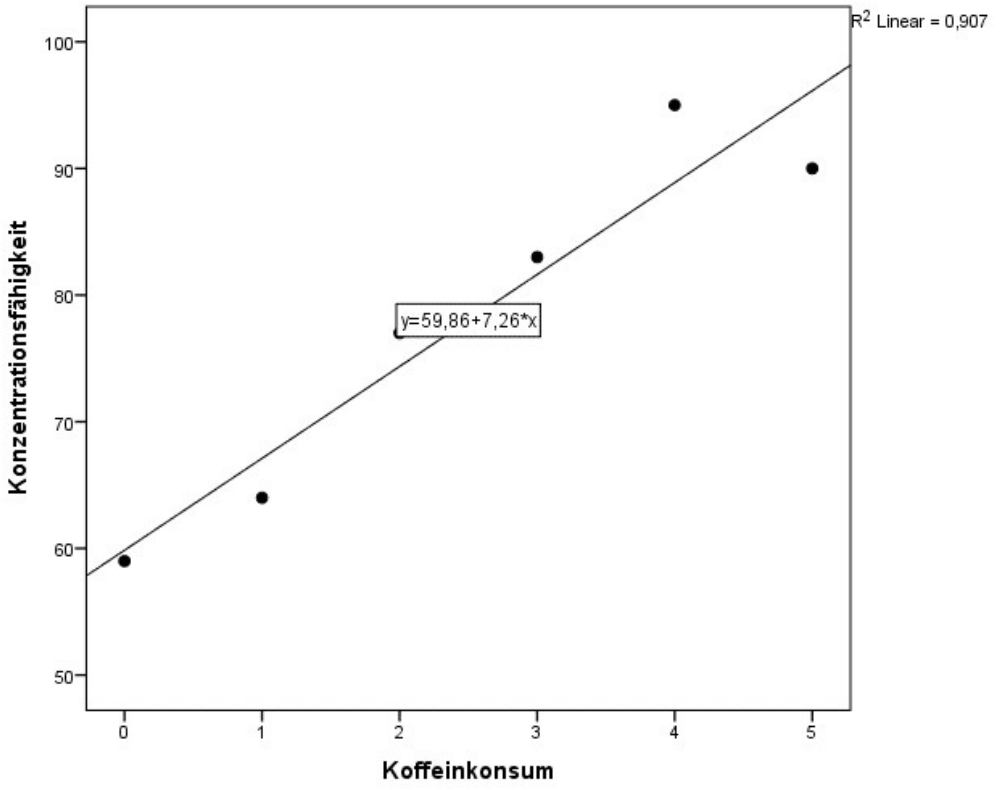

8RT

Einfachregression

Univariate Regression

- Der einfachste Fall ist die lineare, univariate Regression, also mit jeweils 1 messbaren abhängigen und unabhängigen Variablen .
  - Es wird ein linearer Zusammenhang zwischen den Variablen vermutet.
- Beispiel: 6 Probanden, die 0 bis 5 Tassen Kaffee trinken.
  - Annahme → Wir haben eine Methode zur Messung der Konzentrationsfähigkeit nach Kaffeekonsum
  - Ergebnis/Visualisierung → s. rechts
    - Y-Achse (abhängige Variable) ist die Konzentrationsfähigkeit
    - X-Achse (unabhängige Variable) ist der Koffeinkonsum
- Der Zusammenhang der beiden Variablen kann mit einer linearen Funktion dargestellt werden

$$y = \beta_0 + \beta_1 x_1$$



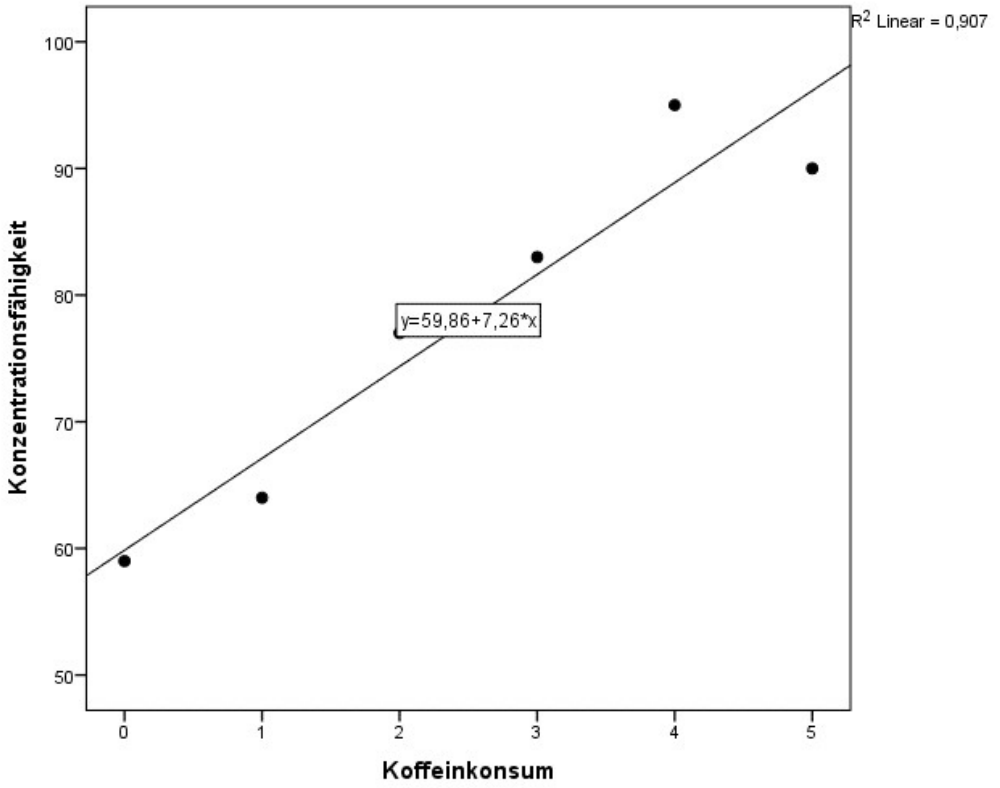
	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6
Koffeinkonsum (x)	3	0	5	2	1	4
Konzentrationsfähigkeit (y)	83	59	90	77	64	95

univariat: nur eine Variable besitzend; eindimensional
linear: vergleiche PCA

Einfachregression

Interpretation

- Man versucht nun, die Regressionsgerade so in die Punktewolke einzupassen, dass die Abstände der Punkte von der Gerade minimiert werden
  - häufig wird hierbei das Quadrat der Abstände genommen → Ziel ist die Optimierung des mittleren quadratischen Fehlers.
- Wenn man mit den statistischen Gütezahlen zufrieden ist (also z. B. den quadrierten Abweichungsdistanzen), erhält man dann ein Modell, das für die Prognose verwendet werden kann.



	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6
Koffeinkonsum (x)	3	0	5	2	1	4
Konzentrationsfähigkeit (y)	83	59	90	77	64	95

$$y = \beta_0 + \beta_1 x_1 \rightarrow y = 59{,}86 + 7{,}26 * x$$


8RT

Mehrfachregression

Komplexität: Häufig ist die Verteilung nicht linear approximierbar.

- Man wird daher:
  - … weitere erklärende Variablen miteinbeziehen wollen. Dadurch bekommt man eine nicht mehr so einfach zu visualisierende multivariate Regression.

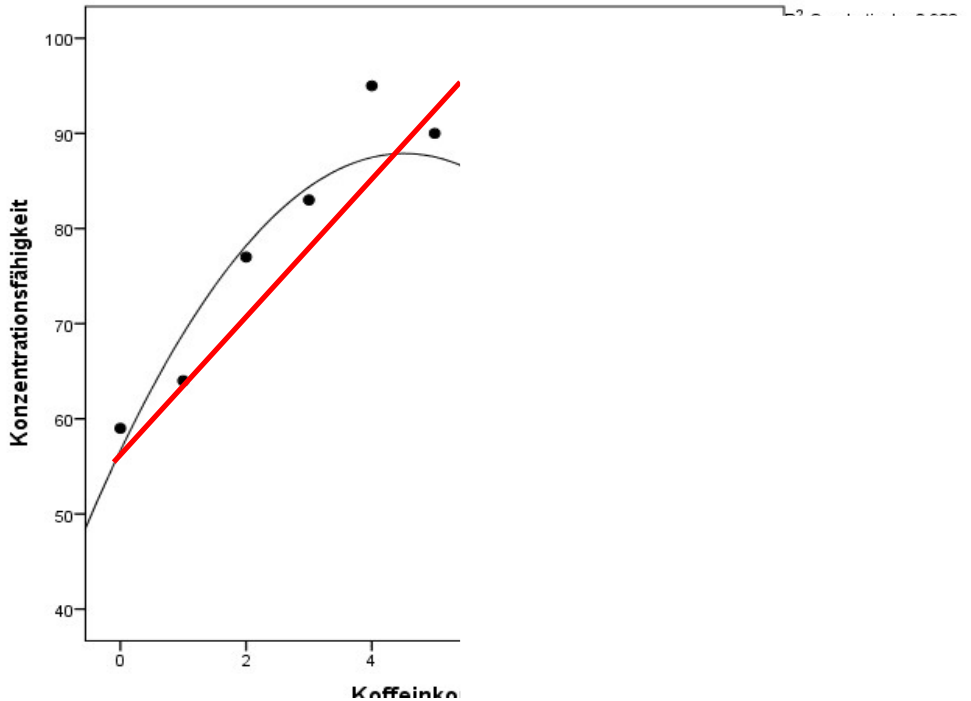
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

  - … das Schätzverfahren ändern. Durch das Quadrieren der Abstände haben einzelne Ausreißer eine relativ große Auswirkung auf die Regressionsfunktion, die man eventuell begrenzen möchte.
  - … keinen linearen Zusammenhang annehmen, sondern versuchen, die Art des funktionalen Zusammenhangs anhand der Daten herzuleiten.
- *Remark: Nichtlineare Zusammenhänge können durch eine nichtlineare Regression modelliert werden.*
  - *Hierzu wird versucht den nichtlinearen Ansatz in einen linearen zu transformieren, um dann die Gleichung mithilfe der Gauß'schen Methode der kleinsten Quadrate zu bestimmen*

Beispiel für eine nichtlineare Regression

„Koffeinkonsum“

- Mglw. führt eine zu hohe Koffeindosierung wieder zu einer Abnahme der Konzentrationsfähigkeit.
  - Dazu erweitert man die Stichprobe um 5 Personen, die zwischen 6 und 10 Tassen Kaffee trinken.



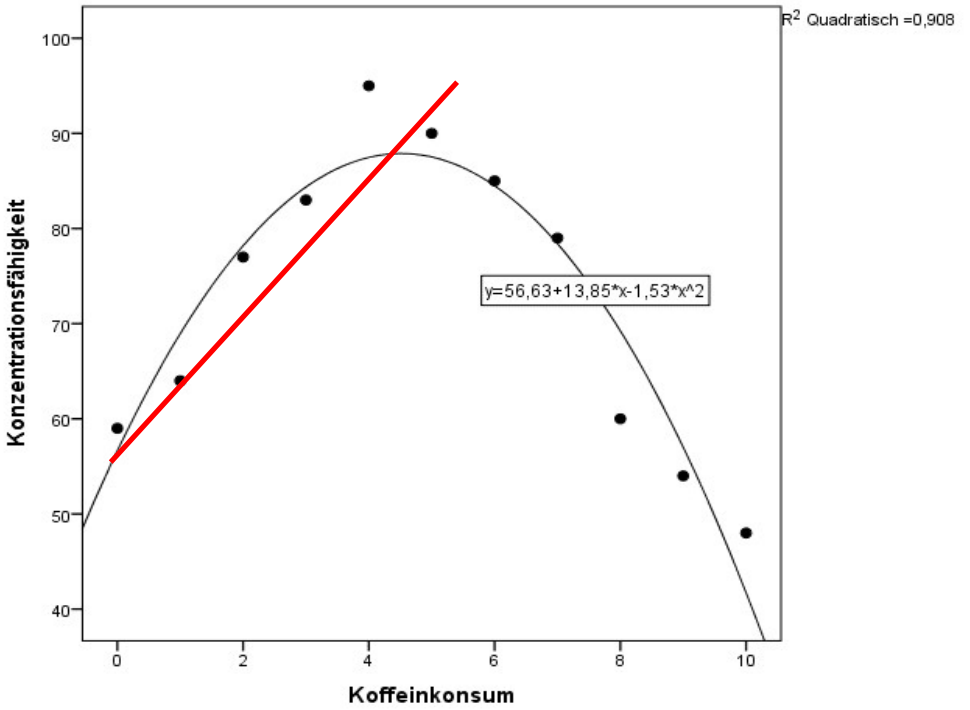
	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6
Koffeinkonsum (x)	3	0	5	2	1	4
Konzentrationsfähigkeit (y)	85	41	87	64	58	53

Beispiel für eine nichtlineare Regression

„Koffeinkonsum“

- Mglw. führt eine zu hohe Koffeindosierung wieder zu einer Abnahme der Konzentrationsfähigkeit.
  - Dazu erweitert man die Stichprobe um 5 Personen, die zwischen 6 und 10 Tassen Kaffee trinken.
  - Vermutung offensichtlich richtig → ein zu hoher Koffeinkonsum führt wieder zu einer Abnahme der Konzentrationsfähigkeit.
  - Der ursprünglich lineare Zusammenhang war eine Täuschung, da man die Beziehung der Variablen lediglich bis 5 Tassen Kaffee untersucht hatte.
- Eine Kurve wie bei unseren Daten ist eine sogenannte quadratische Funktion.
  - Quadratische Funktionen sind allerdings ähnlich unkompliziert wie lineare Funktionen, da man sie im Plot leicht anhand ihrer Kurvenform erkennen kannst.
- Man erweitert die ursprüngliche Regressions-gleichung um den Term x^2.
  - Die Gleichung sieht dann folgendermaßen aus:

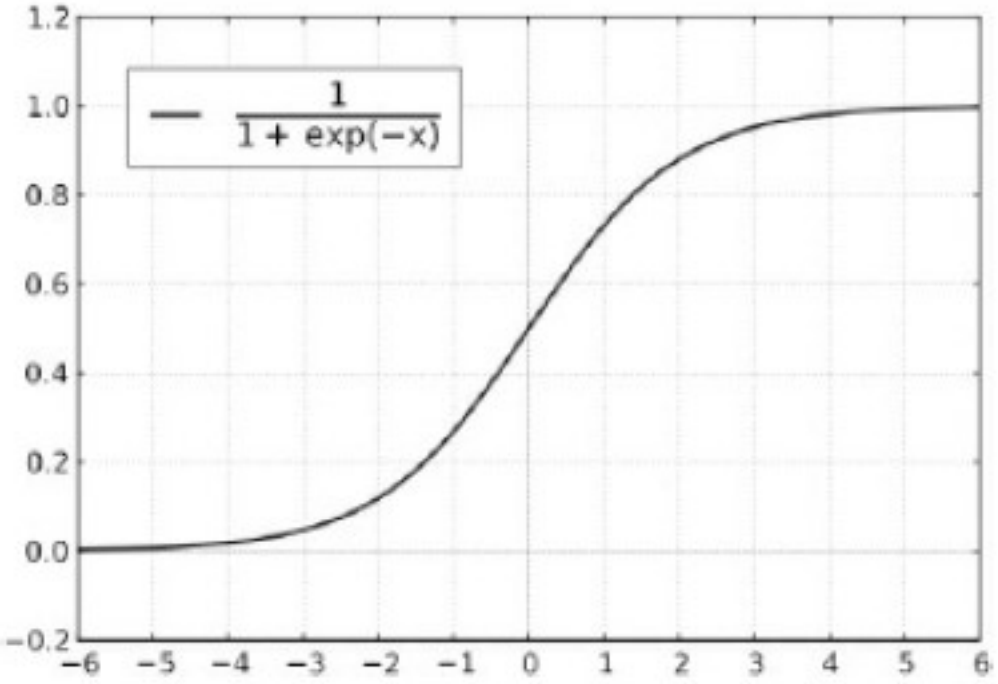
$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2$$



	Person 1	Person 2	Person 3	Person 4	Person 5	Person 6
Koffeinkonsum (x)	3	0	5	2	1	4
Konzentrationsfähigkeit (y)	85	41	87	64	58	53
	Person 7	Person 8	Person 9	Person 10	Person 11	
Koffeinkonsum (x)	8	10	7	9	6	
Konzentrationsfähigkeit (y)	60	48	79	54	85	

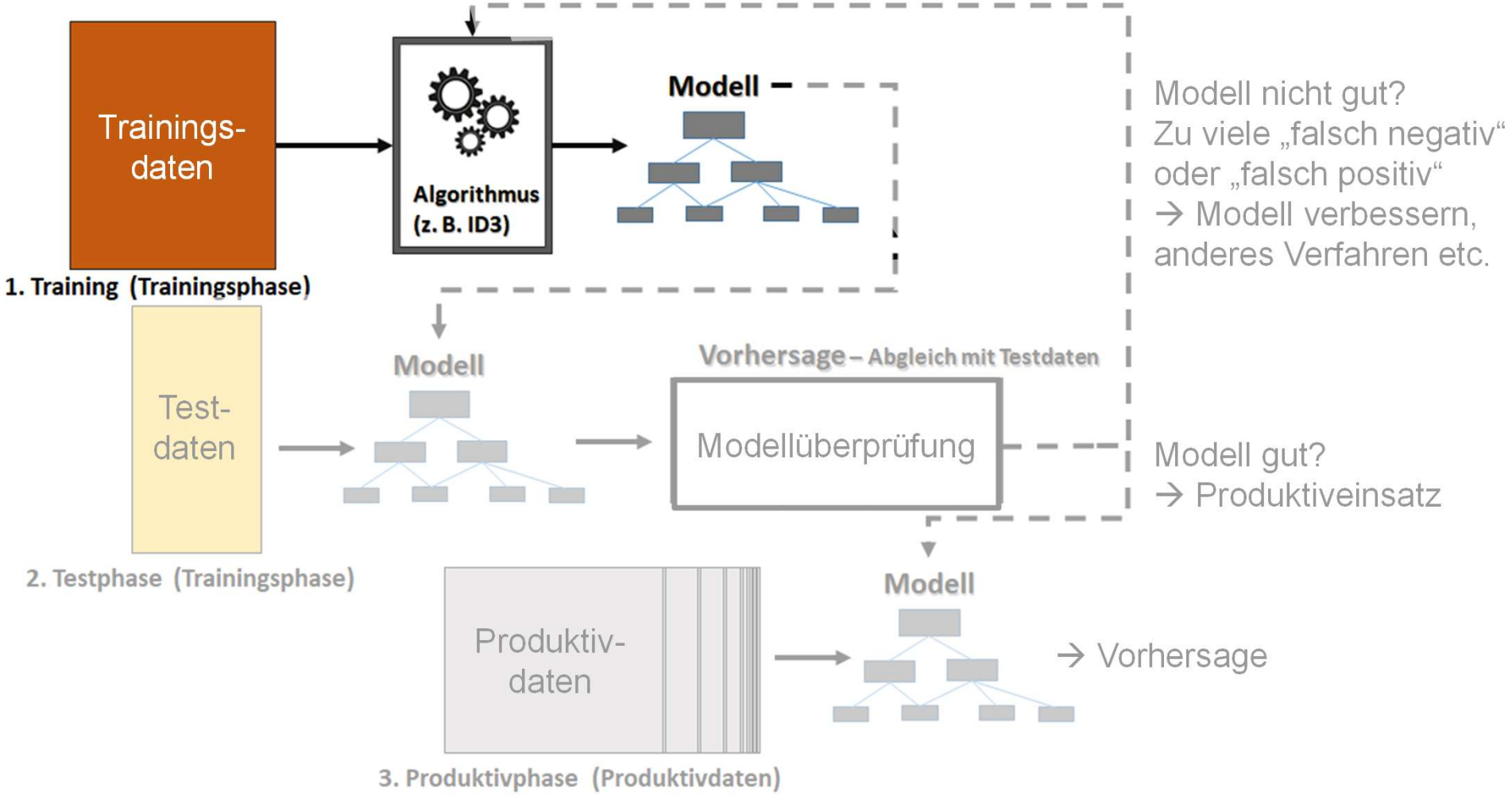
Logistische Regression

- Darunter versteht man eine Regressions-analyse zur Modellierung der Verteilung diskreter abhängiger Variablen, z.B. im Falle von Ja-Nein-Entscheidungen (z.B. ob der Kunde kreditwürdig ist oder nicht).
- Man könnte derartige Entscheidungsmodelle auch mit einer linearen Regression lösen, wobei man dann einen Wert für y als Grenze zwischen „Ja“ und „Nein“ definiert.
- Das führt dazu, dass Werte knapp an diesem Wert relativ willkürlich in die eine oder andere Klasse zugeordnet würden.
- Die logistische Verteilfunktion mit ihrer hohen Steigung im „Mittelteil“ der Kurve teilt sozusagen das „Ja“ besser von dem „Nein“, da der Bereich in der Nähe des Cut Offs sehr steil ist und damit nur wenige Werte „betrifft“.



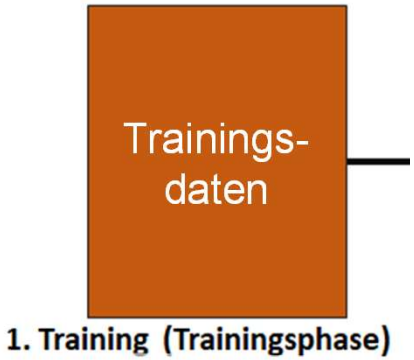
Grundsätzliches Vorgehen beim überwachten Lernen

Trainieren

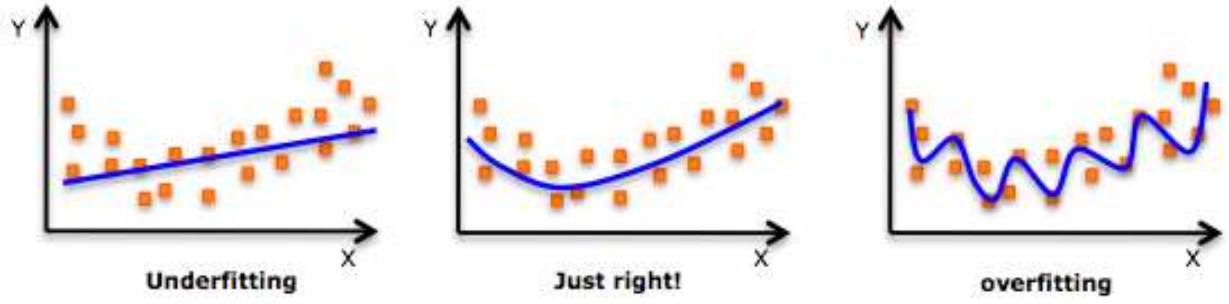


Vorgehen beim überwachten Lernen

Over-/Underfitting

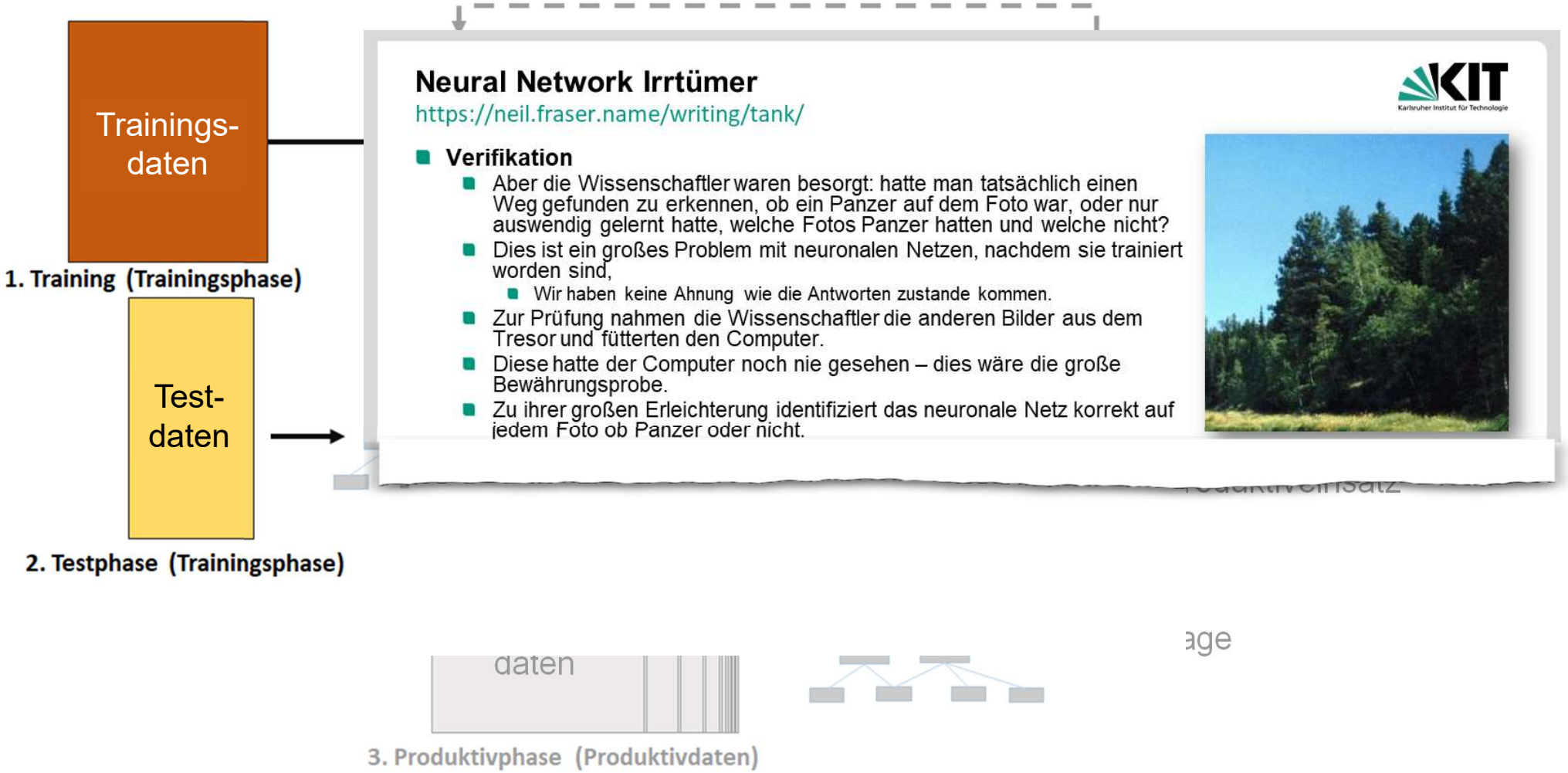


- Overfitting
  - Trainiertes Modell ist "zu gut" trainiert und nun zu eng mit dem Trainingsdatensatz übereinstimmend.
  - Dies geschieht, wenn das Modell zu komplex ist (d.h. zu viele Merkmale/Variablen im Vergleich zur Anzahl der Beobachtungen).
  - Das Modell wird sehr genau auf die Trainingsdaten sein, aber wahrscheinlich sehr ungenau auf untrainierte oder neue Daten.
- Underfitting
  - Im Gegensatz dazu bedeutet Underfitting eines Modells, dass das Modell nicht zu den Trainingsdaten passt und somit die Muster in den Daten nicht erkennt.
  - Es kann nicht auf neue Daten verallgemeinert werden kann.

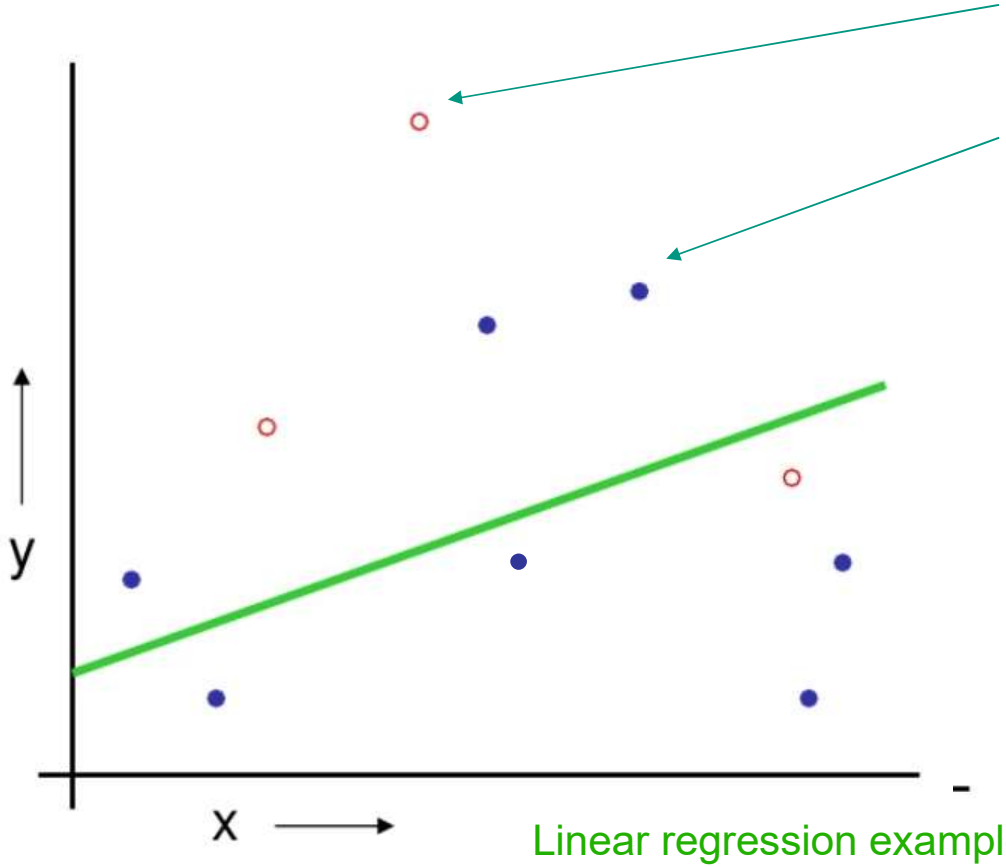


Test trainierter Modelle

Aufteilung der Daten in Training und Test notwendig, um Überanpassung zu vermeiden



Train/Test Split



Linear regression example

1. Wähle zufällig 30% der Daten als Testset aus
2. Die restlichen 70% der Daten ist das Trainingset
3. Führe Regression anhand des Trainingssets aus

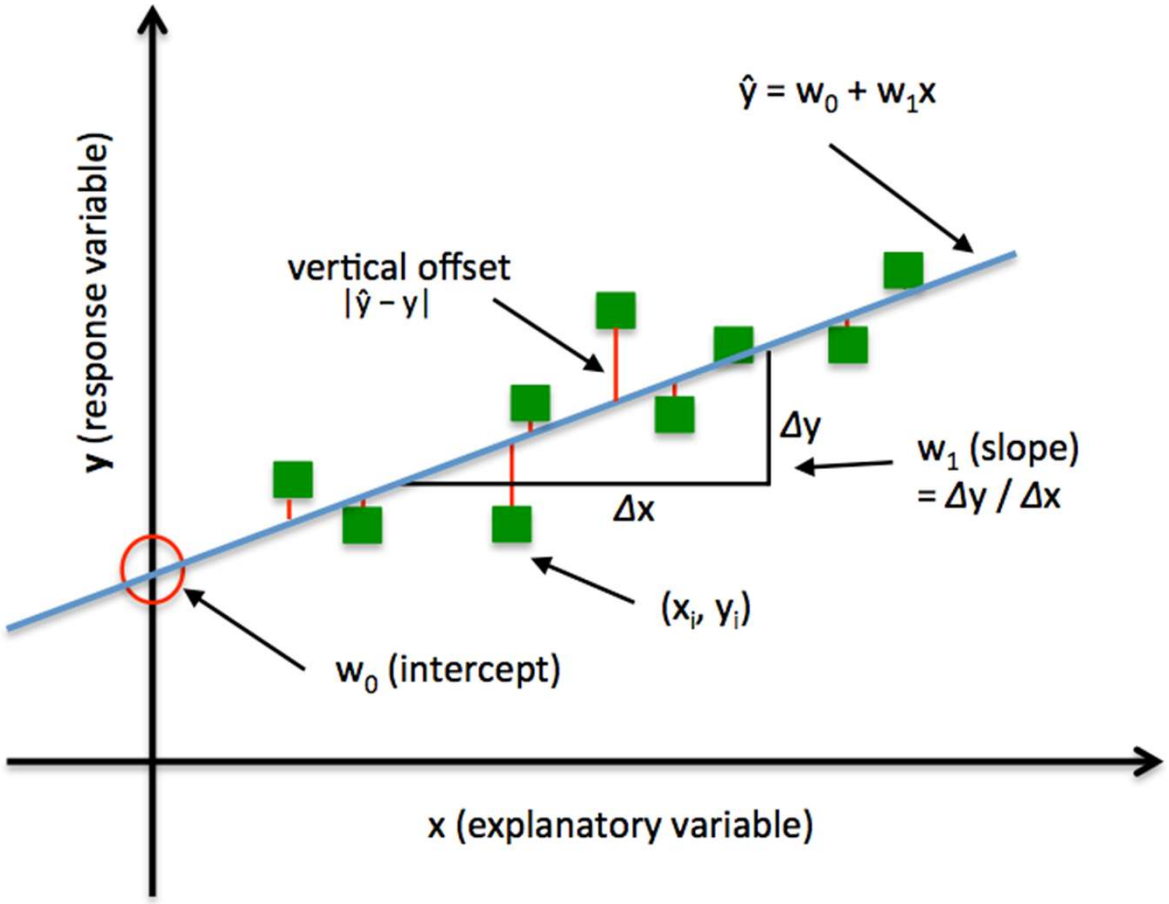
Gütemaße für die Evaluation von Modellen

Regression

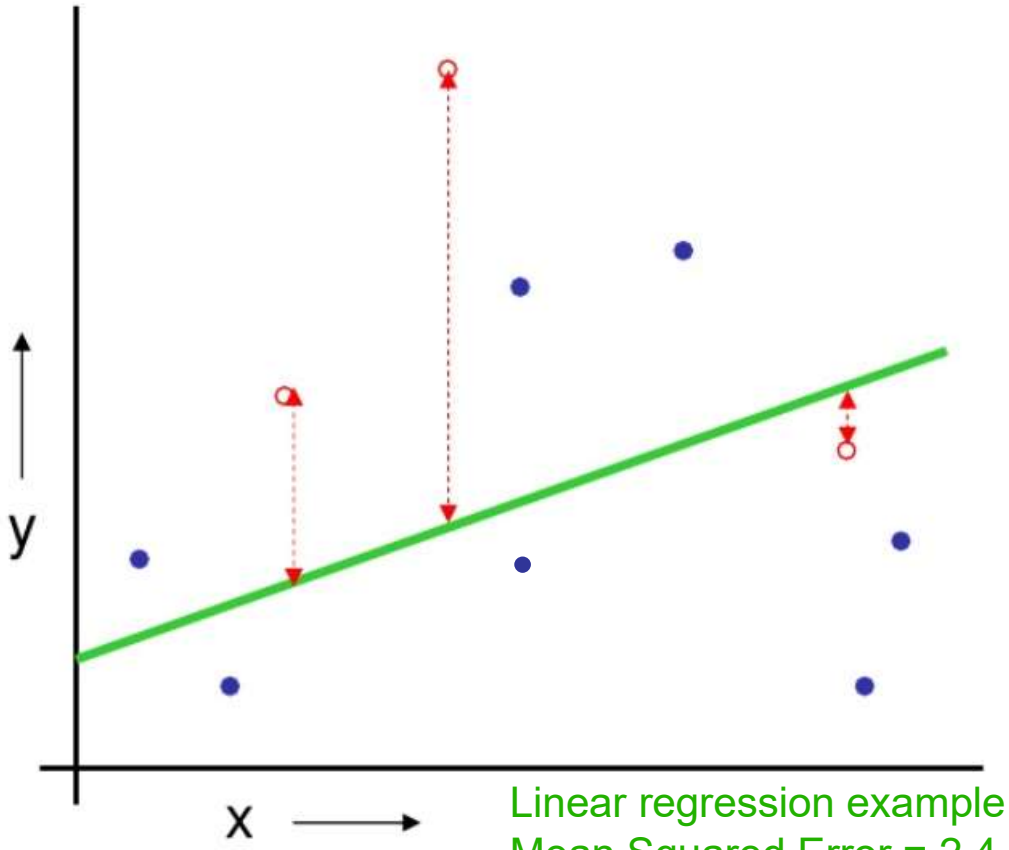
- **Mittlerer Quadratischer Fehler (MQF) (Mean Squared Error** MSE)
  - $MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$
- **Root Mean Squared Error** (RMSE)
  - $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}\, y$
- **Mean Absolut Error** (MAE)
  - $MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$

$y_i = tatsächlicher\ Werte$

$\hat{y}_i = korrekte\ Werte$



Train/Test Split

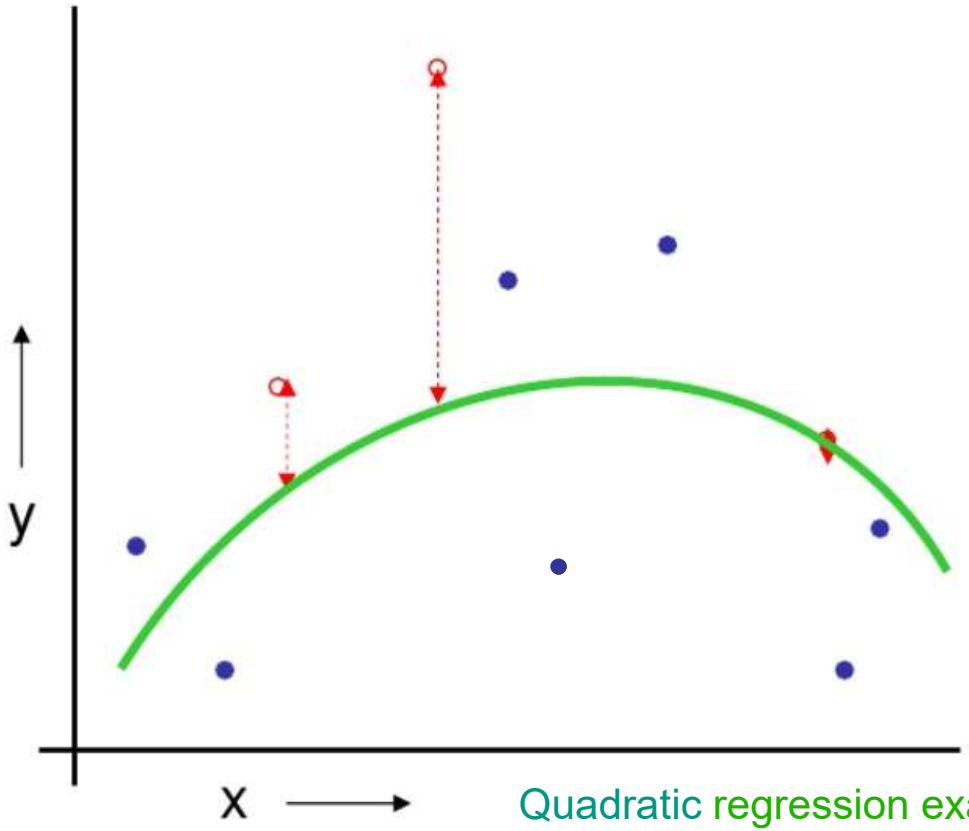


Linear regression example
Mean Squared Error = 2.4

1. Wähle zufällig 30% der Daten als Testset aus
2. Die restlichen 70% der Daten ist das Trainingset
3. Führe Regression anhand des Trainingssets aus
4. Schätze zukünftige Performance anhand des Testsets

Train/Test Split

Quadratische Regression

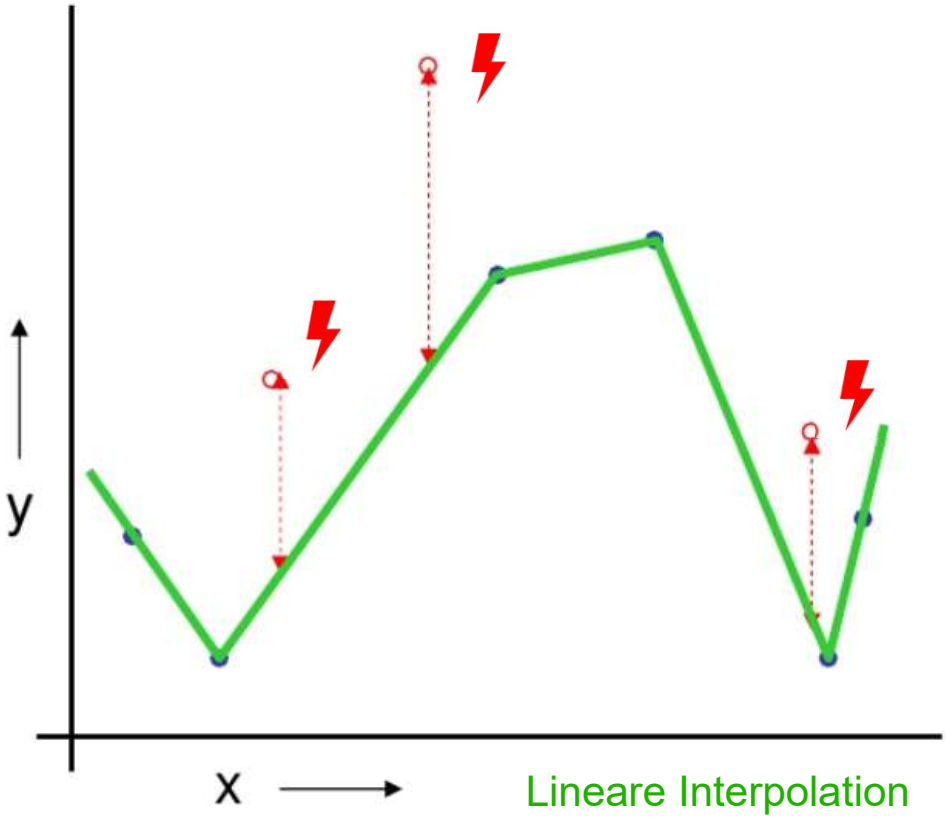


Quadratic regression example
Mean Squared Error = 0.9

1. Wähle zufällig 30% der Daten als Testset aus
2. Die restlichen 70% der Daten ist das Trainingset
3. Führe Regression anhand des Trainingssets aus
4. Schätze zukünftige Performance anhand des Testsets

Train/Test Split

„Connect the Dots“ → Ein Verfahren welches jeden Datenpunkt linear verbindet



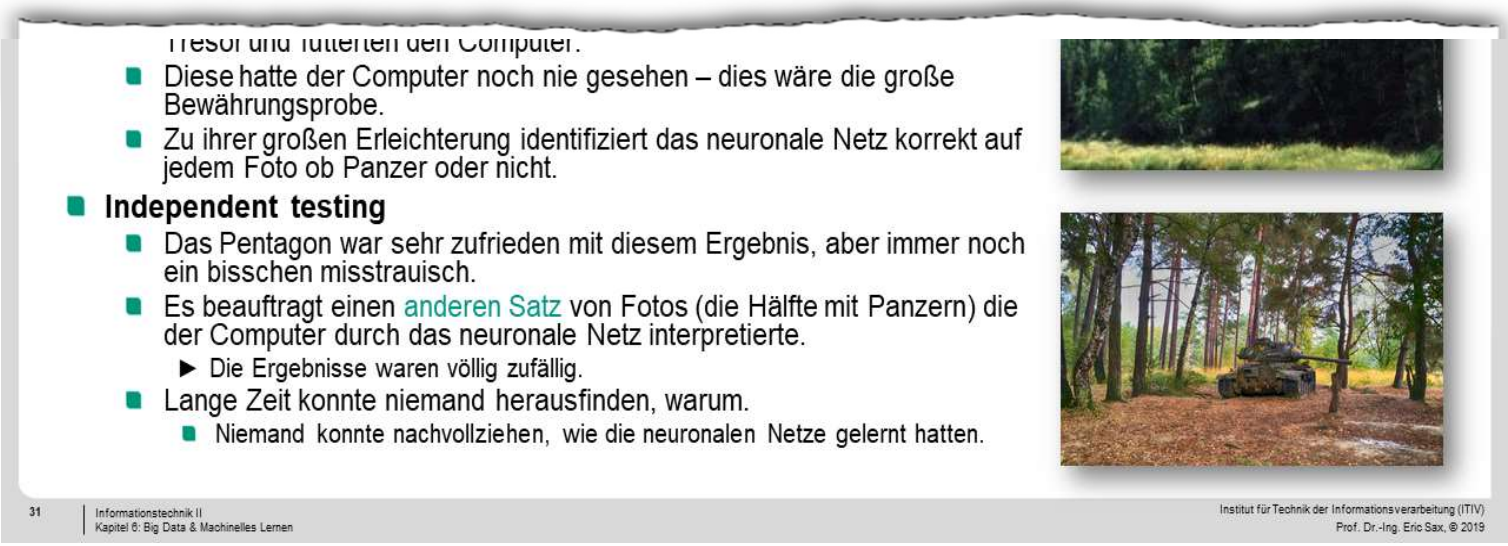
Lineare Interpolation
Mean Squared Error = 2.2

overfitted

1. Wähle zufällig 30% der Daten als Testset aus
2. Die restlichen 70% der Daten ist das Trainingset
3. Führe Regression anhand des Trainingssets aus
4. Schätze zukünftige Performance anhand des Testsets

Train/Test Split

- Vorteile
  - Sehr einfach
  - Einfache Auswahl der Methode anhand des besten test-set score (= Mean Squared Error)
- Nachteile
  - Datenverschwendung
    - Wir benutzen 30% weniger Daten zu Erzeugung des Modells
  - Falls nur wenige Daten zur Verfügung stehen, besitzt das Testset nur geringe Varianz gegenüber Trainingsdaten



26.06.20

LOOCV- Leave-one-out Cross Validation (1)

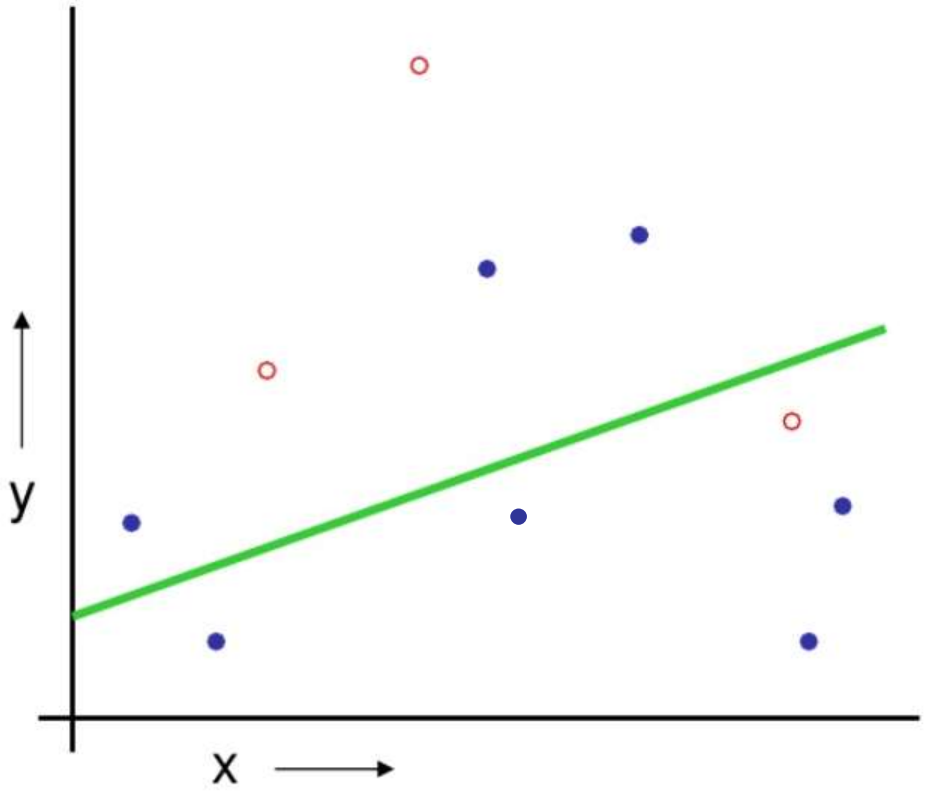
- Bei der Leave-one-out-Methode wird eine Klassifikationsmethode bei gegebener Größe N des Lerndatensatzes jeweils auf $(N-1)$ Beobachtungen angewendet und für die N-te Beobachtung getestet.
- Dieses Vorgehen wird N mal, d.h. für jede Beobachtung, wiederholt, wobei jedes Mal eine neue Klassifikationsregel bestimmt wird.
- Als **kreuzvalidierte Fehlerrate** wird die Anzahl Fehler bei den individuellen Testfällen geteilt durch N verwendet.

LOOCV- Leave-one-out Cross Validation (2)

- Damit kommt jede Beobachtung im Lerndatensatz als Testfall zum Einsatz, und bei jeder Konstruktion einer Klassifikationsregel wird nahezu der gesamte Lerndatensatz genutzt, also fast keine Information verschenkt.
- Die leave-one-out-Methode hat günstige statistische Eigenschaften:
  - Der Leave-one-out-Schätzer der Fehlerrate ist beinahe **unverzerrt**, d.h. über viele verschiedene Lernstichproben der Größe N gemittelt wird er gegen die wahre Fehlerrate streben.
- Für große Lerndatensätze ist diese Methode allerdings rechenzeitintensiv.

LOOCV- Leave-one-out Cross Validation

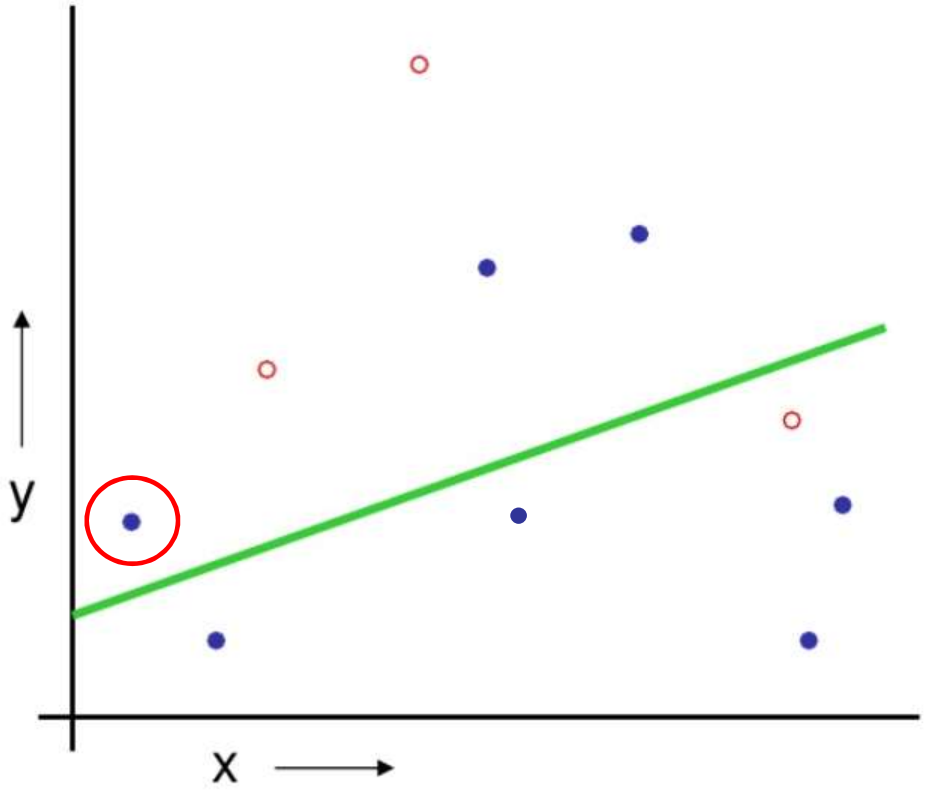
Kreuzvalidierungsverfahren (lineare Regression)



- Es wird ein Modell durch lineare Regression erstellt und dann der MSE (mean squared error) über alle Folds für dieses Modell berechnet.

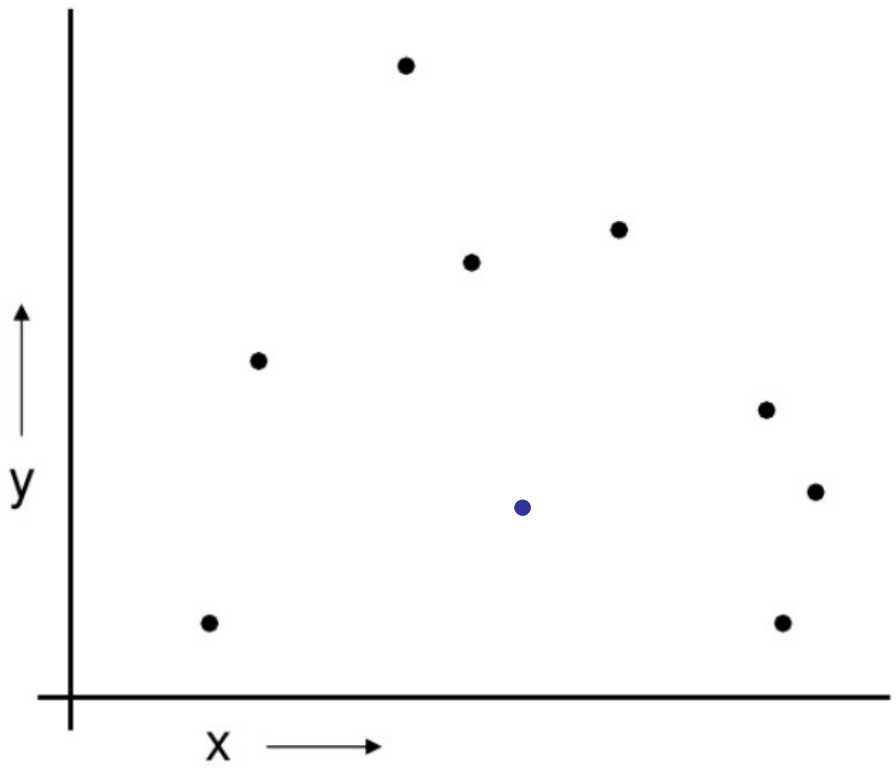
LOOCV- Leave-one-out Cross Validation

Kreuzvalidierungsverfahren



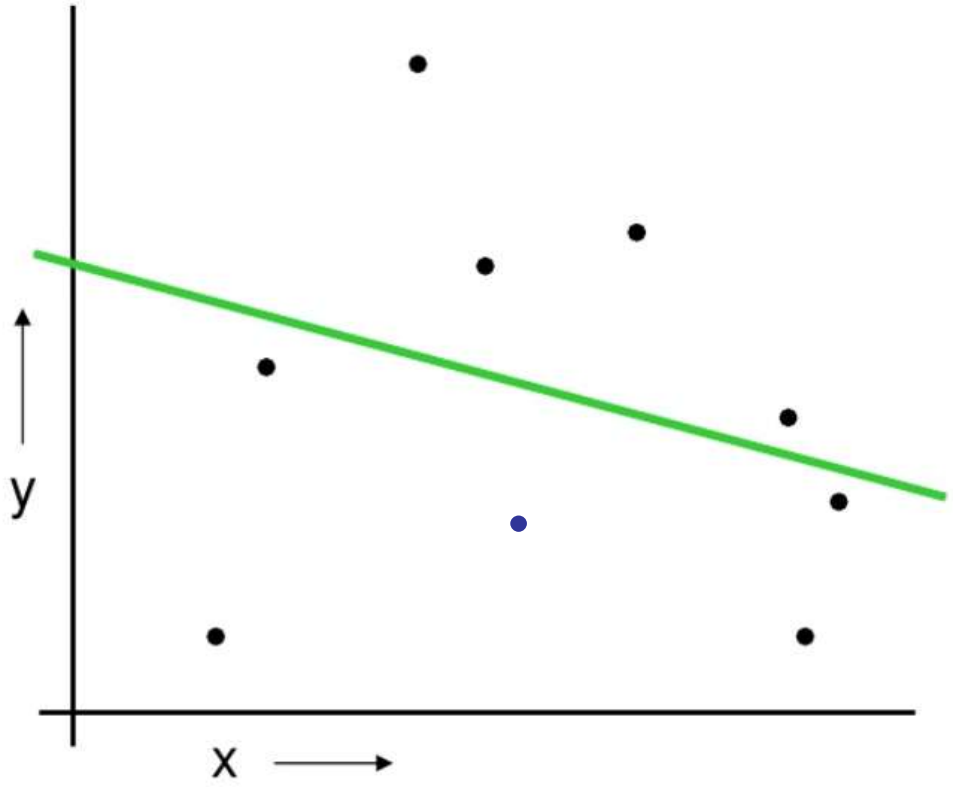
- Für k=1 bis n Datenpunkte
  1. Wähle (x_k, y_k) aus

LOOCV- Leave-one-out Cross Validation



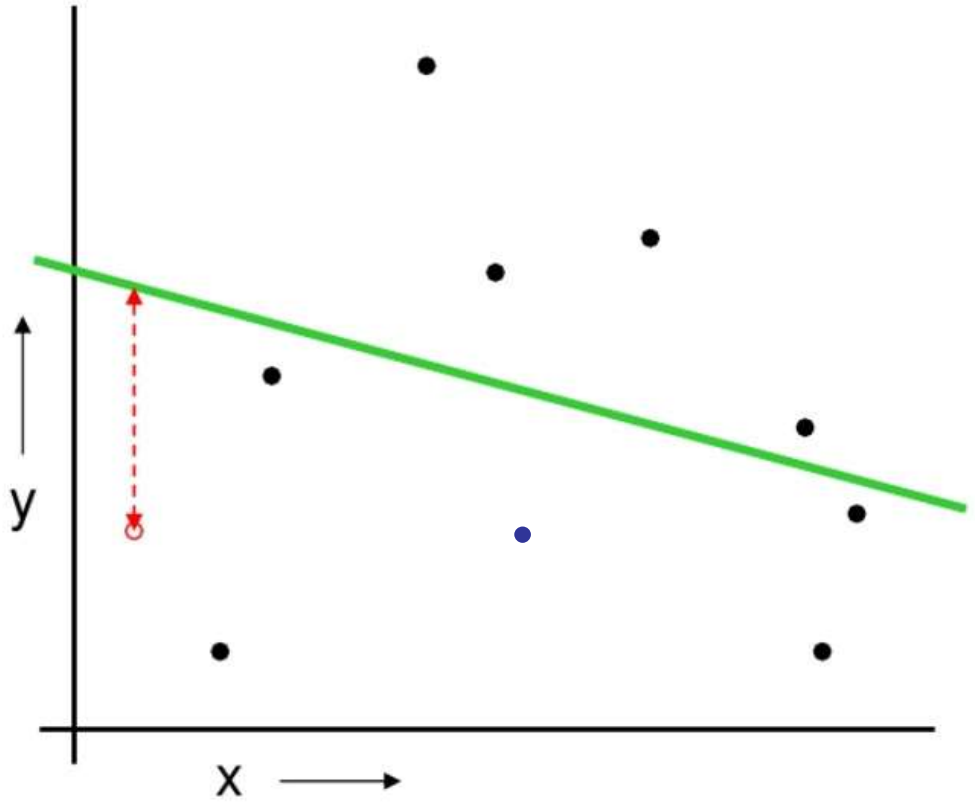
- Für k=1 bis n Datenpunkte
  1. Wähle (x_k, y_k) aus
  2. Entferne Datenpunkt (x_k, y_k) temporär aus Dataset

LOOCV- Leave-one-out Cross Validation



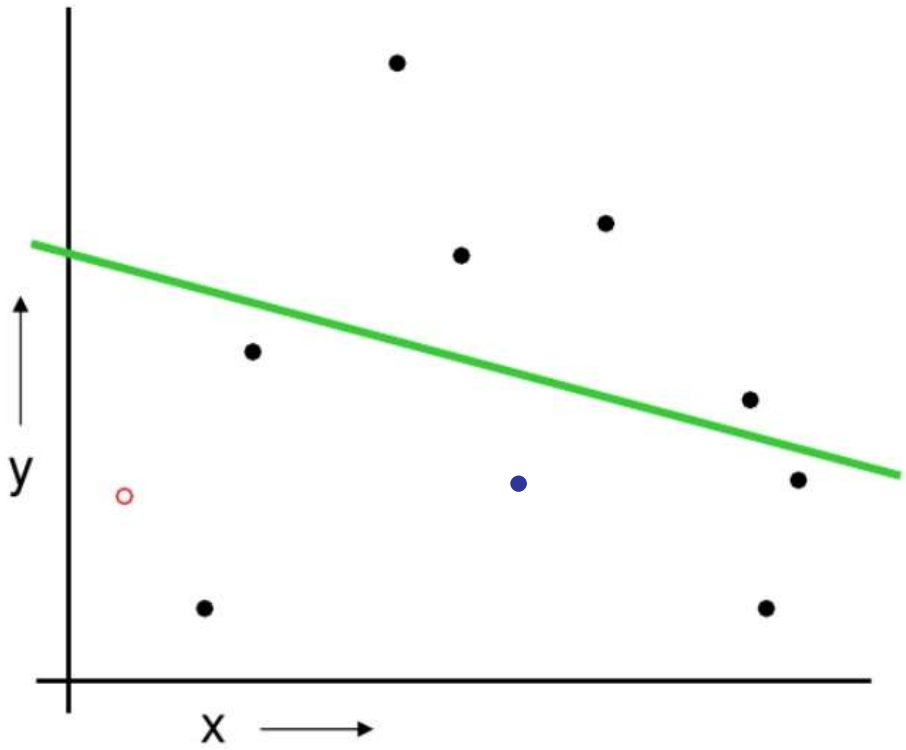
- Für k=1 bis n Datenpunkte
  1. Wähle (x_k, y_k) aus
  2. Entferne Datenpunkt (x_k, y_k) temporär aus Dataset
  3. Trainiere anhand der verbliebenen Datenpunkten

LOOCV- Leave-one-out Cross Validation



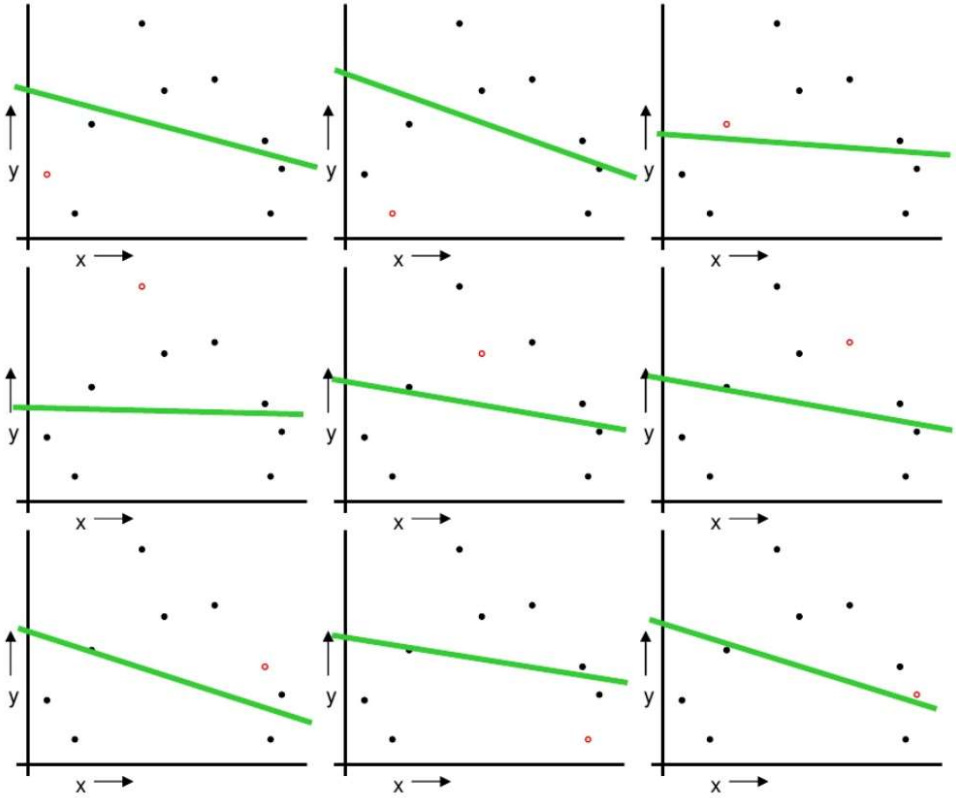
- Für k=1 bis n Datenpunkte
  1. Wähle (x_k, y_k) aus
  2. Entferne Datenpunkt (x_k, y_k) temporär aus Dataset
  3. Trainiere anhand der verbliebenen Datenpunkten
  4. Berechne Fehler von (x_k, y_k)

LOOCV- Leave-one-out Cross Validation



- Für k=1 bis n Datenpunkte
  1. Wähle (x_k, y_k) aus
  2. Entferne Datenpunkt (x_k, y_k) temporär aus Dataset
  3. Trainiere anhand der verbliebenen Datenpunkten
  4. Berechne Fehler von (x_k, y_k)
  5. Wiederhole Vorgang für alle Punkte

LOOCV- Leave-one-out Cross Validation

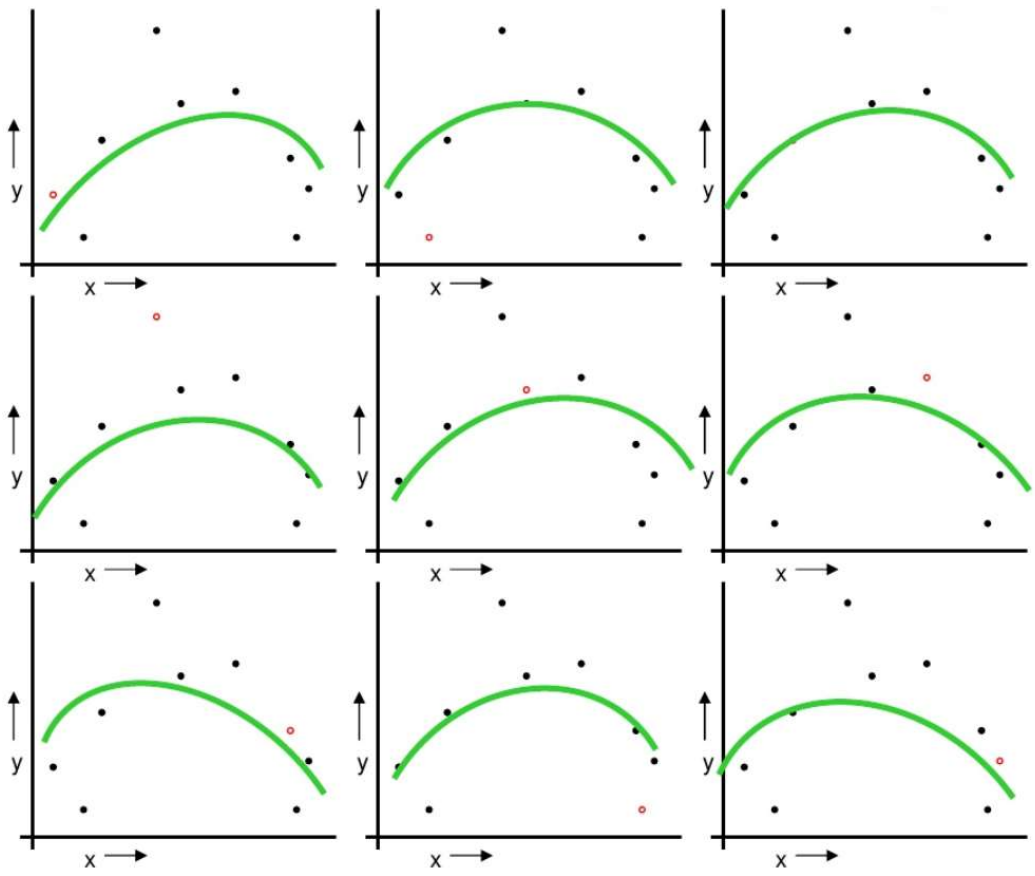


- Für k=1 bis n Datenpunkte
  1. Wähle (x_k, y_k) aus
  2. Entferne Datenpunkt (x_k, y_k) temporär aus Dataset
  3. Trainiere anhand der verbliebenen Datenpunkten
  4. Berechne Fehler von (x_k, y_k)
  5. Wiederhole Vorgang für alle Punkte
  6. Bilde durchschnittlichen Fehler für alle Punkte

     $MSE_{LOOCV} = 2.12$

LOOCV- Leave-one-out Cross Validation

Zum Vergleich: quadratische Regression

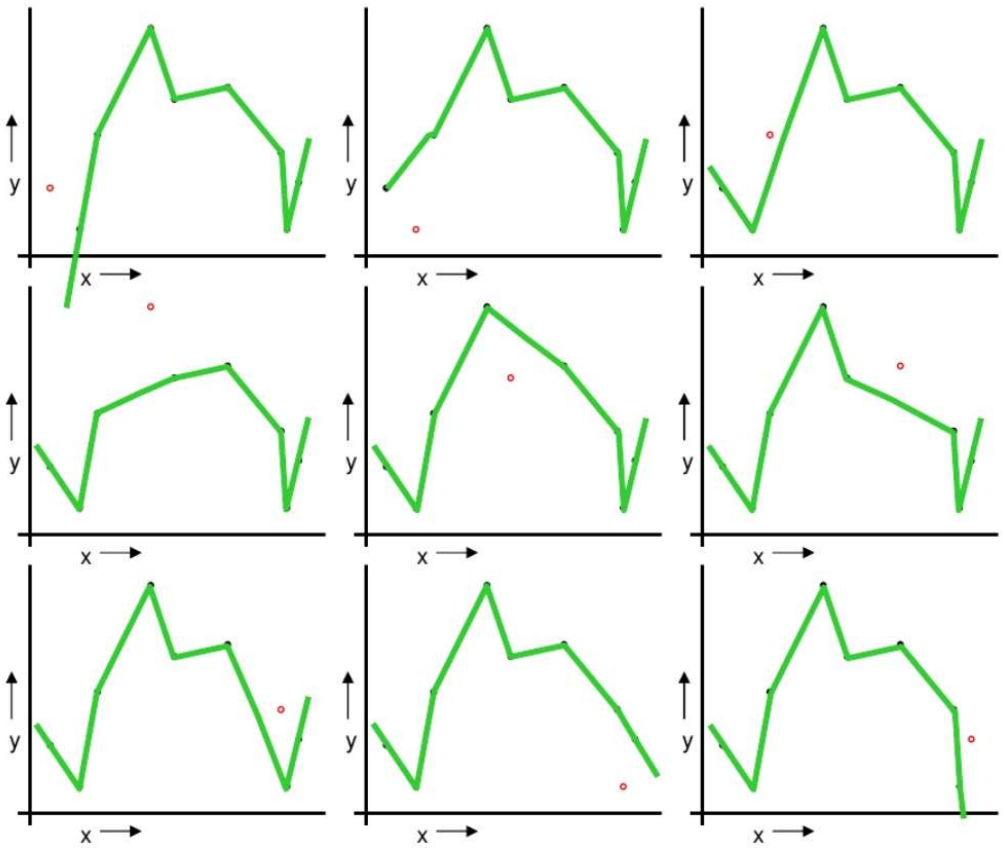


- Für k=1 bis #Datenpunkte
  1. Wähle (x_k, y_k) aus
  2. Entferne Datenpunkt (x_k, y_k) temporär aus Dataset
  3. Trainiere anhand der verbliebenen Datenpunkten
  4. Berechne Fehler von (x_k, y_k)
  5. Wiederhole Vorgang für alle Punkte
  6. Bilde durchschnittlichen Fehler für alle Punkte

  $MSE_{LOOCV} = 0.962$

LOOCV- Leave-one-out Cross Validation

Zum Vergleich: Connect the dots



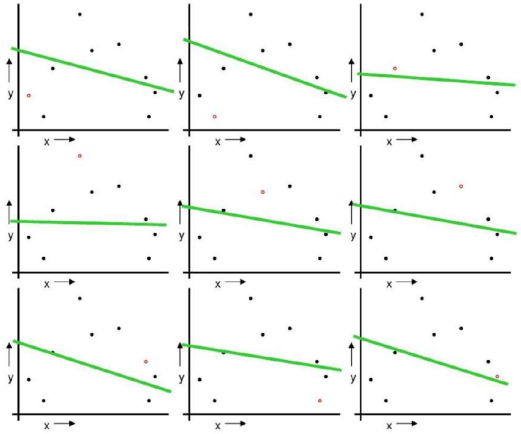
- Für k=1 bis #Datenpunkte
  1. Wähle (x_k, y_k) aus
  2. Entferne Datenpunkt (x_k, y_k) temporär aus Dataset
  3. Trainiere anhand der verbliebenen Datenpunkten
  4. Berechne Fehler von (x_k, y_k)
  5. Wiederhole Vorgang für alle Punkte
  6. Bilde durchschnittlichen Fehler für alle Punkte

  $MSE_{LOOCV} = 3.33$

LOOCV – Leave-one-out-Cross Validation

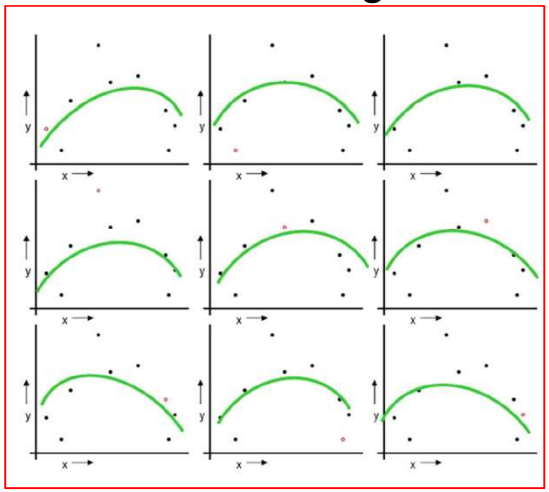
Conclusion – Endergebnis: Welches Verfahren ist zu wählen?

Lineare Regression



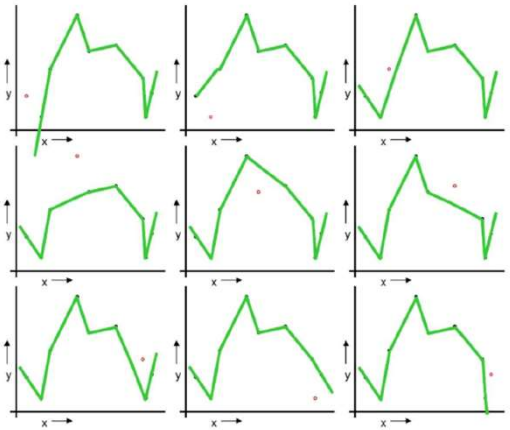
$\mathrm{MSE}_{LOOCV} = 2.12$

Quadratische Regression



$\mathrm{MSE}_{LOOCV} = 0.962$

Connect the dots



$\mathrm{MSE}_{LOOCV} = 3.33$

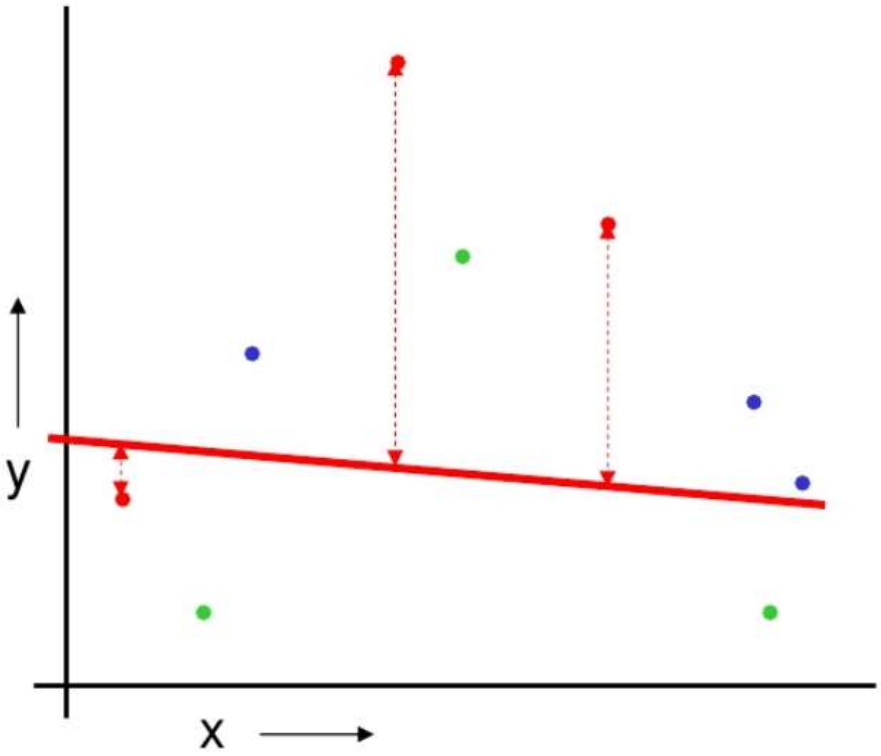
Geringster mittlerer Fehler wird mit Quadratischer Regression erreicht!

„The winner is“ → Quadratische Regression

k-fold Cross Validation

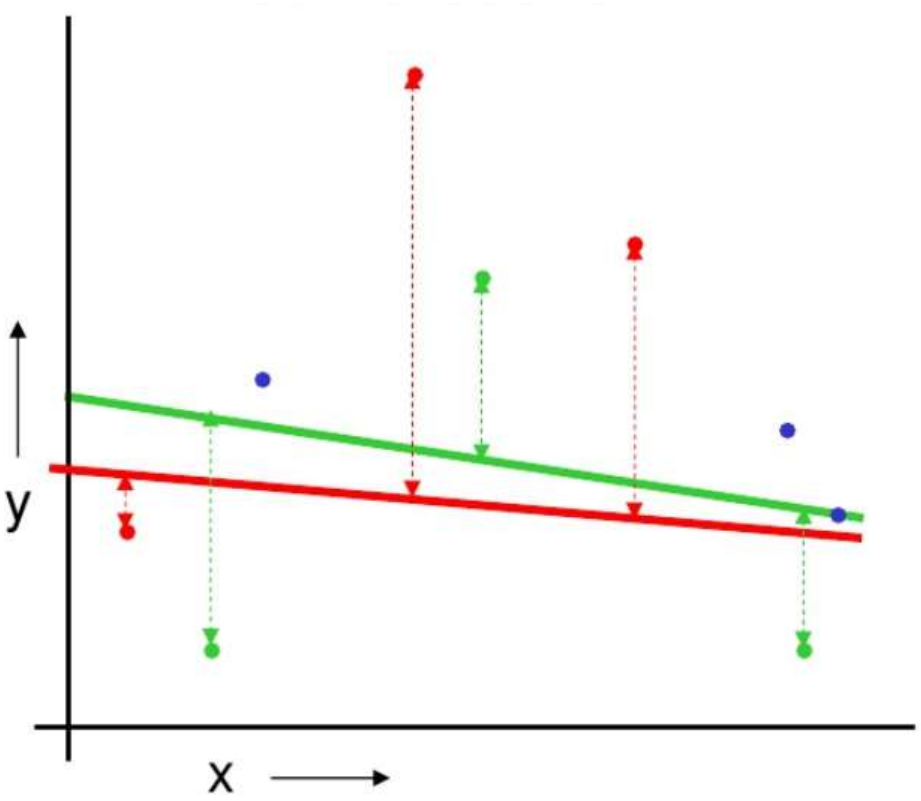
- Zur Bewertung eines Machine Learning Models wird normalerweise der Datensatz (zufällig) in Trainings- und Test-Daten aufgeteilt.
  - Man erstellt das Modell anhand von Trainingsdaten und bewertet das Modell anhand der Testdaten.
  - Allerdings ist dieses Vorgehen nicht sehr verlässlich, da es nur gegen einen (1!) Testdatensatz validiert wird.
  - K-fold Cross Validierung löst dieses Problem indem es den Datensatz in k Teile (*folds*) aufteilt und jeder Fold einmal zum Testdatensatz wird.
- Es wird im folgenden 3-fold Cross Validation vorgestellt
  - ► d.h. 2 „folds“ sind Trainingsdaten, 1 „fold“ sind Testdaten.
- Diese Aufteilung wird für das Training von 3 Modellen verwendet.
  - Es wird ein Modell durch lineare Regression erstellt und dann der MSE (mean squared error) über alle Folds für dieses Modell berechnet
  - Dann wird ein Modell durch quadratische Regression erstellt und auch der MSE berechnet
  - Dann wird „Connect the dots“ angewandt
- Ziel:
  - ► Man hat 3 Modelle erstellt und diese mithilfe des k-fold cross validation validiert.
  - ► Man ist sich also sicher(er), dass das jeweilige Modell generell gute Ergebnisse liefert.

k-fold Cross Validation



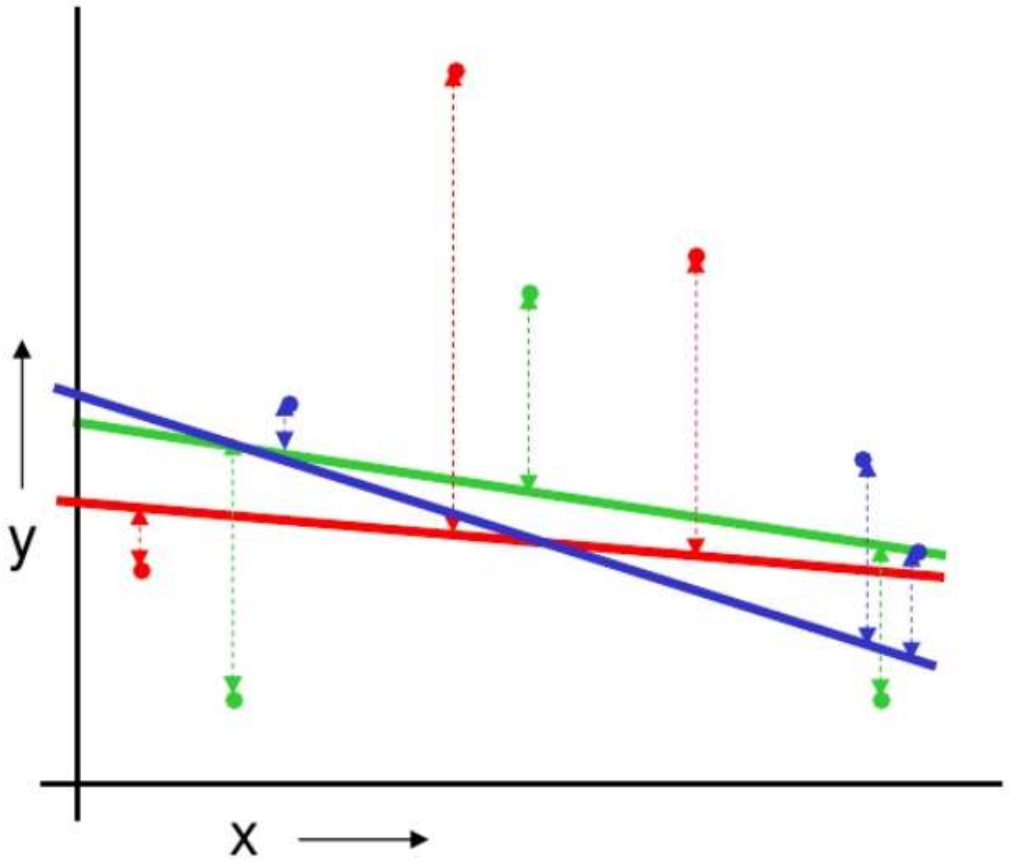
- Teile den Datensatz zufällig in k gleiche Partitionen auf (hier: k=3, rot, blau, grün)
- Rote Partition: Trainiere anhand der Datenpunkte, die in blauer und grüner Partition sind und berechne Fehler der Regression zu roten Punkten

k-fold Cross Validation



- Teile den Datensatz zufällig in k gleiche Partitionen auf (hier: k=3, rot, blau, grün)
- Rote Partition: Trainiere anhand der Datenpunkte, die in blauer und grüner Partition sind und berechne Fehler der Regression zu roten Punkten
- Grüne Partition: Trainiere anhand der Datenpunkte, die in roter und blauer Partition sind und berechne Fehler der Regression zu grünen Punkten

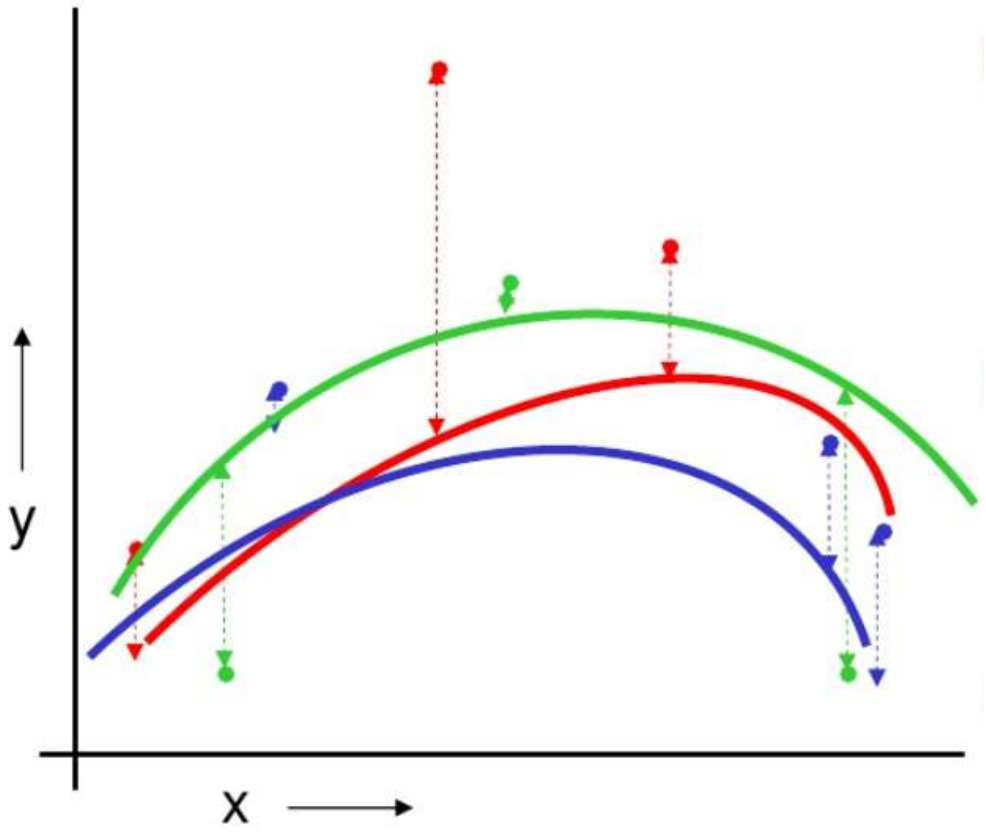
k-fold Cross Validation



- Teile den Datensatz zufällig in k gleiche Partitionen auf (hier: k=3, rot, blau, grün)
- Rote Partition: Trainiere anhand der Datenpunkte, die in blauer und grüner Partition und berechne Fehler der Regression zu roten Punkten
- Grüne Partition: Trainiere anhand der Datenpunkte, die in roter und blauer Partition sind und berechne Fehler der Regression zu grünen Punkten
- Blaue Partition: Trainiere anhand der Datenpunkte, die in roter und grüner Partition sind und berechne Fehler der Regression zu blauen Punkten
- Berechne durchschnittlichen Fehler:
  - Lineare Regression
  - MSE_{3Fold}= 2.05 (mean squared error)

k-fold Cross Validation

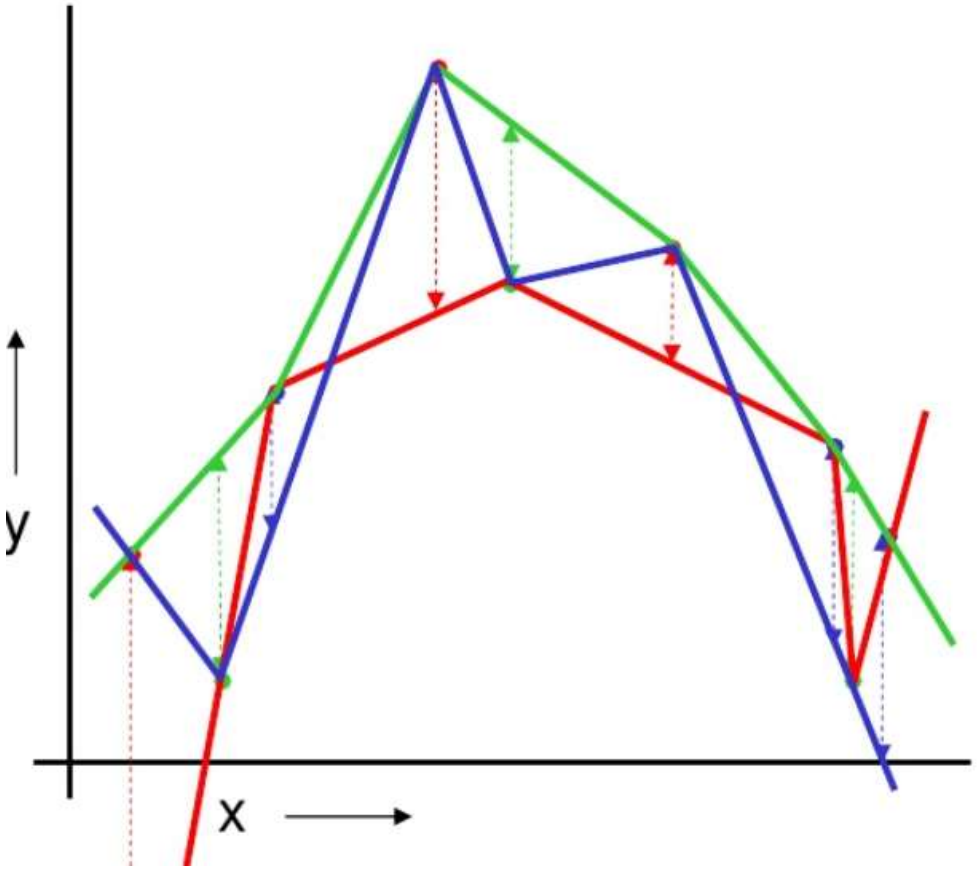
Quadratische Regression



- Teile den Datensatz zufällig in k gleiche Partitionen auf (hier: k=3, rot, blau, grün)
- Rote Partition: Trainiere anhand der Datenpunkte, die in blauer und grüner Partition und berechne Fehler der Regression zu roten Punkten
- Grüne Partition: Trainiere anhand der Datenpunkte, die in roter und blauer Partition sind und berechne Fehler der Regression zu grünen Punkten
- Blaue Partition: Trainiere anhand der Datenpunkte, die in roter und grüner Partition sind und berechne Fehler der Regression zu blauen Punkten
- Berechne durchschnittlichen Fehler:
  - Quadratische Regression
  - MSE_{3Fold}= 1.11 (mean squared error)

k-fold Cross Validation

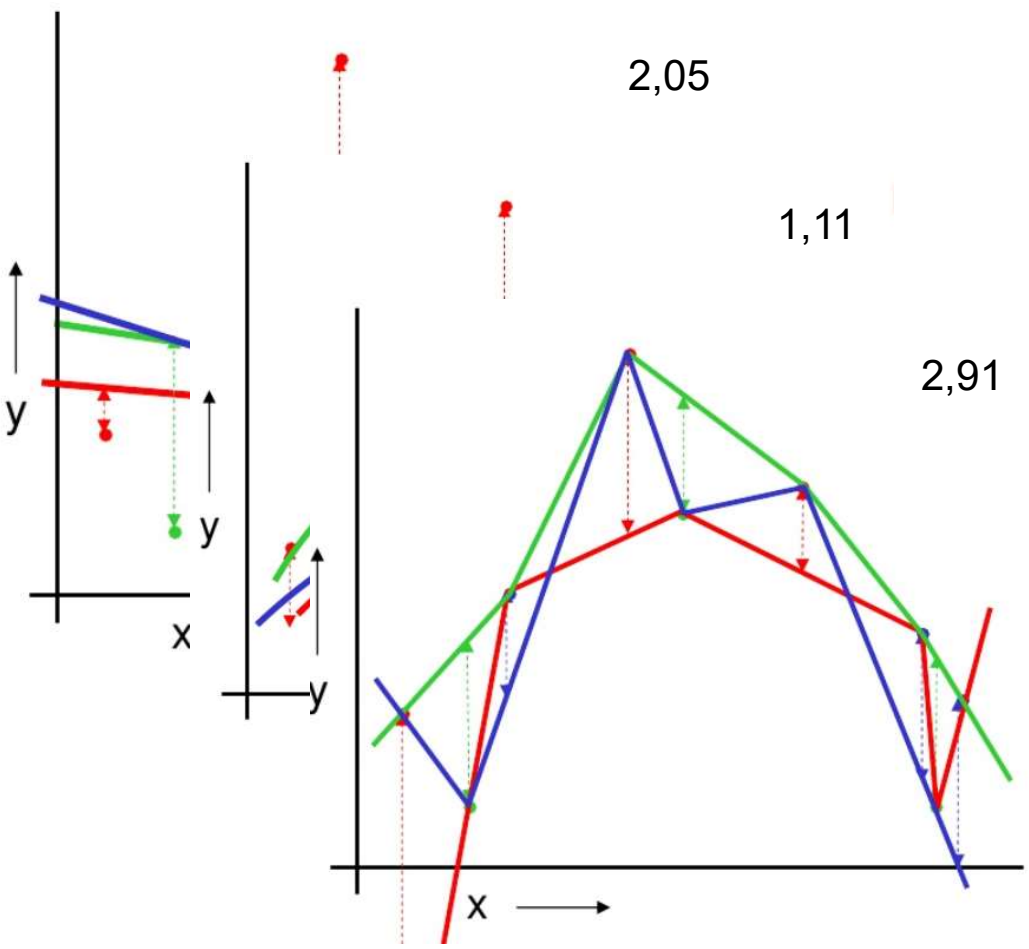
Connect the dots



- Teile den Datensatz zufällig in k gleiche Partitionen auf (hier: k=3, rot, blau, grün)
- Rote Partition: Trainiere anhand der Datenpunkte, die in blauer und grüner Partition und berechne Fehler der Regression zu roten Punkten
- Grüne Partition: Trainiere anhand der Datenpunkte, die in roter und blauer Partition sind und berechne Fehler der Regression zu grünen Punkten
- Blaue Partition: Trainiere anhand der Datenpunkte, die in roter und grüner Partition sind und berechne Fehler der Regression zu blauen Punkten
- Berechne durchschnittlichen Fehler:
  - „Connect the dots“ Interpolation
  - MSE_{3Fold}= 2.91 (mean squared error)

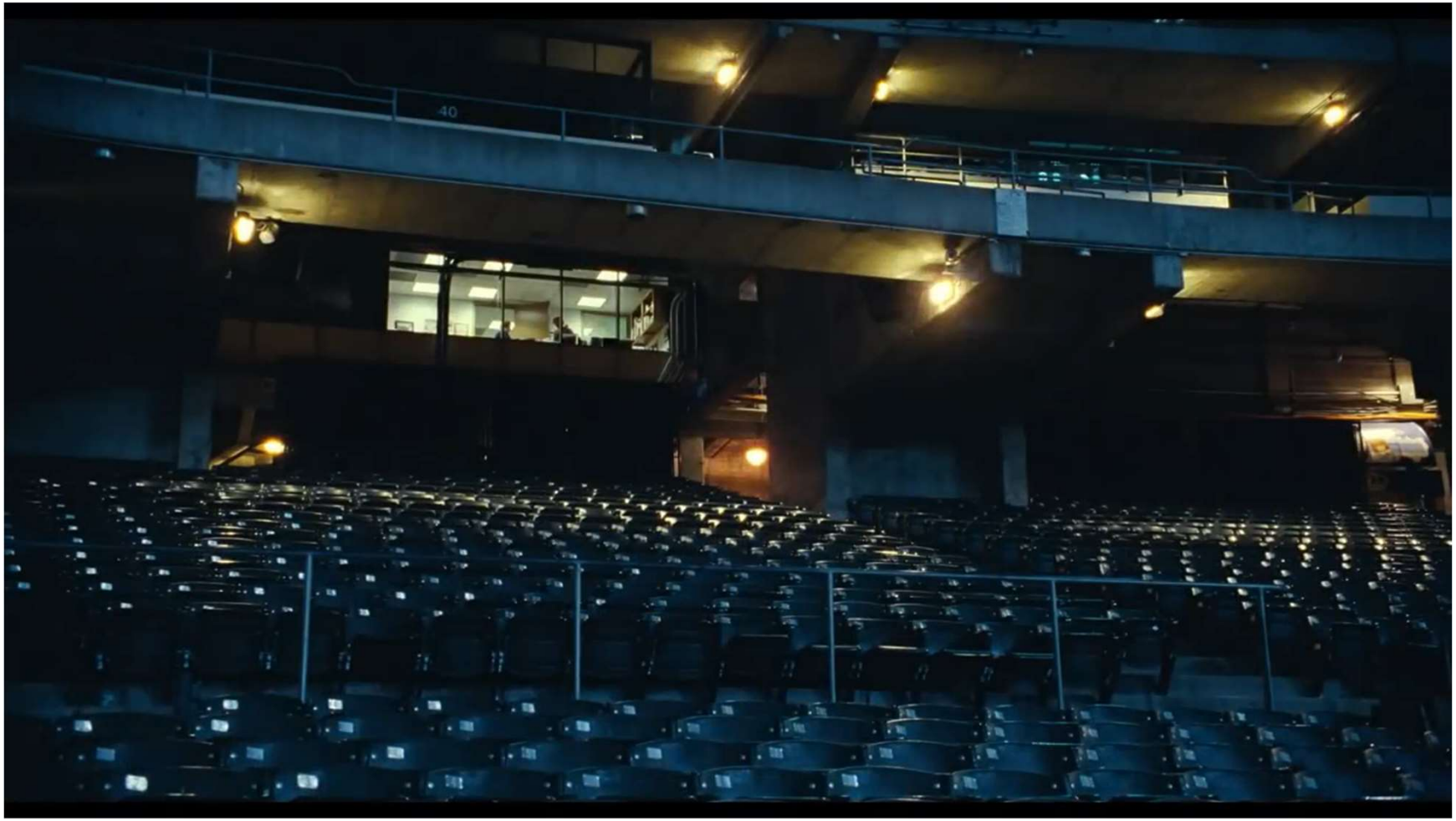
k-fold Cross Validation

Methodenauswahl



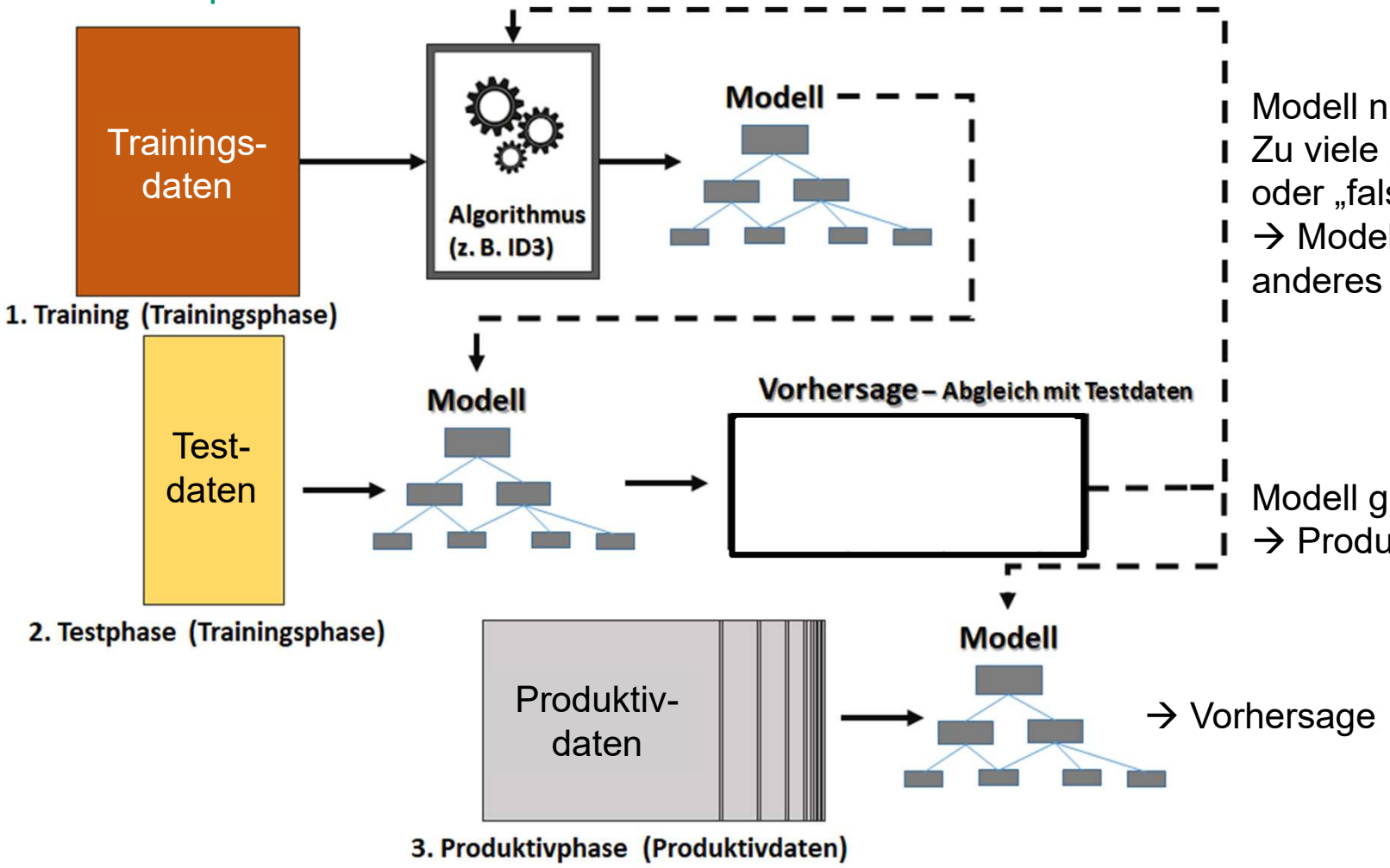
- Am Ende vergleicht man alle mittleren MSE der unterschiedlichen Methoden und entscheidet sich für die Methode mit dem kleinsten mittleren MSE (im Beispiel hier: quadratische Regression)
- Man bildet nun ein neues Modell mit allen Datenpunkten als Trainingsdatensatz.
- Im Idealfall hat man nun noch „unangetastete“ Daten zum Testen oder gar Validieren des endgültigen Modells.
- Es geht also letztlich um die Methodenauswahl.

Moneyball



Test trainierter Modelle

Train/Test-Split

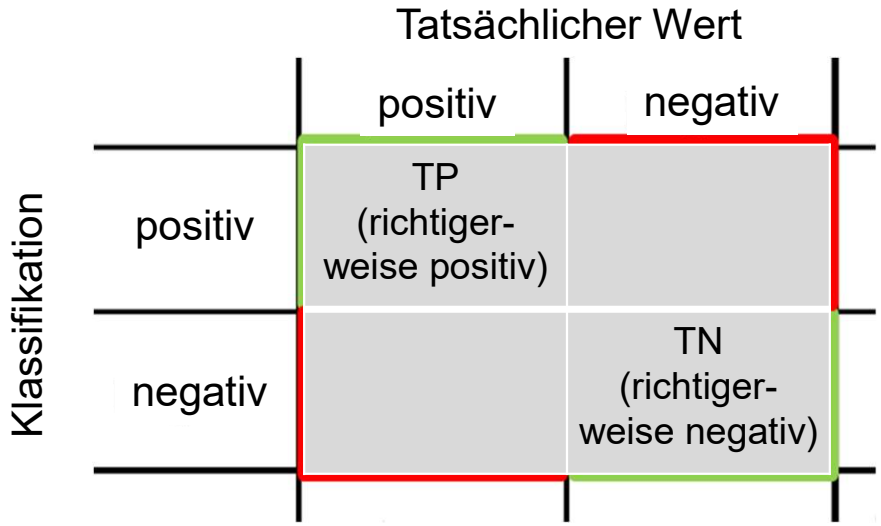


Konfusionsmatrix

Es werden 4 Kategorien für die Bewertung der Klassifikationsgüte eingeführt

- Richtig positiv (engl.: true positives):
  - Das Endprodukt entspricht den Qualitätsansprüchen, und das Klassifikationsmodell hat dies korrekt mit gut klassifiziert/vorhergesagt.
- Richtig negativ (engl.: true negatives):
  - Das Endprodukt entspricht nicht den Qualitätsansprüchen, und das Klassifikationsmodell hat dies korrekt mit schlecht klassifiziert/vorhergesagt.

- Falsch positiv (engl.: false positives):
  - Das Endprodukt entspricht nicht den Qualitätsansprüchen, aber das Klassifikationsmodell hat dies fälschlich mit gut klassifiziert.
- Falsch negativ (engl.: false negatives):
  - Das Endprodukt entspricht den Qualitätsansprüchen, aber das Klassifikationsmodell hat dies fälschlich mit schlecht klassifiziert.

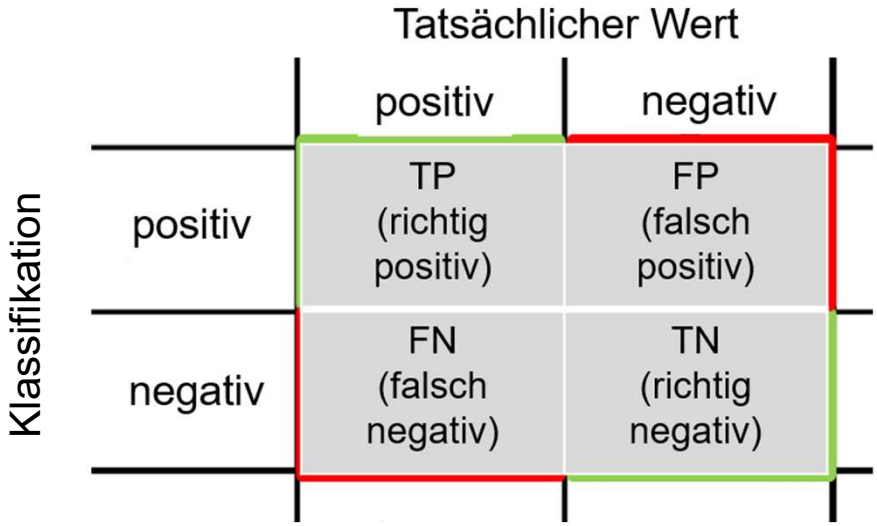


Konfusionsmatrix

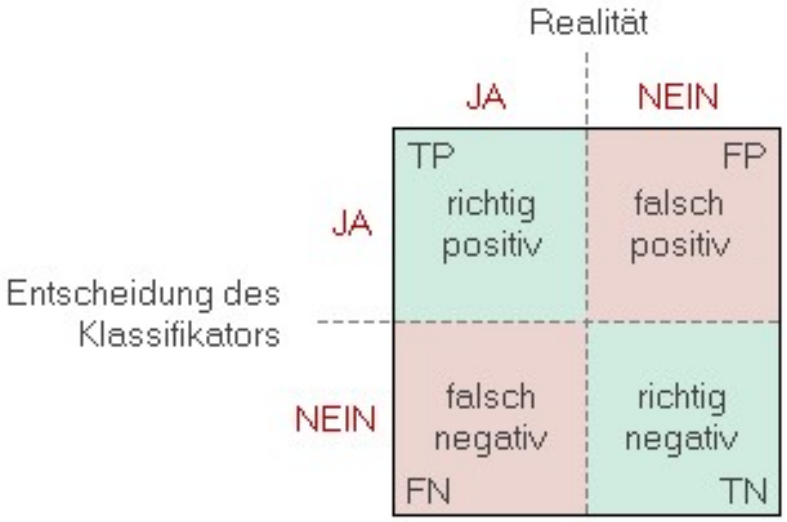
Es werden 4 Kategorien für die Bewertung der Klassifikationsgüte eingeführt

- Diese vier Bewertungskategorien werden in einer Konfusionsmatrix (engl.: *confusion matrix*) dargestellt.
  - Die Konfusionsmatrix besteht aus den Spalten, die das Klassifikationsmodell vorhersagt.
  - Die Spalten repräsentieren die tatsächlichen Werte, die in den Testdaten vorhanden sind.
  - Die Zellen «richtig positiv» (TP) und «richtig negativ» (TN) geben die Anzahl der richtig vorhergesagten Gut- und Schlecht-Werte an.
  - Die Zellen «falsch positiv» (FP) und «falsch negativ» (FN) geben die Anzahl der falschen Vorhersagen an

- Werden TP und FN miteinander summiert, so ergibt die Summe die Anzahl der tatsächlich verfügbaren Gut-Werte in dem Testdatensatz.
- Gleiches gilt für FP und TN → die Summe entspricht der Anzahl der Schlecht-Werte im Testdatensatz.



Vorsicht, in der Literatur finden sich unterschiedliche Darstellungen



Ermittelte Klasse (Array)

Tatsächliche Klasse	positiv	negativ	
positiv	TP (richtig positiv)	FN (falsch negativ)	P
negativ	FP (falsch positiv)	TN (richtig positiv)	P'
	Q	Q'	1

Konfusions-Matrix

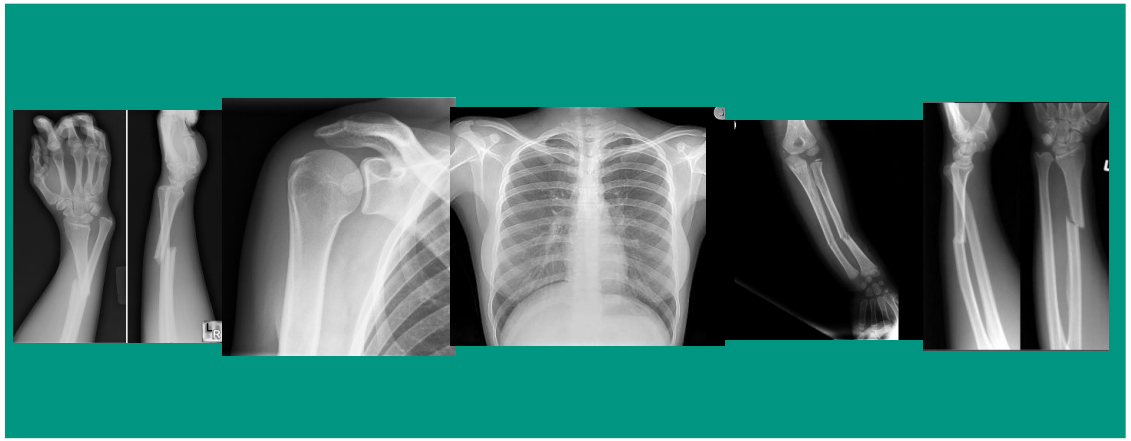
Ziel ist es die gebrochenen (*positiv*) von den gesunden Knochen zu unterscheiden.

Knochenbruch-Datenbank

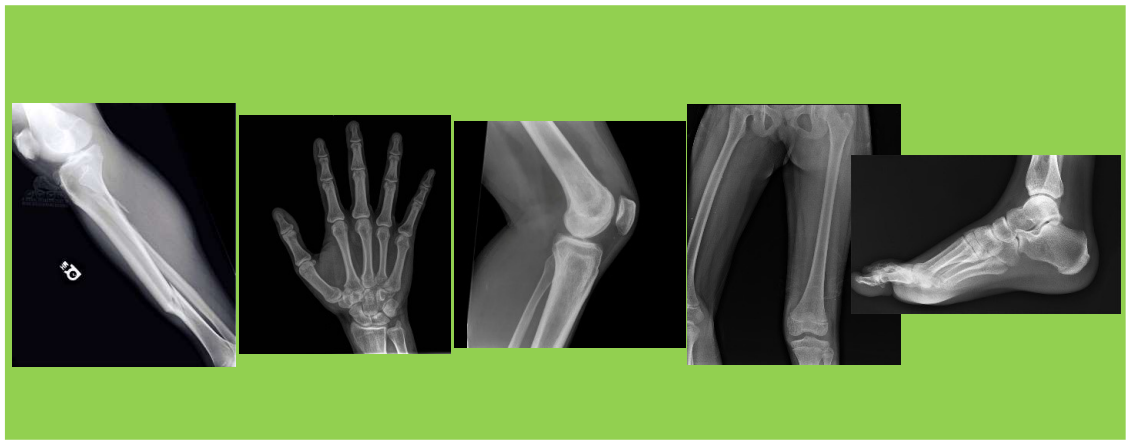
Vorhergesagte Situation

„Der Befund ist positiv“

gebrochen



gesund



Konfusions-Matrix

Beispiel: Jetzt schaut der Arzt drauf.

Knochenbruch-Datenbank

Wahre Situation

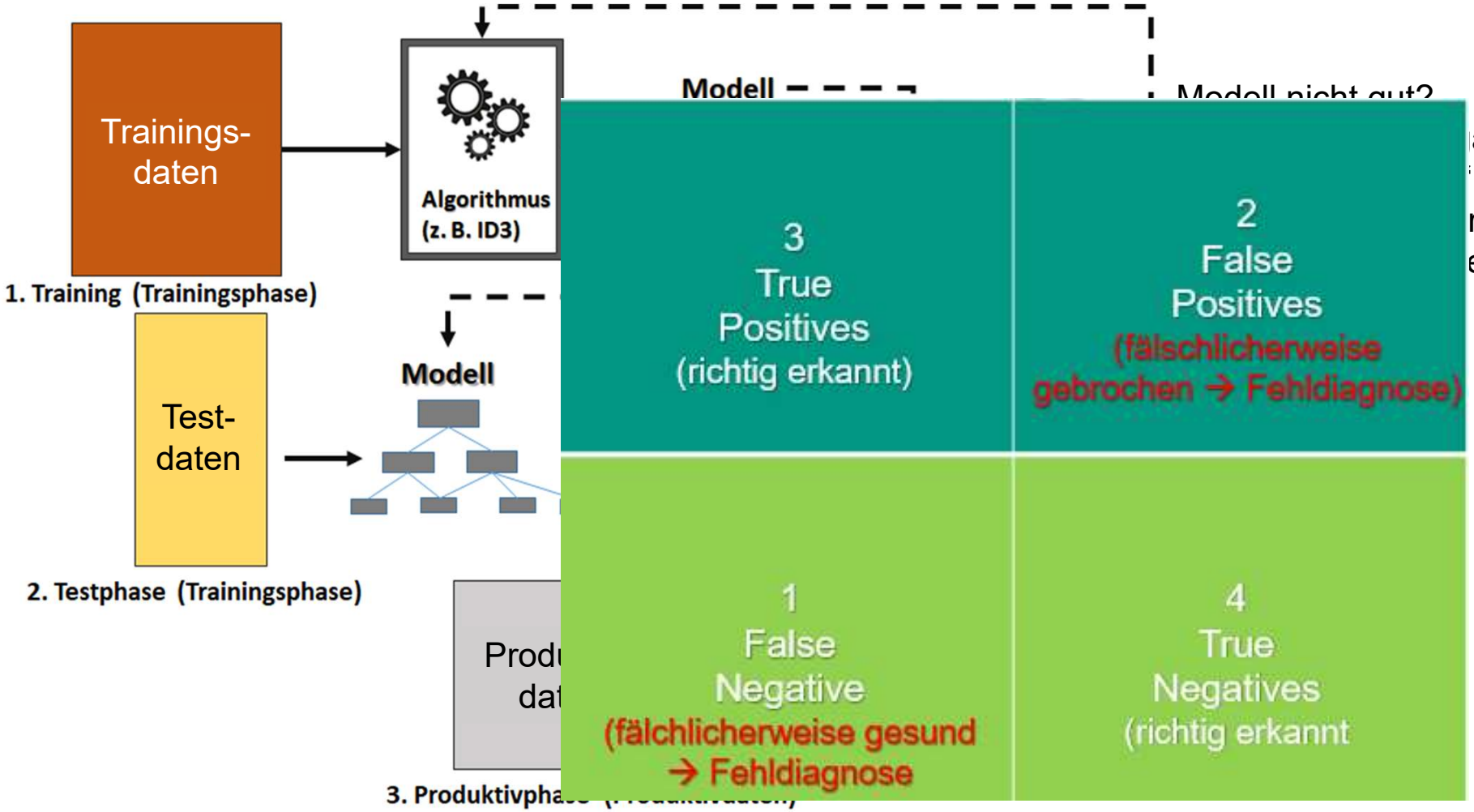
Vorhergesagte Situation	gebrochen	gesund
gebrochen	3 True Positives (richtig erkannt)	2 False Positives (fälschlicherweise gebrochen → Fehldiagnose)
gesund	1 False Negative (fälschlicherweise gesund → Fehldiagnose	4 True Negatives (richtig erkannt

Konfusions-Matrix

Knochenbruch-Datenbank

Wahre Situation

Vorhergesagte Situation	gebrochen	gesund
gebrochen	3 True Positives (richtig erkannt)	2 False Positives (fälschlicherweise gebrochen → Fehldiagnose)
gesund	1 False Negative (fälschlicherweise gesund → Fehldiagnose	4 True Negatives (richtig erkannt


Trainings-
daten
Algorithmus
(z. B. ID3)
Modell
1. Training (Trainingsphase)
Test-
daten
Modell
2. Testphase (Trainingsphase)
Prod
dat
3. Produktivpha
3
True
Positives
(richtig erkannt)
2
False
Positives
(fälschlicherweise
gebrochen → Fehldiagnose)
1
False
Negative
(fälschlicherweise gesund
→ Fehldiagnose
4
True
Negatives
(richtig erkannt

Metriken

- Die Richtig-positiv-Rate (auch Sensitivität, Empfindlichkeit oder Trefferquote) gibt den Anteil der korrekt als positiv klassifizierten Objekte an der Gesamtheit der tatsächlich positiven Objekte an (True Positive Rate TPR).
- Die Falsch-negativ-Rate (englisch *false negative rate* oder *miss rate*) gibt den Anteil der fälschlich als negativ klassifizierten Objekte an der Gesamtheit der tatsächlich positiven Objekte an (False negative rate FNR).
- Übrig bleiben
  - Specificity, Selectivity (True negative rate TNR).
  - Fall-out (False positive rate FPR).

$$\mathrm{TPR} = \frac{\sum \text{True positives}}{\sum \text{Real positive}}$$

$$\mathrm{FNR} = \frac{\sum \text{False negatives}}{\sum \text{Real positive}}$$

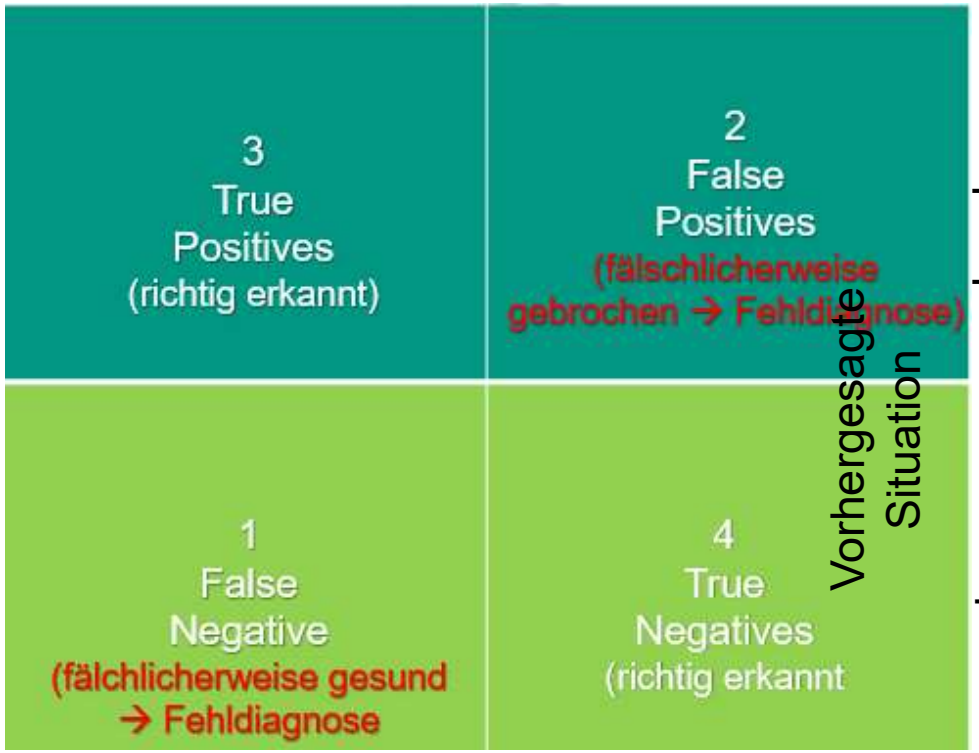
$$\mathrm{TNR} = \frac{\sum \text{True negatives}}{\sum \text{Real negative}}$$

$$\mathrm{FPR} = \frac{\sum \text{False positives}}{\sum \text{Real negative}}$$

- Da sich beide Maße auf den Fall beziehen, dass in Wirklichkeit die positive Kategorie vorliegt (erste Spalte der Wahrheitsmatrix), addieren sich die Sensitivität und die Falsch-negativ-Rate zu 1 bzw. 100 %.

Konfusions-Matrix

„Knochenbrüche“



Wahre Situation

Vorhergesagte Situation	gebrochen	gesund
gebrochen	$\mathrm{TPR} = \frac{\sum \text{True positives}}{\sum \text{Condition positive}}$ Sensitivity, Recall, True positive rate(TPR).	$\mathrm{FPR} = \frac{\sum \text{False positives}}{\sum \text{Condition negative}}$ Fall-out, False positive rate(FPR).
gesund	$\mathrm{FNR} = \frac{\sum \text{False negatives}}{\sum \text{Condition positive}}$ Miss rate, False negative rate(FNR).	$\mathrm{TNR} = \frac{\sum \text{True negatives}}{\sum \text{Condition negative}}$ Specificity, Selectivity, True negative rate(TNR).

Konfusions-Matrix

Beispiel „165 vermeintliche Knochenbrüche“

- Untersuchte Fälle: 165
- Wirklich gebrochen: 105
- Wirklich nicht gebrochen: 60
- Vorhergesagt gebrochen: 110
- Vorhergesagt nicht gebrochen: 55

Vorhergesagte Situation \ Wahre Situation	gebrochen	gesund
gebrochen	$\mathrm{TPR} = \frac{\sum \text{True positives}}{\sum \text{Condition positive}}$ Sensitivity, Recall, True positive rate(TPR).	$\mathrm{FPR} = \frac{\sum \text{False positives}}{\sum \text{Condition negative}}$ Fall-out, False positive rate(FPR).
gesund	$\mathrm{FNR} = \frac{\sum \text{False negatives}}{\sum \text{Condition positive}}$ Miss rate, False negative rate(FNR).	$\mathrm{TNR} = \frac{\sum \text{True negatives}}{\sum \text{Condition negative}}$ Specificity, Selectivity, True negative rate(TNR).

Konfusions-Matrix

Beispiel „Knochenbrüche“

- Untersuchte Fälle: 165
- Wirklich gebrochen: 105
- Wirklich nicht gebrochen: 60
- Vorhergesagt gebrochen: 110
- Vorhergesagt nicht gebrochen: 55

- TPR: 100/105 = 0,952
- FNR: 5/105 = 0,048

= 1

- FPR: 10/60 = 0,167
- TNR: 50/60 = 0,833

= 1

Vorhergesagte Situation

Wahre Situation

Vorhergesagte Situation \ Wahre Situation	gebrochen	gesund
gebrochen	$\mathrm{TPR} = \frac{\sum \text{True positives}}{\sum \text{Condition positive}}$ 100	$\mathrm{FPR} = \frac{\sum \text{False positives}}{\sum \text{Condition negative}}$ 10
gesund	$\mathrm{FNR} = \frac{\sum \text{False negatives}}{\sum \text{Condition positive}}$ 5	$\mathrm{TNR} = \frac{\sum \text{True negatives}}{\sum \text{Condition negative}}$ 50

… und wieder verdreht

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Aktuelle Projektarbeit

Ko

Wo

- L
- Z u
- A g v
- F s d s

Zie Ab Er

Daiml

Messa

Identisch

Daimler Buses

Übersicht

Künstliche Intuitio

Unüberwach

Alle ML-Techniker Annotation

Reduzierung statistischer Merkmale

Ein-Kla Klassifik (Anomalie-E

Daimler Buses

Anomalie-Erkennung: Ein-Klassen-Erkennung

Beispiel: Time Embedding Isolation Forest (TE-IF)

- Öffnen GTF

Erkannte Fehler	Nicht erkannte Fehler
1,3%	98,7%
Gut, als Fehler erkannt	Gut, als gut erkannt
0,6%	99,4%

- Schließen GTF

Erkannte Fehler	Nicht erkannte Fehler
7,69%	92,31%
Gut, als Fehler erkannt	Gut, als gut erkannt
0,6%	99,4%

- Öffnen STD

Erkannte Fehler	Nicht erkannte Fehler
85,1%	14,9%
Gut, als Fehler erkannt	Gut, als gut erkannt
1,94%	98,06%

- Schließen STD

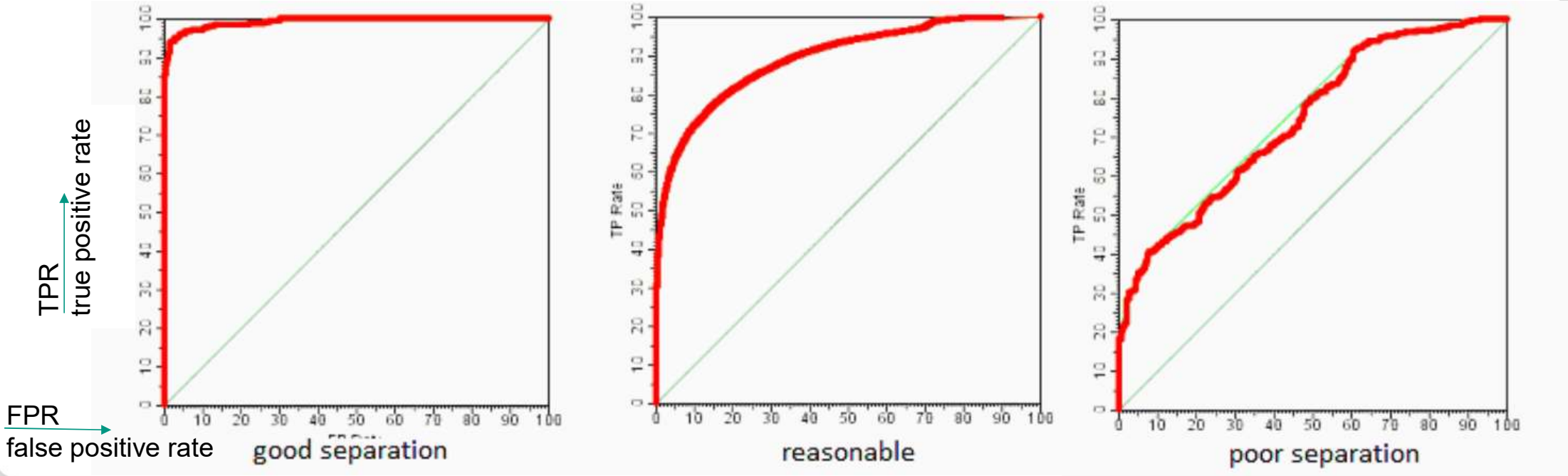
Erkannte Fehler	Nicht erkannte Fehler
19,15%	80,85%
Gut, als Fehler erkannt	Gut, als gut erkannt
2,5%	97,5%

Daimler Buses 15

Gütemaße für die Evaluation von Modellen

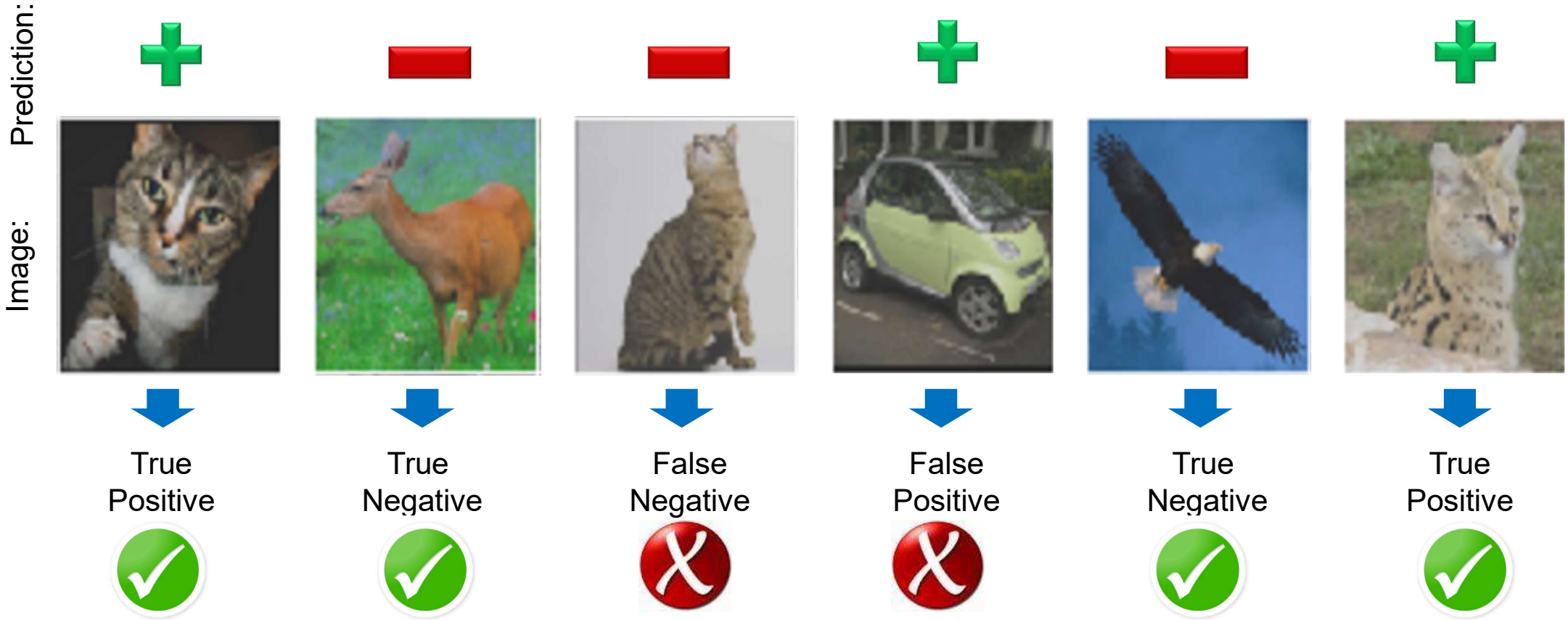
ROC (Receiver Operating Characterstic)-Kurve

- Die **ROC-Kurve** oder Grenzwertoptimierungskurve …
  - … ist eine Methode zur Bewertung und Optimierung von Analyse-Strategien.
  - … stellt visuell die Abhängigkeit der Effizienz mit der Fehlerrate für verschiedene Parameterwerte dar.
  - … plottet die *True Positives* (auf der y-Achse) gegen die *False Positives* (auf der x-Achse)



Gütemaße für die Evaluation von Modellen

Klassifikation am Beispiel „Katzen“



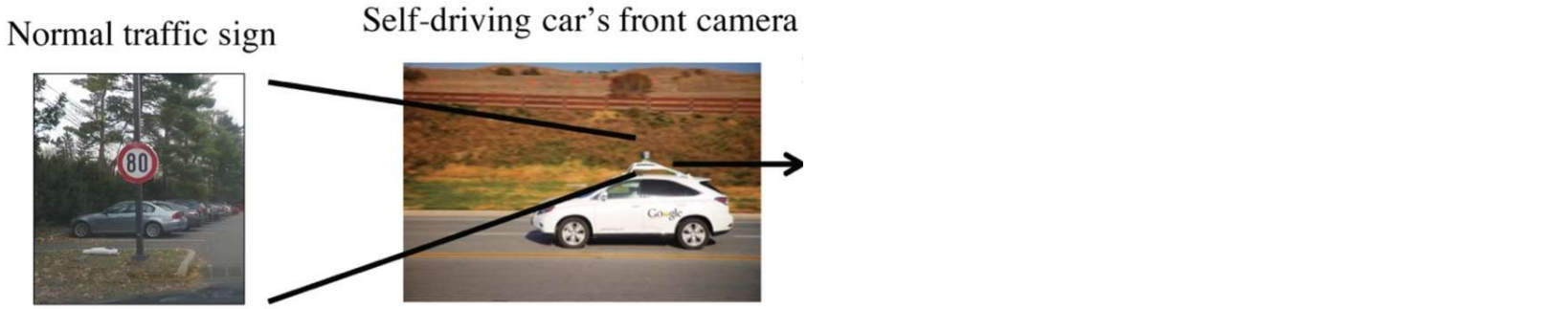
Images from the STL-10 dataset

Beispiel
„False Prediction“

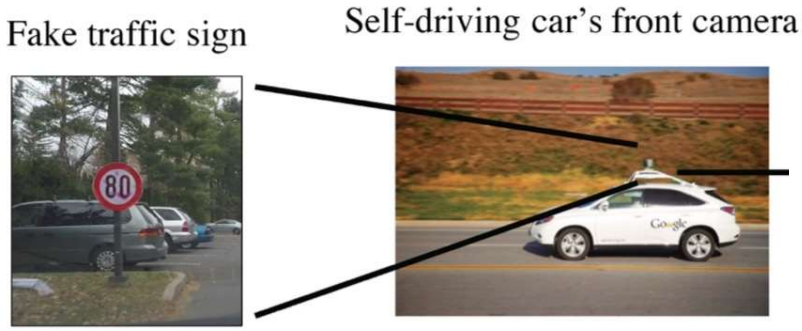
- Autonome Autos orientieren sich unter anderem ebenso wie menschliche Fahrer an Verkehrszeichen.
  - Dabei können sie in böswilliger Absicht in die Irre geführt werden.
  - Das so genannte *Deceiving Autonomous caRs with Toxic Signs (DARTS)* kann zu katastrophalen Folgen führen.
- Augenscheinlich unverdächtig
  - Verkehrs- oder andere Schilder, die für normale Beobachter unverdächtig aussehen, können aus der Sicht eines automatischen Bilderkennungsystems eine andere Bedeutung haben.
  - So haben Forscher beispielsweise das Verkehrszeichen für ein Tempolimit von 80 km/h mit Hilfe eines Musters aus dem ersten Anschein nach unverdächtigen Flecken so modifiziert, dass es ein Zeichenerkennungssystem als Stoppschild interpretieren würde.
  - Auf offener Strecke könnte ein autonomes Auto plötzlich stoppen und so einen Auffahrunfall provozieren.

Augenscheinlich unverdächtig

https://www.heise.de/newsticker/meldung/Autonome-Autos-Forscher-fuehren-Bilderkennung-mit-manipulierten-Schildern-in-die-Irre-3974483.html

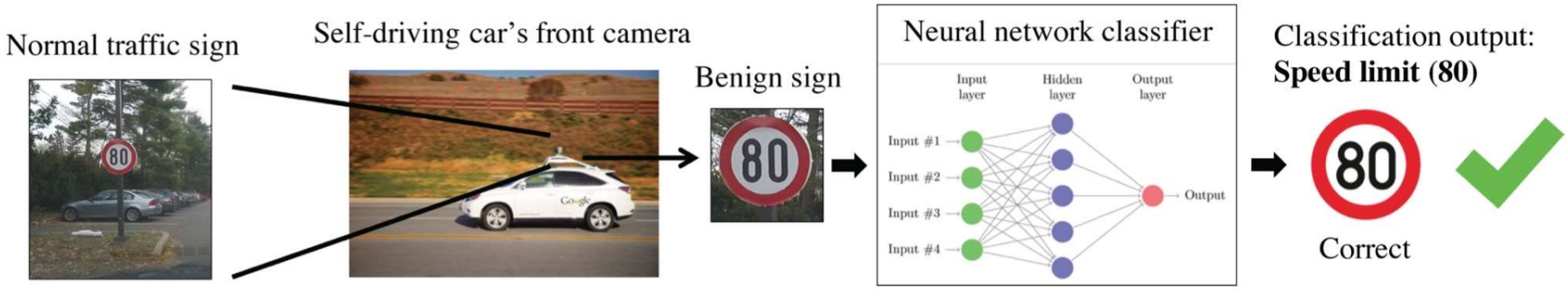


(a) Operation of the computer vision subsystem of an AV under *benign conditions*

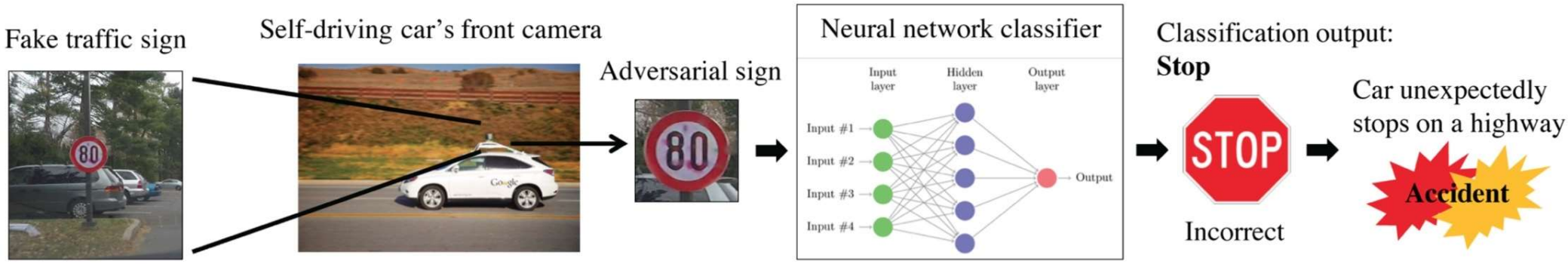


Augenscheinlich unverdächtig

https://www.heise.de/newsticker/meldung/Autonome-Autos-Forscher-fuehren-Bilderkennung-mit-manipulierten-Schildern-in-die-Irre-3974483.html

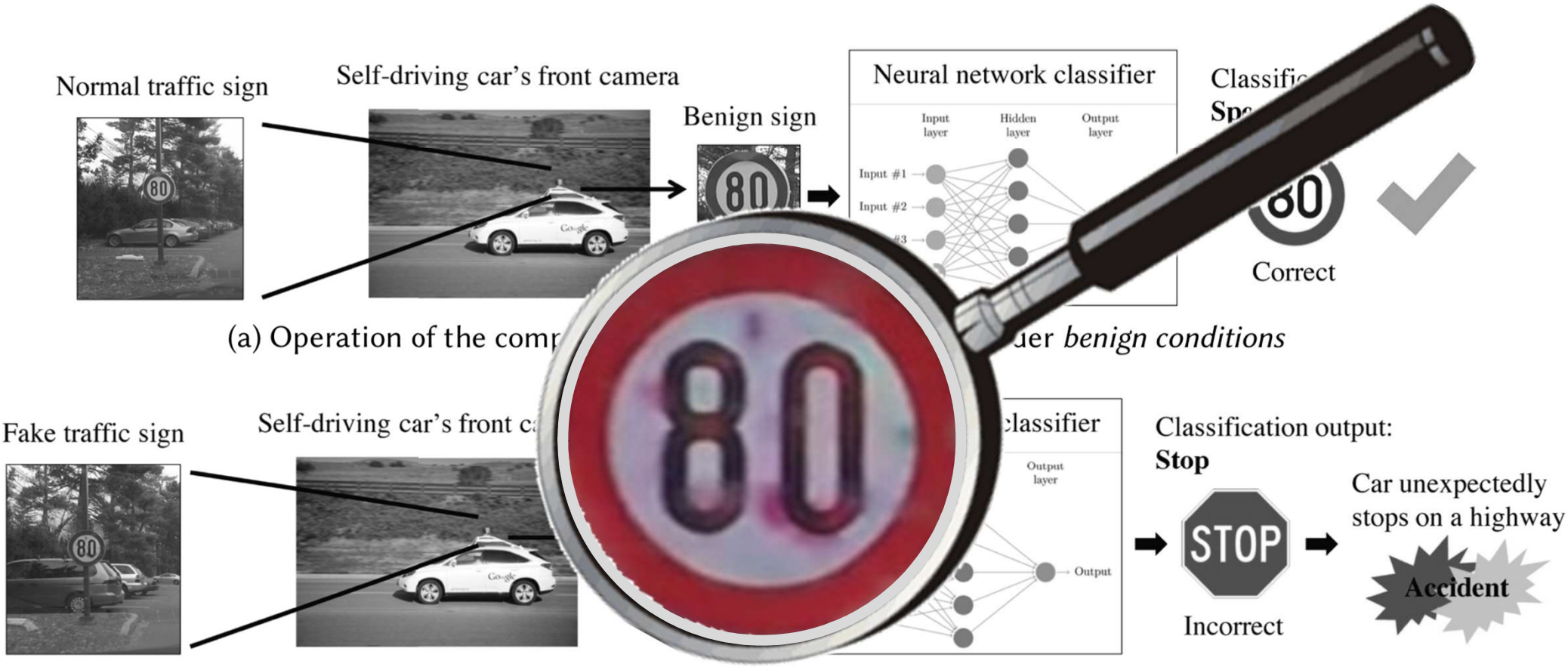


(a) Operation of the computer vision subsystem of an AV under *benign conditions*



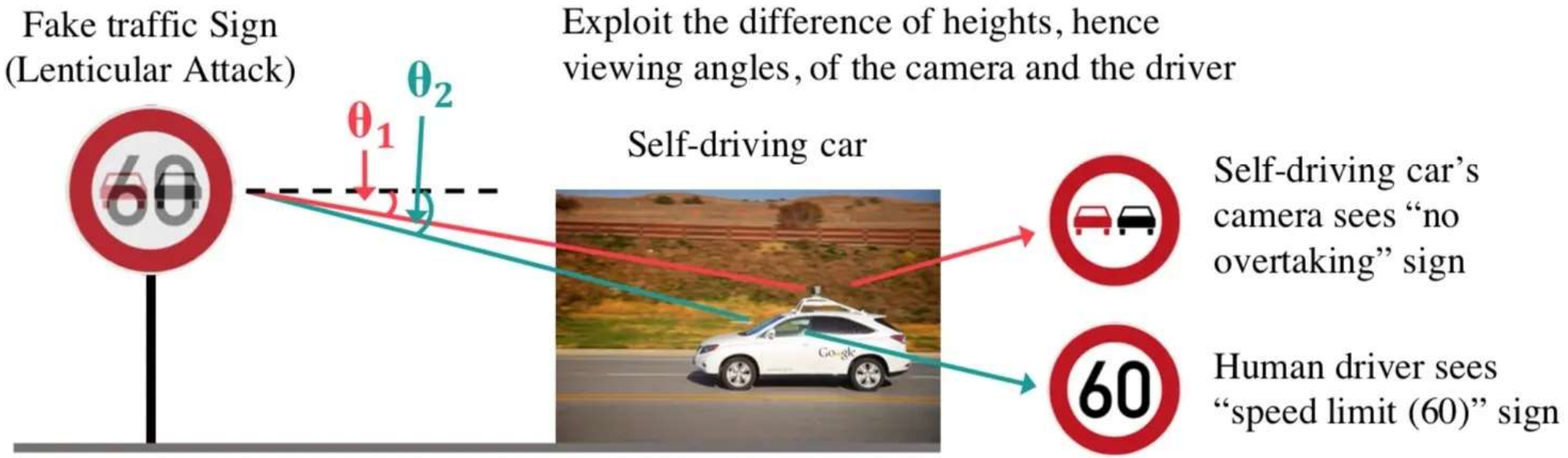
Augenscheinlich unverdächtig

https://www.heise.de/newsticker/meldung/Autonome-Autos-Forscher-fuehren-Bilderkennung-mit-manipulierten-Schildern-in-die-Irre-3974483.html



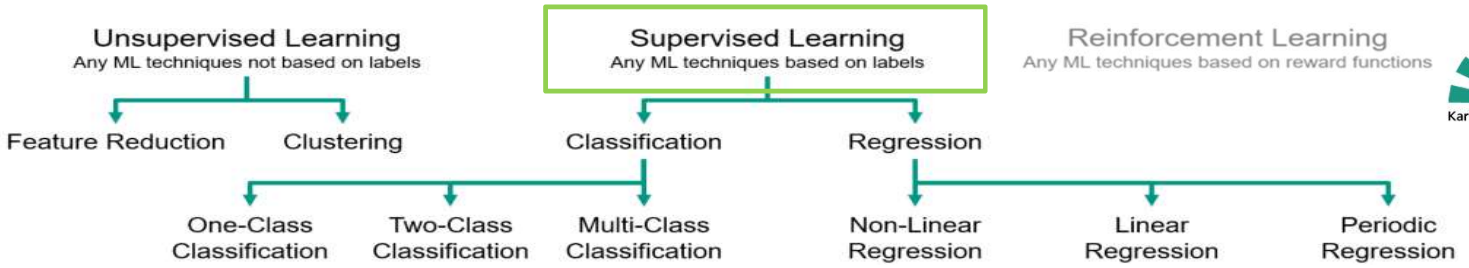
Überholverbot statt Tempolimit

Irreführung von Fahrer und Kamera mit Hilfe einer Linsenrasterfolie



(b) Principle behind an attack based on lenticular printing.

Machinelles Lernen

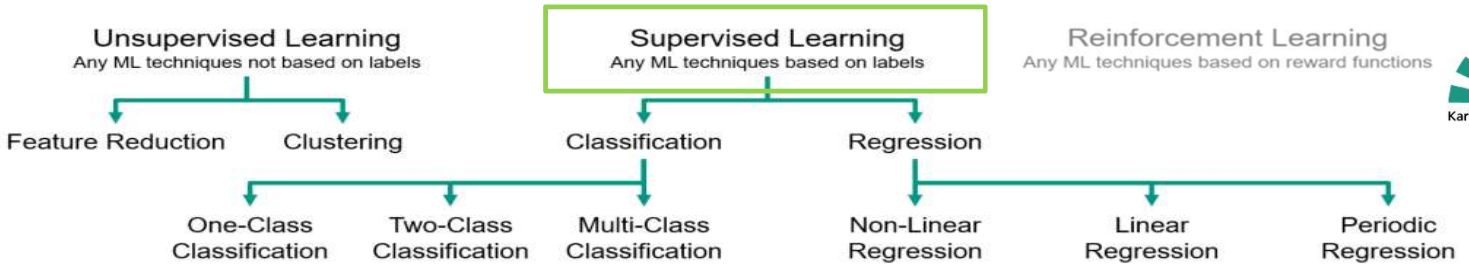


Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Klassifizierung – Algorithmen

- Im Gegensatz zu Regressionsproblemen erkennt man Klassifikationsprobleme daran, dass der Output y nur wenige diskrete Werte annehmen kann.
  - Meistens liegen diese Werte in qualitativer Form vor, beispielsweise, wenn es darum geht auf der Basis von mehreren erklärenden Variablen zu bestimmen, ob es sich bei einer E-Mail um Spam oder keinen Spam handelt.
  - In diesem Beispiel wären die erklärenden Variablen dann x und der Output y wäre 1, wenn es sich um eine Spam E-Mail handelt und 0, wenn keine Spam E-Mail vorliegt.
- Man unterscheidet zudem zwischen *one-class*, *two-class* und Klassifikationsproblemen, bei denen *multiple* Klassen vorliegen.
  - Ein Beispiel hierfür wäre zu klassifizieren von welcher Farben ein Bleistift ist.
  - Die Klassen sind in diesem Fall die rechts stehenden Farben.

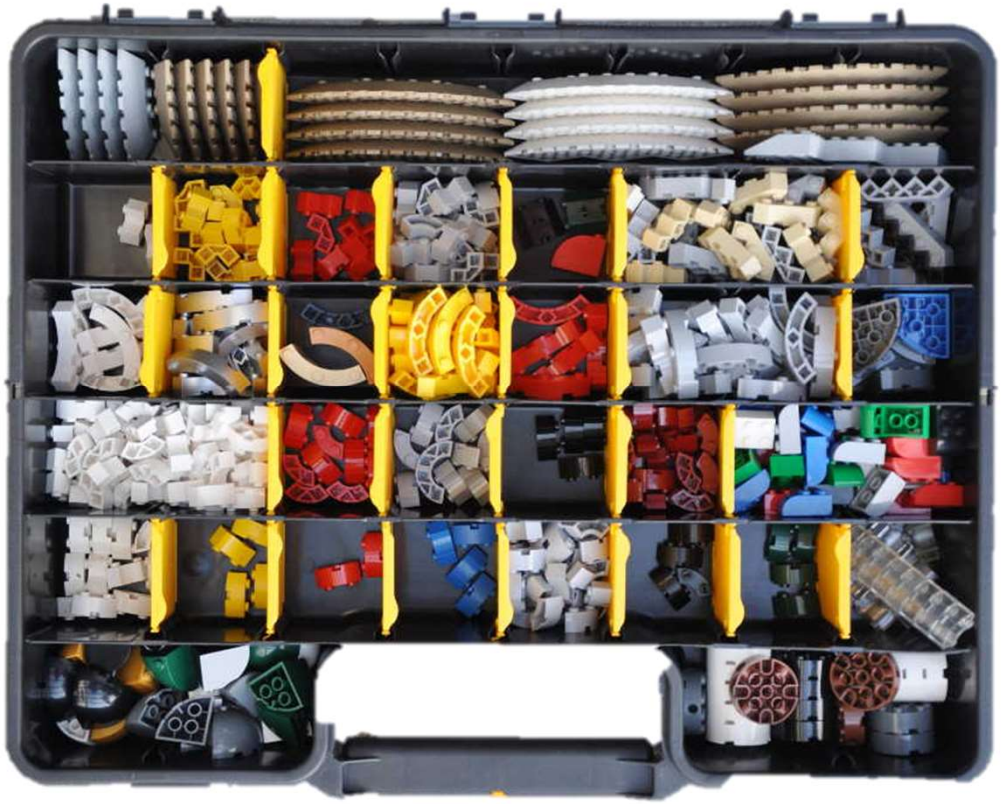
Machinelles Lernen



Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Diskriminanzanalyse
Klassifikation

- Die Diskriminanzanalyse ist ein Verfahren zur Analyse von Gruppenunterschieden.
  - Es werden die „diskriminierenden“ Variablen erkannt, die für die Gruppenzugehörigkeit sorgen.
  - Dadurch kann anhand der vorliegenden Ausprägung der unabhängigen Variablen auf die abhängigen Variable geschlossen werden (z. B. bei der Kreditwürdigkeitsprüfung).
- Die Diskriminanzanalyse erstellt ein Vorhersagemodell für Gruppenzugehörigkeiten.



Diskriminanzanalyse
Klassifikation

- Das Vorhersagemodell besteht aus einer Diskriminanzfunktion (oder bei mehr als zwei Gruppen einem Set von Diskriminanzfunktionen), die auf der Grundlage derjenigen linearen Kombinationen der Prädiktorvariablen bestimmt wird, die die beste Diskriminanz zwischen den Gruppen ergeben.
  - *Die zu untersuchenden Variablen in einem Experiment (die gemessen und beobachtet werden) werden als Antwortvariablen bzw. abhängige Variablen bezeichnet.*
  - *Andere Variablen im Experiment, die die Antwortvariable beeinflussen und vom Versuchsleiter festgelegt oder gemessen werden können, werden als Prädiktorvariablen, erklärende Variablen oder unabhängige Variablen bezeichnet.*
- Die Funktionen werden aus einer Stichprobe der Fälle generiert, bei denen die Gruppenzugehörigkeit bekannt ist.
  - Diese Funktionen können dann auf neue Fälle mit Werten für die Prädiktorvariablen zur Bestimmung der Gruppenzugehörigkeit angewandt werden.

Diskriminanzanalyse

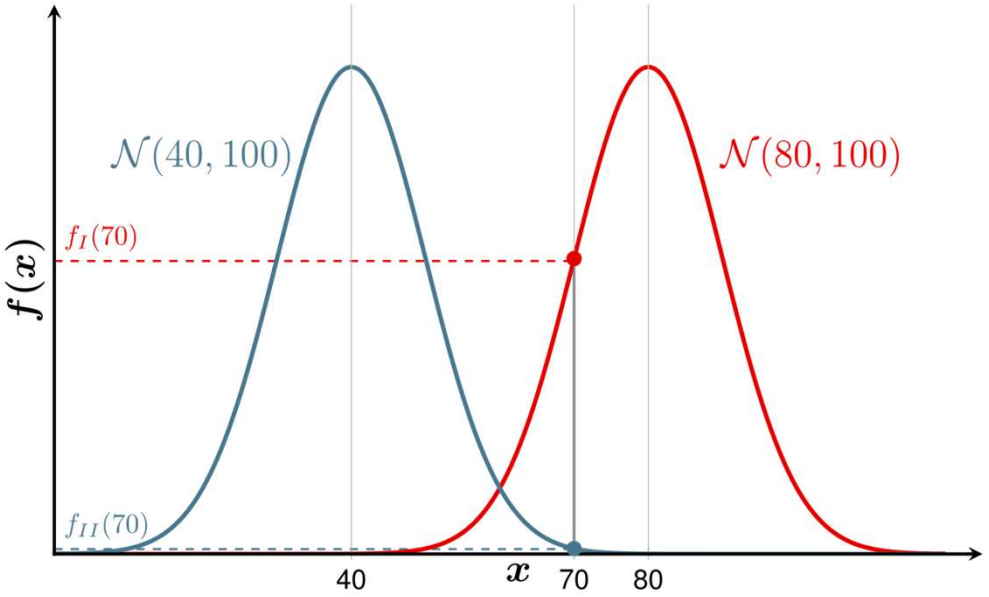
Beispiel

- Eine Gärtnerei hat die Möglichkeit, eine größere Menge Samen einer bestimmten Sorte Nelken günstig zu erwerben.
  - Um den Verdacht auszuräumen, dass es sich dabei um alte, überlagerte Samen handelt, wird eine Keimprobe gemacht.
  - Man sät also 1g Samen aus und zählt, wie viele dieser Samen keimen.
  - Aus Erfahrung ist bekannt,
    - dass die Zahl der keimenden Samen pro 1g Saatgut annähernd normalverteilt ist.
    - Bei frischem Saatgut (Population I) keimen im Durchschnitt 80 Samen, bei altem (Population II) sind es nur 40 Samen.

Diskriminanzanalyse

Beispiel: Set von Diskriminanzfunktionen durch Normalverteilung

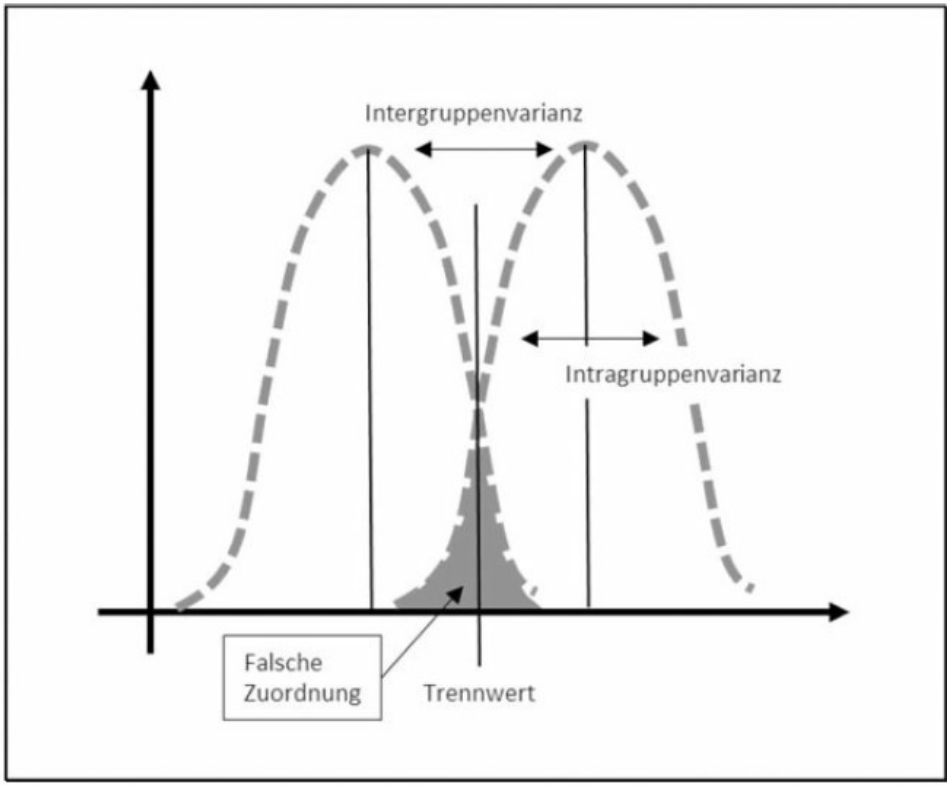
- Eine Gärtnerei hat die Möglichkeit, eine größere Menge Samen einer bestimmten Sorte Nelken günstig zu erwerben.
  - Um den Verdacht auszuräumen, dass es sich dabei um alte, überlagerte Samen handelt, wird eine Keimprobe gemacht.
  - Man sät also 1g Samen aus und zählt, wie viele dieser Samen keimen.
  - Aus Erfahrung ist bekannt,
    - dass die Zahl der keimenden Samen pro 1g Saatgut annähernd normalverteilt ist.
    - Bei frischem Saatgut (Population I) keimen im Durchschnitt 80 Samen, bei altem (Population II) sind es nur 40 Samen.
- Die Keimprobe hat nun x=70 ergeben
- Die Grafik zeigt, dass bei dieser Probe die „Likelihood“ der Population I am größten ist.
  - ► Man ordnet also diese Keimprobe als *frisch* ein.
- Antwortvariable ist binär „keimt“ / „keimt nicht“.
- Prädiktorenvariablen (z.B.): Menge der Wasserzufuhr, Anteil Sonnenlicht, Bodenbeschaffenheit, …
- Der Schnittpunkt der Verteilungsdichten (bei x=60) entspricht der Entscheidungsgrenze.



- Population I: Die Zahl der frischen Samen, die keimen, ist verteilt als $X_I \sim N(80; 10^2)$,
- Population II: Die Zahl der alten Samen, die keimen, ist verteilt als $X_{II} \sim N(40; 10^2)$,

Diskriminanzanalyse
Regeln

- Die Bildung der Diskriminanzfunktion erfolgt unter folgenden Bedingungen:
  - Die Varianz zwischen den Gruppenmittelwerten (Intergruppenvarianz) sollte möglichst groß sein.
  - Die Varianz innerhalb einer Gruppe (Intragruppenvarianz) sollte möglichst klein sein.
  - Die sich „überlappende“ Fläche (die eine Falschklassifikation bedeutet) sollte möglichst klein sein.

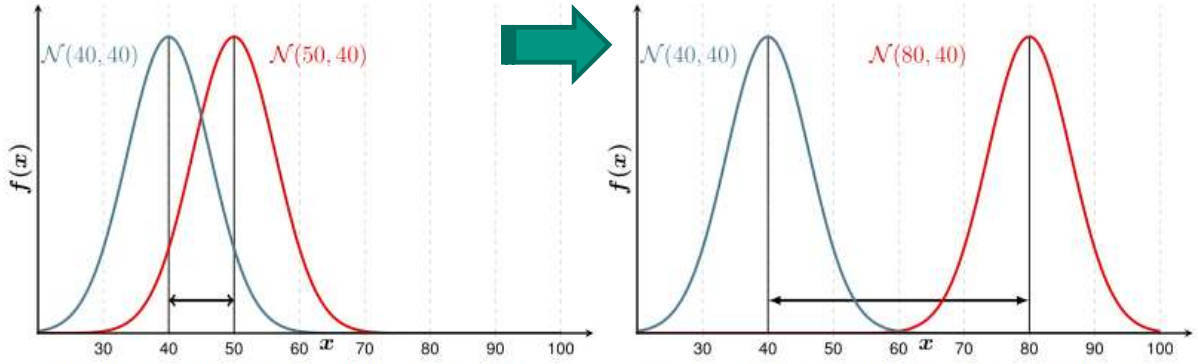


„Standardmodell“ der Diskriminanzanalyse

Es wird von gleichen Varianzen und Kovarianzen ausgegangen.

- Vergleich zwischen Gruppenmittelwerte

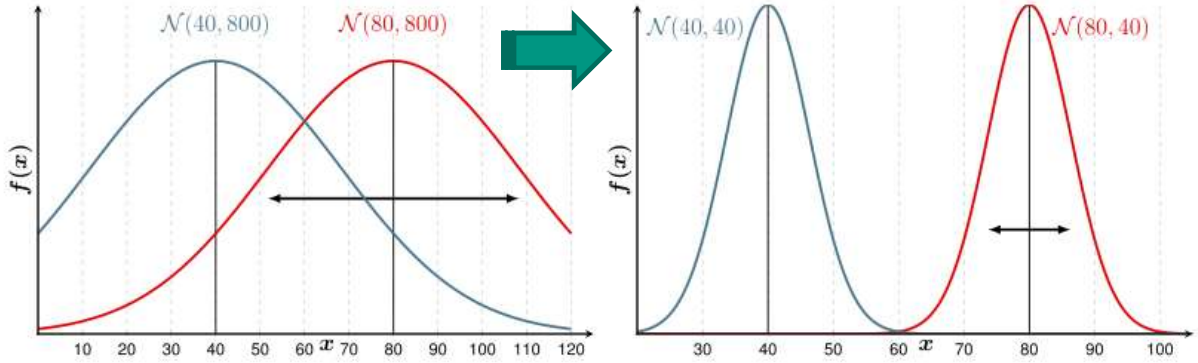
Die Varianz **zwischen** den Gruppenmittelwerten, die **Intergruppenvarianz**, sollte groß sein, weil sich dann die Verteilungen nicht durchmischen: Die Trennung der Gruppen ist schärfer.



Schlechter: Kleine Varianz zwischen den Gruppen | **Besser: Große Varianz zwischen den Gruppen**

- Varianz innerhalb einer Gruppe

Die Varianz **innerhalb** einer Gruppe, die **Intragruppenvarianz**, sollte möglichst klein sein, dann durchmischen sich die Verteilungen nicht, die Trennung ist besser.



Schlechter: Große Varianz in einer Gruppe | **Besser: Kleine Varianz in einer Gruppe**

03.07.20

Wissenschaftliche Hilfskraft

Mitarbeit beim Thema Augmentation mit maschinellem Lernen

Augmentieren von Bildern mit maschinellen Lernverfahren für Entwicklung & Test von automatisierten Fahrfunktionen

Beispiele

- Zusätzliche Fahrzeuge in Bilddaten synthetisch einfügen, vorhandene Fahrzeuge verändern
- Wetter einer Fahrszene anpassen
- Straßenschilder erkennen und verändern

Wir bieten

- Interdisziplinäres Arbeitsumfeld mit Partnern aus Wissenschaft, Wirtschaft und Anwendern
- Kreative und angenehme Arbeitsatmosphäre
- Flexible Arbeitszeiten

Mehr Informationen

Betreuer

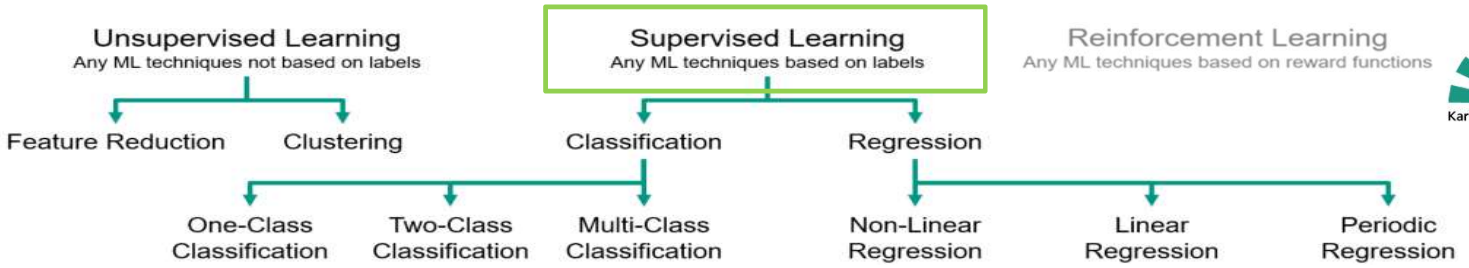
Philipp Rigoll

E-Mail: philipp.rigoll@fzi.de

Telefon: +49 721 9654-198

FZI Forschungszentrum Informatik

Machinelles Lernen

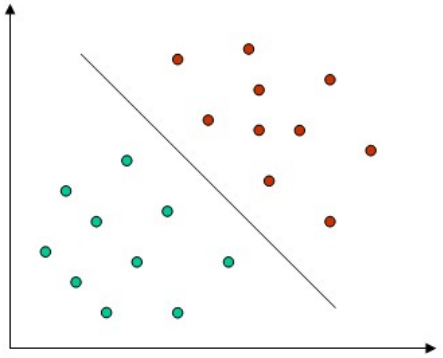


Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

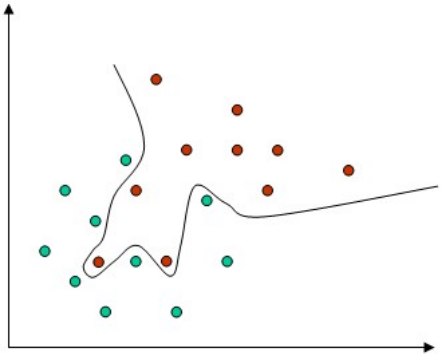
Support Vector Machine

Stützvektormethode: statistisches Verfahren, zur Klassifizierung von Datenobjekten

- Jedes Objekt wird durch einen Vektor in einem Vektorraum repräsentiert.
  - Aufgabe der Support Vector Machine ist es, in diesen Raum eine Hyperebene einzupassen, die als Trennfläche fungiert und die Trainingsobjekte in zwei Klassen teilt.
  - Der Abstand derjenigen Vektoren, die der Hyperebene am nächsten liegen, wird dabei maximiert.
  - Dieser breite, leere Streifen soll später dafür sorgen, dass auch Objekte, die nicht genau den Trainingsobjekten entsprechen, möglichst zuverlässig klassifiziert werden.
- Die Grenzfläche kann entweder einer linearen Funktion folgen oder auch von nicht linearem Charakter sein.
- Dazu wird eine Trainingsdatenmenge benötigt, bei der die Klassenzugehörigkeit bekannt ist.



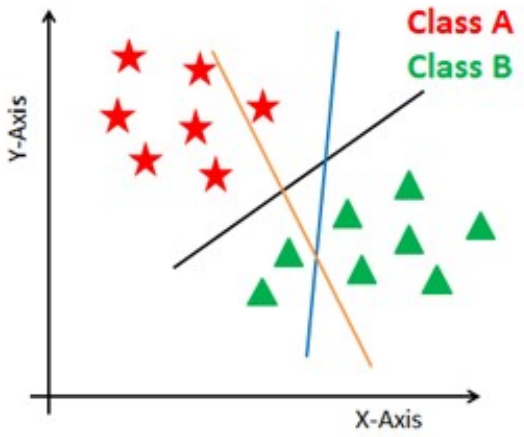
linear trennbar

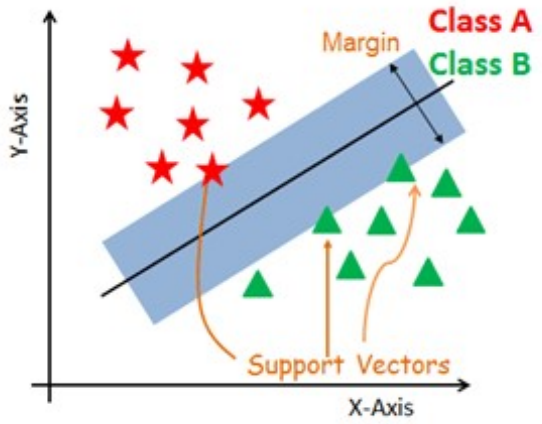


nicht linear trennbar

Support Vector Machine
Stützvektormethode

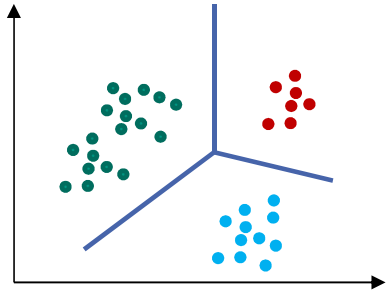
- Die Daten werden solange in einen höherdimensionalen Raum transformiert, bis eine lineare Trennung möglich ist.
- Diese lineare Trennung aus einem höherdimensionalen Raum wird dann wieder in die Ausgangsdimension zurücktransformiert und erscheint dann als „krumme Linie“, die die Gruppen trennt.
- Die Hyperebene ist nur von den ihr am nächsten liegenden Vektoren abhängig – und auch nur diese werden benötigt, um die Ebene mathematisch exakt zu beschreiben.
  - Diese nächstliegenden Vektoren werden nach ihrer Funktion Stützvektoren (engl. *support vectors*) genannt und verhalfen den Support Vector Machines zu ihrem Namen.





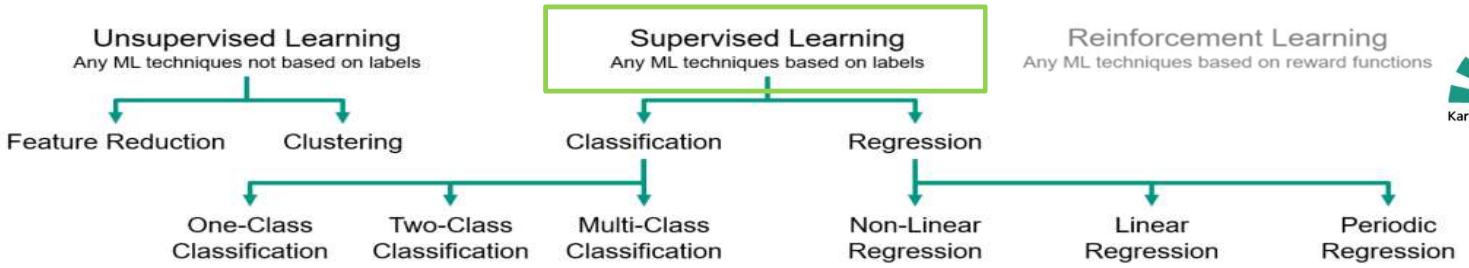
Support Vector Machine (überwacht)

- Gruppierung/ Separierung von vorsortierten Daten



- Vorteil:
  - Zielgerichtet durch Vorsortierung
  - Evaluation durch Testdaten möglich
    - Vergleich der entdeckten Muster (Optimierung)
    - Vergleich erleichtert Merkmalselektion
- Nachteil:
  - Nur anwendbar auf vorsortierte Daten
  - Neues Wissen beschränkt auf die vorgegeben Klassen

Machinelles Lernen



Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Bayes-Klassifikation (überwacht)

- Die Bayes-Klassifikation ist eine statistische Klassifikationsmethode, die die Wahrscheinlichkeit vorhersagt, mit der ein Objekt zu einer bestimmten Gruppe gehört.
  - Basiert auf der Formel von Bayes, mit der die bedingte Wahrscheinlichkeit eines Ereignisses berechnet werden kann.
- Bayes' Klassifikationsverfahren gehören zu den überwachten Klassifikatoren, da sie erst durch Trainingsdaten mit bekannter Klassifikation trainiert und dann auf neue Instanzen angewendet werden können.

Wahrscheinlichkeit für B unter der Bedingung, dass A_i eingetreten ist (Test erkennt das Merkmal korrekt)

Wahrscheinlichkeit für den Eintritt des Ereignisses A_i (Merkmal lag vor)

$$P(A_i \mid B) = \frac{P(B \mid A_i) * P(A_i)}{P(B)}$$

Wahrscheinlichkeit für A_i unter der Bedingung, dass B eingetreten ist

(Merkmal lag wirklich vor (A_i) wenn der Test es als vorliegend erkennt (B))

Wahrscheinlichkeit dafür, dass B eintritt (die Summe aller Pfade, an deren Ende eine Merkmalsidentifikation steht – egal ob zu Recht oder zu Unrecht)

Bayes-Klassifikation (überwacht)

- Die Entscheidungsregel funktioniert nach dem Prinzip, eine neue Instanz der Klasse zuzuordnen, bei der die berechnete Wahrscheinlichkeit für diese Klasse am größten ist.
  - Dabei wird eine A-priori-Wahrscheinlichkeit mit den gewichteten bedingten Wahrscheinlichkeiten berechnet.
- Bei vielen Variablen wird die Berechnung der (optimalen) Bayes-Klassifikation sehr aufwendig, sodass als Näherung die *Naive* Bayes-Klassifikation als Verfahren angewendet werden kann.
  - Alle Attribute werden dabei so behandelt als wären sie statistisch unabhängig.
  - Damit entfällt die Notwendigkeit der Berechnung der bedingten „Kreuzwahrscheinlichkeiten“ und der Berechnungsaufwand steigt nicht exponentiell mit der Anzahl der Variablen.
  - Anmerkung: *Obwohl die Annahme der Unabhängigkeit der Variablen in der Praxis häufig verletzt wird, liefert die Naive Bayes-Klassifikation trotzdem gute Ergebnisse (zumindest für den Fall, dass sich die Korrelationen in Grenzen halten).*

Naive Bayes-Klassifikation

Beispiel

- Wir haben Früchte, die mit drei Variablen beschrieben werden
  - Länge
  - Geschmack
  - Farbe
- In der Tabelle sind 1.000 Trainingsdaten dargestellt, mit denen das Klassifizierungsmodell erstellt werden soll.

	Länge		Geschmack		Farbe		Gesamt
	lang	kurz	süß	nicht süß	gelb	and. Farbe	
Banane	400	100	350	150	450	50	500
Orange	0	300	150	150	300	0	300
Sonst. Frucht	100	100	150	50	50	150	200
Gesamt	**500**	**500**	**650**	**350**	**800**	**200**	**1000**

Die A-priori-Wahrscheinlichkeiten sind:

$P_{(Banane)} = 0{,}5$ $P_{(Orange)} = 0{,}3$ $P_{(Sonst)} = 0{,}2$

$P_{(lang)} = 0{,}5$ $P_{(süß)} = 0{,}65$ $P_{(gelb)} = 0{,}8$

Likelihood – Ausprägungswahrscheinlichkeiten:

$P_{(lang \mid Banane)} = 0{,}8$ (400/500)

$P_{(lang \mid Orange)} = 0$ (es gibt keine langen Orangen)

.....

$P_{(and.\ Farbe \mid Orange)} = 0$

$P_{(and.\ Farbe \mid Sonst.)} = 0{,}75$ (150/200)

Beispiel: Lang, süß, gelb

Es soll nun eine neue Frucht klassifiziert werden, die lang, süß und gelb ist:

- Man berechnet zuerst die bedingte Wahrscheinlichkeit, dass die Frucht eine Banane, eine Orange oder eine andere Frucht ist.
- Die Formel für die Berechnung ist das Produkt aus den Ausprägungs-wahrscheinlichkeiten mit der A-priori-Wahrscheinlichkeit, geteilt durch die A-priori-Wahrscheinlichkeiten der Variablen.

- Sieht z. B. für die Banane so aus:

```
P(Banana|Long, Sweet and Yellow)
      P(Long|Banana) * P(Sweet|Banana) * P(Yellow|Banana) * P(banana)
    = ____________________________________________________________
                        P(Long) * P(Sweet) * P(Yellow)

    = 0.8 * 0.7 * 0.9 * 0.5 /P(evidence)

    = 0.252 /P(evidence)


P(Orange|Long, Sweet and Yellow) = 0


P(Other Fruit|Long, Sweet and Yellow)
      P(Long|Other fruit) * P(Sweet|Other fruit) * P(Yellow|Other fruit) * P(Other Fruit)
    = ____________________________________________________________
                                    P(evidence)

    = (100/200 * 150/200 * 50/200 * 200/1000) /P(evidence)

    = 0.01875 /P(evidence)
```

Wahrscheinlichkeit für B unter der Bedingung, dass A_i eingetreten ist (Test erkennt das Merkmal korrekt)

Wahrscheinlichkeit für den Eintritt des Ereignisses A_i (Merkmal lag vor)

$$P(A_i \mid B) = \frac{P(B \mid A_i) * P(A_i)}{P(B)}$$

Wahrscheinlichkeit für A_i unter der Bedingung, dass B eingetreten ist

(Merkmal lag wirklich vor (A_i) wenn der Test es als vorliegend erkennt (B))

Wahrscheinlichkeit dafür, dass B eintritt (die Summe aller Pfade, an deren Ende eine Merkmalsidentifikation steht – egal ob zu Recht oder zu Unrecht)

	Länge		Geschmack		Farbe		
	lang	kurz	süß	nicht süß	gelb	and. Farbe	
Banane	400	100	350	150	450	50	**500**
Orange	0	300	150	150	300	0	**300**
Sonst. Frucht	100	100	150	50	50	150	**200**
Gesamt	**500**	**500**	**650**	**350**	**800**	**200**	**1000**

Beispiel: Lang, süß, gelb

- Man berechnet zuerst die bedingte Wahrscheinlichkeit, dass die Frucht eine Banane, eine Orange oder eine andere Frucht ist.
- Die Formel für die Berechnung ist das Produkt aus den Ausprägungs-wahrscheinlichkeiten mit der A-priori-Wahrscheinlichkeit, geteilt durch die A-priori-Wahrscheinlichkeiten der Variablen.

$$P(\text{Banana}|\text{Long, Sweet and Yellow}) = \frac{P(\text{Long}|\text{Banana}) * P(\text{Sweet}|\text{Banana}) * P(\text{Yellow}|\text{Banana}) * P(\text{banana})}{P(\text{Long}) * P(\text{Sweet}) * P(\text{Yellow})}$$

$$= 0.8 * 0.7 * 0.9 * 0.5 / P(\text{evidence})$$

$$= 0.252 / P(\text{evidence})$$

$$P(\text{Orange}|\text{Long, Sweet and Yellow}) = 0$$

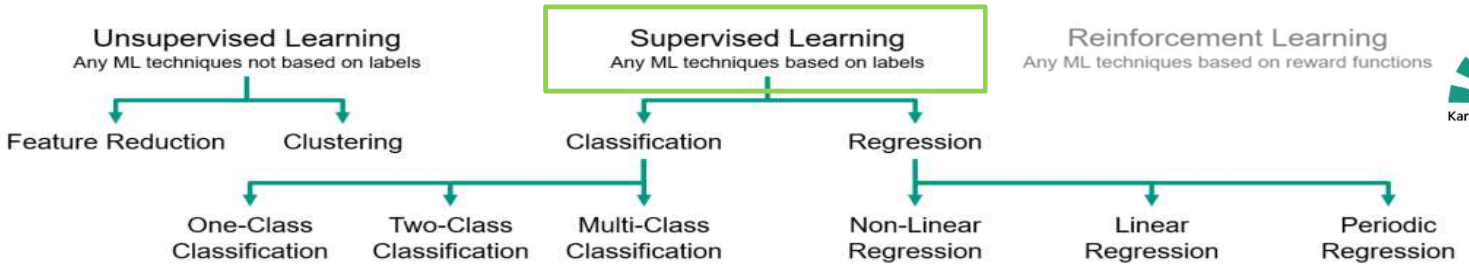
$$P(\text{Other Fruit}|\text{Long, Sweet and Yellow}) = \frac{P(\text{Long}|\text{Other fruit}) * P(\text{Sweet}|\text{Other fruit}) * P(\text{Yellow}|\text{Other fruit}) * P(\text{Other Fruit})}{P(\text{evidence})}$$

$$= (100/200 * 150/200 * 50/200 * 200/1000) / P(\text{evidence})$$

$$= 0.01875 / P(\text{evidence})$$

- Danach wählt man den wahrscheinlichsten Wert aus, sofern man mit dem Unterschied der Wahrscheinlichkeit zufrieden ist.
  - Da die lange, süße, gelbe Frucht mit dem Wert 0,252 mehr als 10-fach so wahrscheinlich eine Banane als eine sonstige Frucht ist (0,01875), kann man die unbekannte neue Frucht „guten Gewissens“ als Banane klassifizieren.
- Die naive Bayes-Klassifikation wird häufig für die Klassifikation von Texten verwendet, beispielsweise in Spam-Filter, die E-Mails als Spam bzw. kein Spam kategorisieren.

Machinelles Lernen

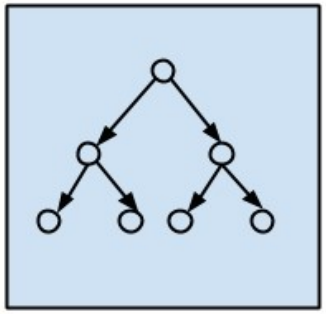


Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

Decision Tree Algorithms
Entscheidungsbäume

- Entscheidungsbäume sind Baumstrukturen, die der Darstellung von Entscheidungsregeln dienen.
  - Sie veranschaulichen hierarchisch aufeinanderfolgende Entscheidungen.
  - Ihre Anwendungsgebiete umfassen dabei die automatische Klassifizierung aus Erfahrungswissen.
  - Ein Entscheidungsbaum besteht immer aus einem Wurzelknoten und beliebig vielen inneren Knoten sowie mindestens zwei Blättern.
  - Dabei repräsentiert jeder Knoten eine Regel und jedes Blatt eine Antwort auf das Entscheidungsproblem.
- Grundsätzlich lassen sich Entscheidungsbäume in zwei Varianten unterteilen:
  - **Klassifikationsbäume** zeigen eine Auswahl von **diskreten** Klassen und deren Beziehungen untereinander.
  - **Regressionsbäume** dienen der Prognose eines **stetigen** Wertes der abhängigen Variable.

- Entscheidungsbäume können entweder von Experten manuell aufgestellt werden oder sie werden mit Techniken des maschinellen Lernens automatisch aus gesammelten Daten erstellt.
  - Bei letztgenanntem besteht die Möglichkeit, dem Algorithmus die Zahl der Ebenen vorzugeben.
  - Die Entscheidungsvariable wird automatisch über den relevantesten Informationsgehalt (Entropie) ausgesucht.

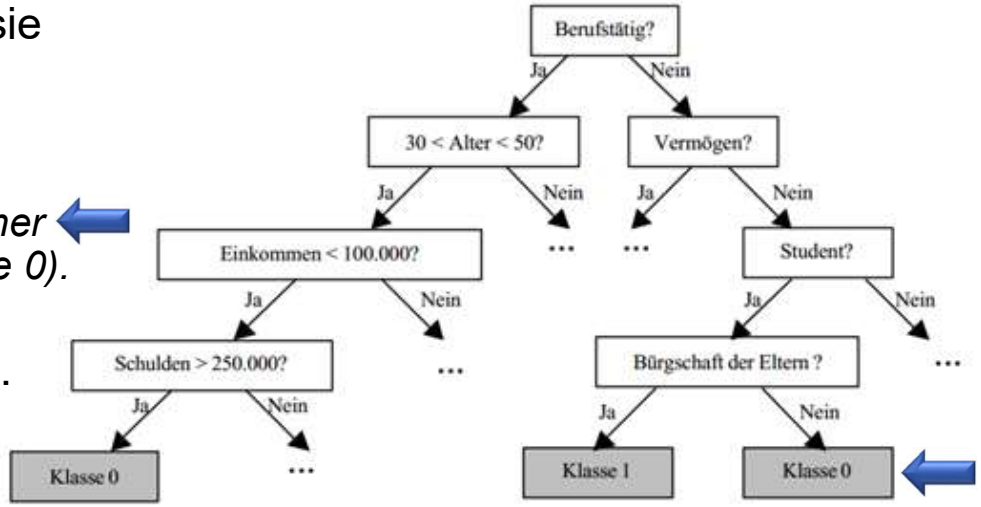


Decision Tree Algorithms

Entscheidungsbäume

Beispiel: Kreditvergabe

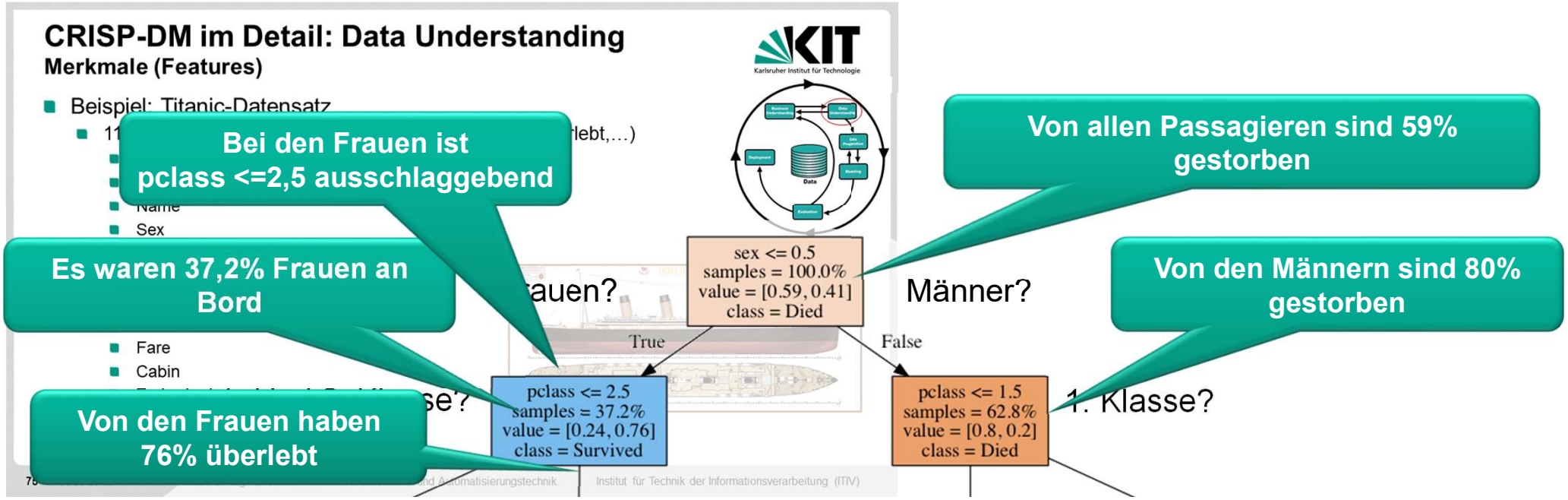
- Vorteile der Entscheidungsbäume liegen darin, dass sie i.d.R. gut zu verstehen und zu interpretieren sind.
  - Beispielsweise kann der Entscheidungsbaum zu einer Kreditentscheidung in Fließtext interpretiert werden:
  - *Einem mittellosen Studenten, der keine Bürgschaft seiner Eltern vorweisen kann, wird kein Kredit gewährt (Klasse 0).*
- Entscheidungsbäume führen häufig zu schlechteren Klassifikationsergebnissen als andere Verfahren (z. B. neuronale Netzwerke).
  - Die Verständlichkeit der Regeln geht auf Kosten der Klassifikationsgüte.
  - Außerdem besteht – je nach Verfahren und Art der Daten – die Gefahr, dass die Bäume zu groß und damit schwerer verständlich bzw. weniger aussagekräftig werden.



- Anmerkung:
  Eine Erweiterung der Entscheidungsbäume sind die Entscheidungswälder (Decision Forests).
  - *Dabei handelt es sich um den kombinierten Einsatz mehrerer Entscheidungsbäume.*
  - *Über eine Kombination der Entscheidungsbäume anhand von Mehrheitsentscheidungen soll die Klassifikationsgüte erhöht werden.*

Decision Tree

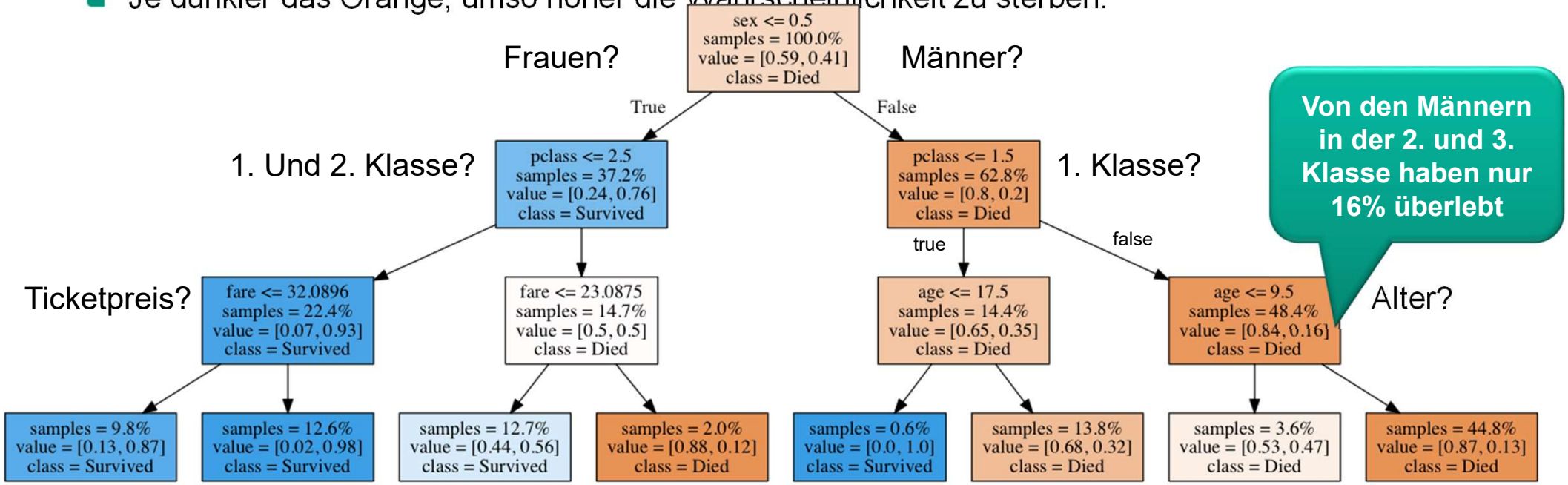
Beispiel: Titanic Dataset



Decision Tree

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

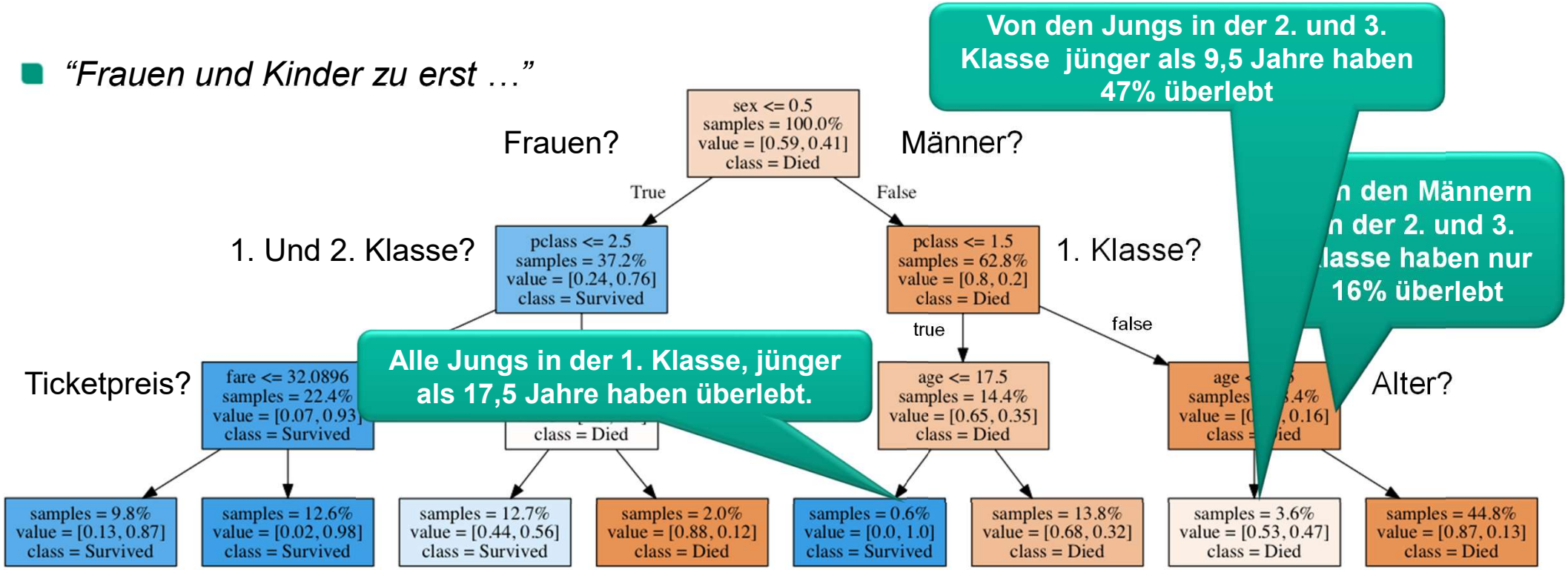
- Es wird dem Algorithmus eine max. Ebenenzahl von 3 (+ 1 Wurzelknoten) vorgegeben.
- Der Algorithmus teilt zunächst nach “sex” und dann “class”, da in der Traingingsphase die beiden Parameter als am einflussreichsten identifiziert wurden (Entropie), um über Leben und Tod zu entscheiden.
  - Je dunkler das Blau, umso höher die Wahrscheinlichkeit zu überleben
  - Je dunkler das Orange, umso höher die Wahrscheinlichkeit zu sterben.



Decision Tree

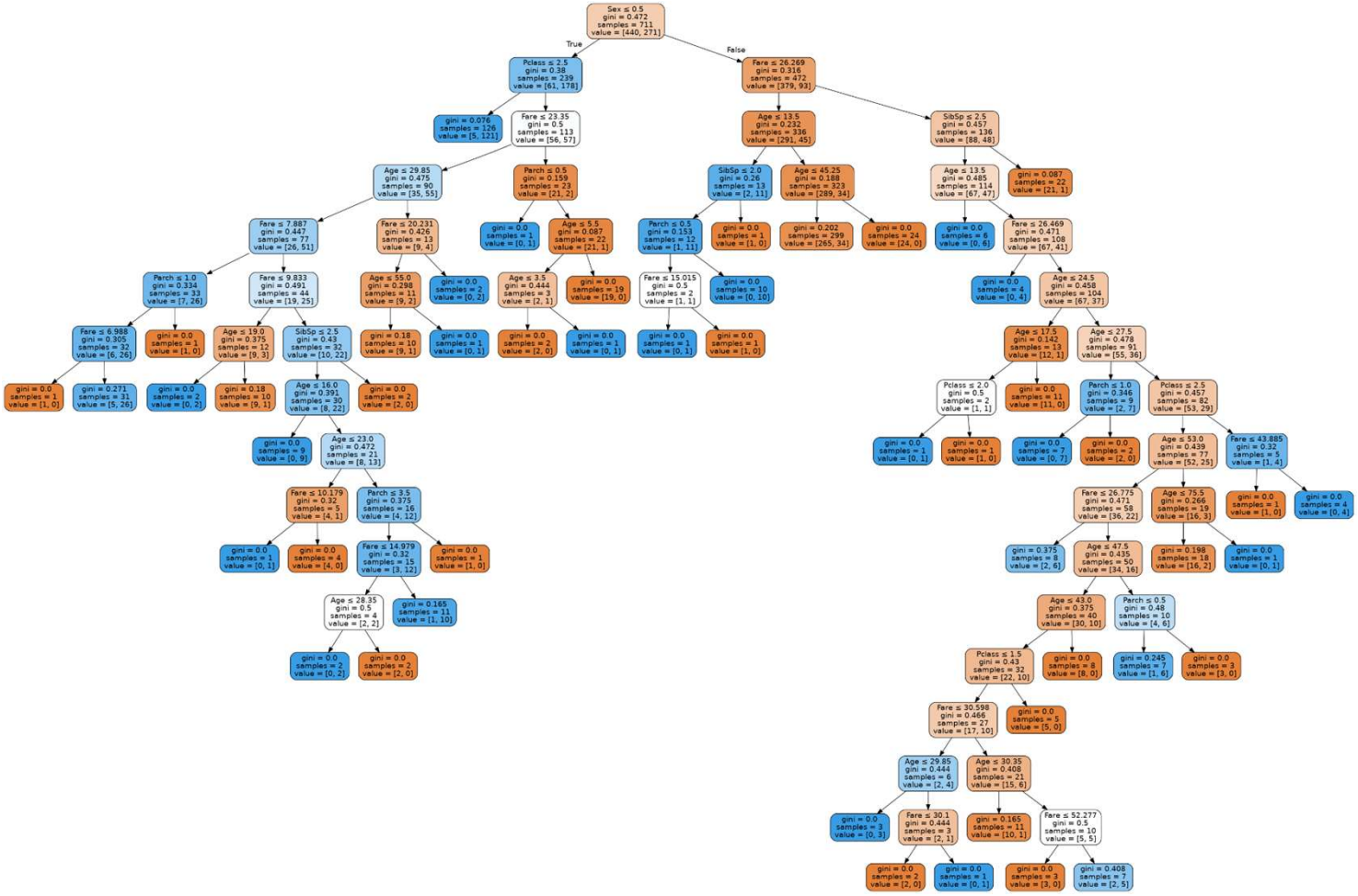
http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

- Interessanterweise ist nach dem Split nach “class” bei Frauen (1. und 2. Klasse) der Ticketpreis und bei Männern (1. Klasse) das Alter der nächst entscheidende Faktor
- `value=[died, survived ]` → der Wert darunter gibt die größere Zahl von beiden an.
- *“Frauen und Kinder zu erst …”*



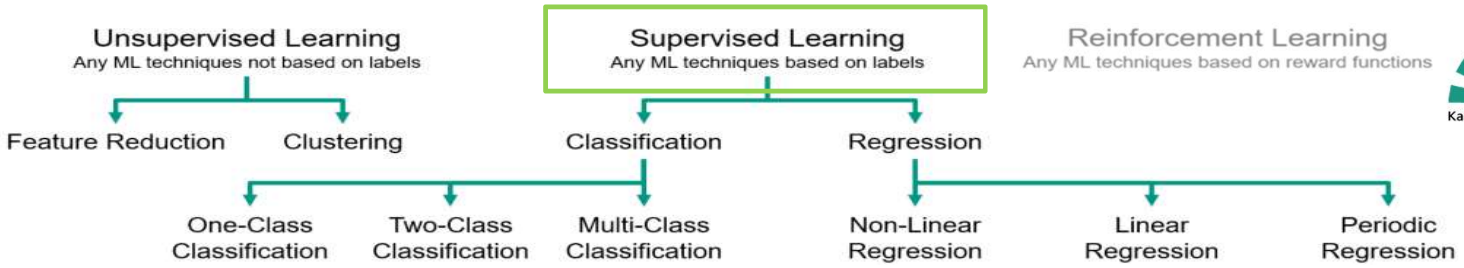
… und falls man das Verfahren nicht auf 3 Ebenen beschränkt …

Die „Blätter“ sind nun eindeutig „survived“ und „died“




5 59

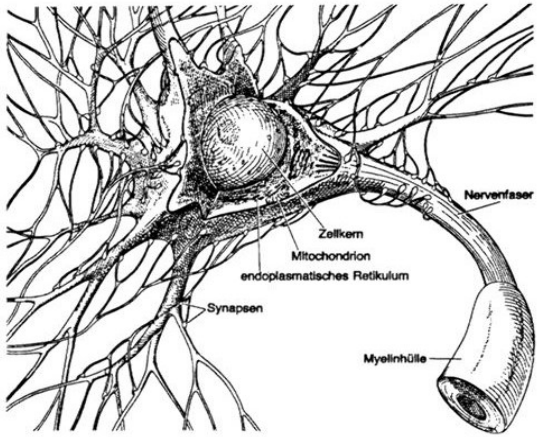
Machinelles Lernen



Algorithm ↓	Abk.	Feature Reduction	Clustering	One-Class Classification	Two-Class Classification	Multi-Class Classification	Non-Linear Regression	Linear Regression	Periodic Regression
Faktorenanalyse		X							
Principal Component Analysis	PCA	X		X					
	K-means		X						
hierarchical cluster analysis	HCA		X						
DBSCAN			X						
One Class Support Vector Machine	OCSVM			X					
Isolation Forest				X					
	LODA			X					
(künstliche) Neuronale Netze	NN			X (Autoencoder)	X	X	X	X	X
Support Vector Machine	SVM				X				
Decision Tree					X	X			
Bayes-Klassifikation					X	X			
Random Forest						X			
Diskriminanzanalyse				x	x	x			
Logistic Regression							X	X	
Linear Regression								X	
Harmonic Regression									X
Nächste-Nachbar-Klassifikation	K-NN	x	x						

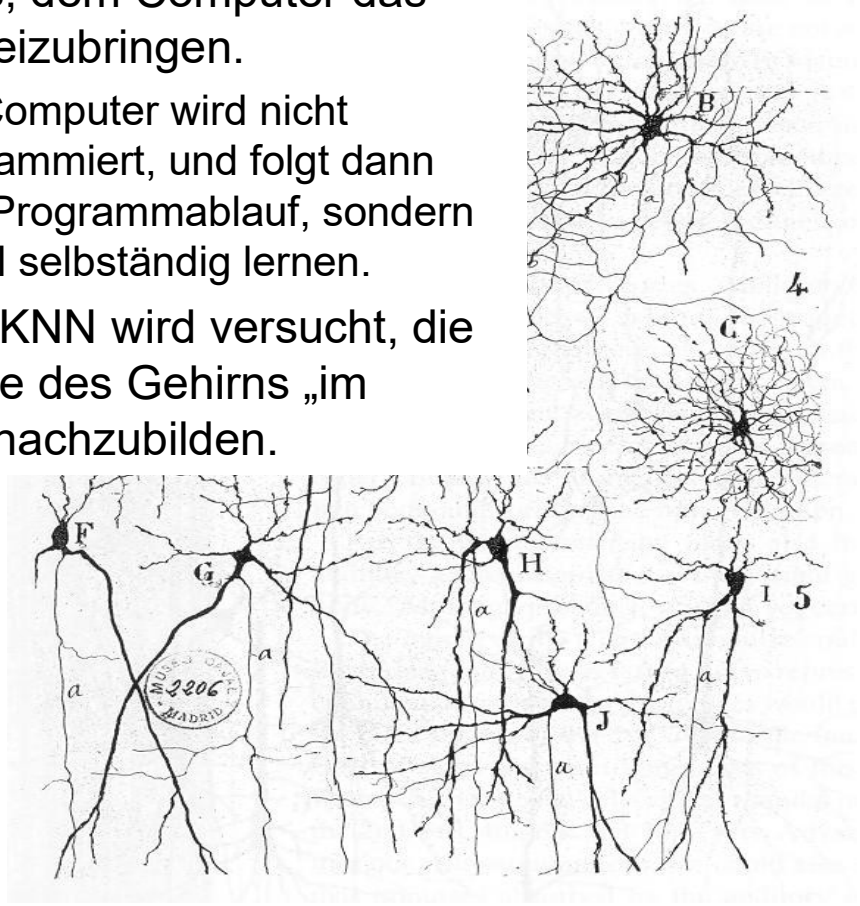
Neuron, (künstliches) Neuronales Netz / KNN

- Kombination aus mathematischer Matrizenrechnung und einem Ablaufschema (Lernalgorithmus), mit dem iterativ Verbesserungen ermöglicht werden „sollen“.
- Keine „Rocket Science“ und keine „Frankenstein’sche Alchemie“
- Aufbau von Neuronen
  - Gehirn besteht aus 100 Milliarden Neuronen



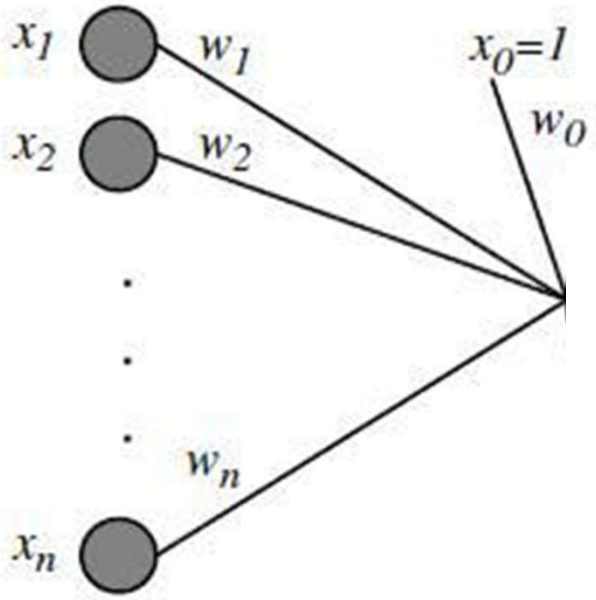
- Dendriten nehmen elektrische Impulse auf
- Zellkörper verarbeitet ankommende Impulse
- Nervenfaser (Axon) leitet gefeuerten Impuls weiter
- Synapsen sind die Endknöpfchen des Axons, übertragen Impulse an nächste Zelle

- Ziel ist es, dem Computer das Lernen beizubringen.
  - Der Computer wird nicht programmiert, und folgt dann dem Programmablauf, sondern er soll selbständig lernen.
- In einem KNN wird versucht, die Grundidee des Gehirns „im Kleinen“ nachzubilden.



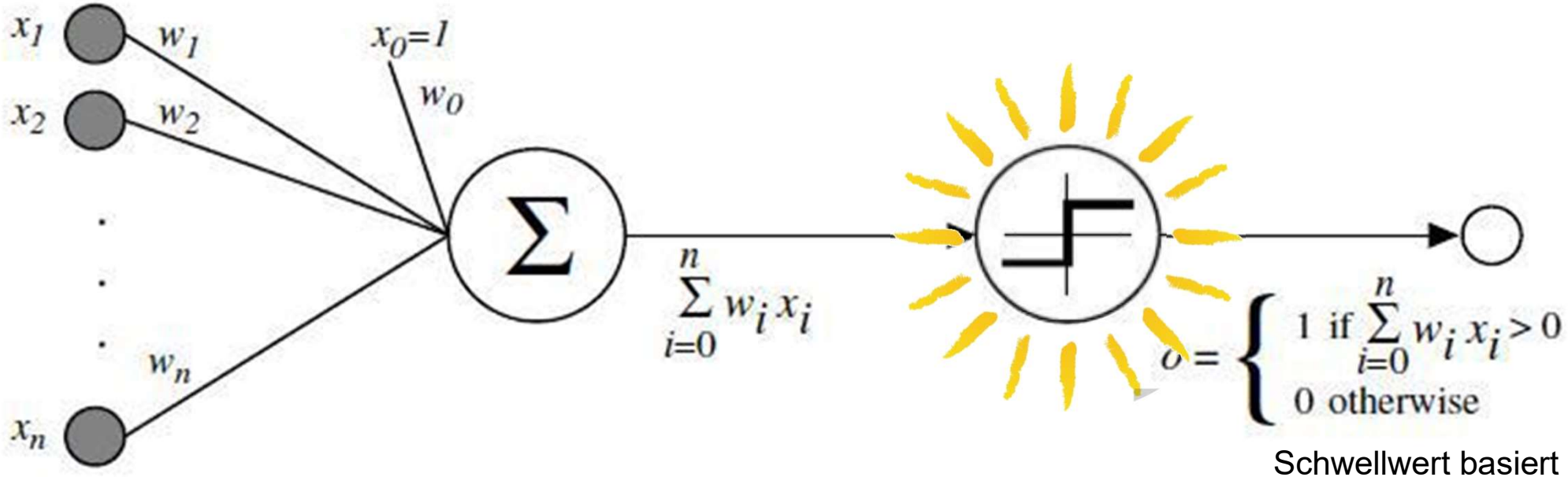
Perceptron

- Das Perzeptron ist ein vereinfachtes künstliches neuronales Netz, das zuerst von Frank Rosenblatt 1958 vorgestellt wurde.
  - Es besteht in der Grundversion aus 1 einzelnen künstlichen Neuron mit anpassbaren Gewichtungen.



Perceptron

- Die Aktivierungsfunktion beschreibt den funktionalen Zusammenhang zwischen Input und Output eines Knotens.
- Der gesamte Input wird über die sog. Propagierungsfunktion bzw. Übertragungsfunktion bestimmt.

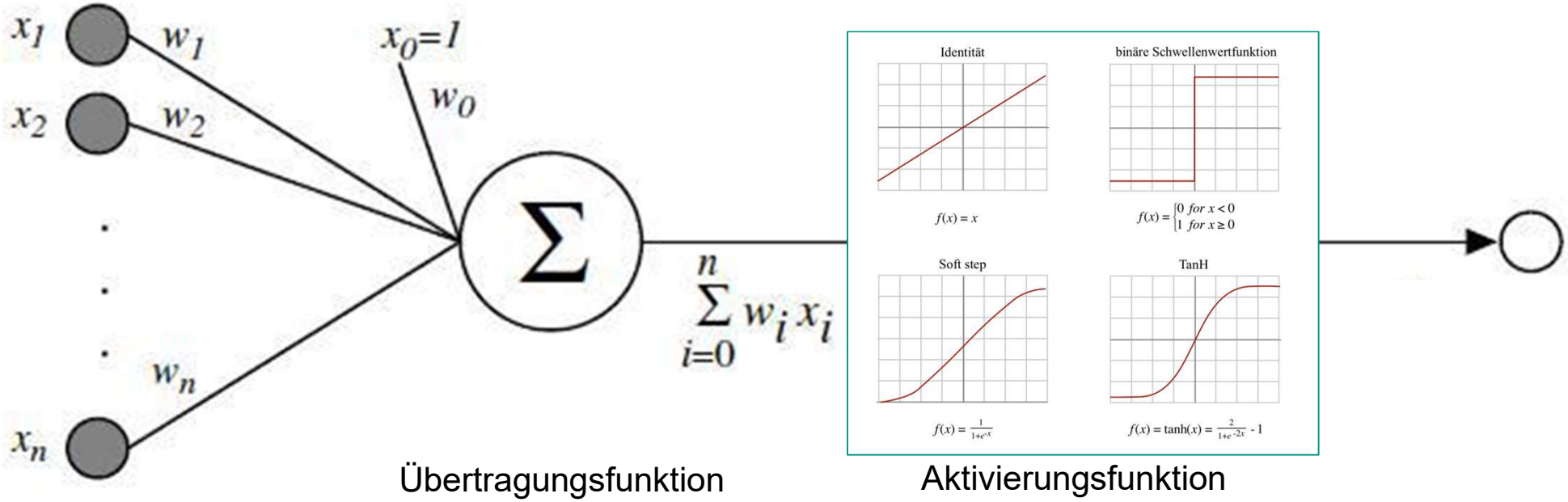


Schwellwert basiert

Quelle: http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/

Perceptron

- Die verbreitetste Übertragungsfunktion ist eine Linearkombination, bei der sich der Netzinput additiv aus sämtlichen gewichteten Inputs zusammensetzt, die das Neuron von anderen Neuronen erhält.



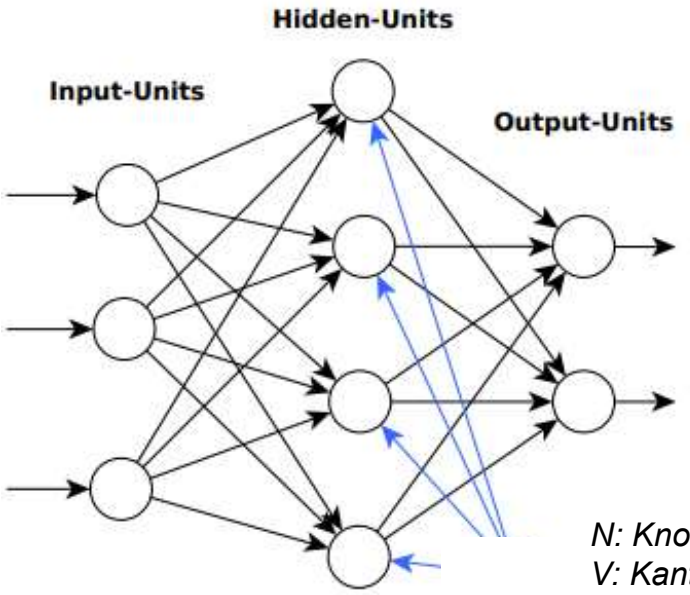
Übertragungsfunktion Aktivierungsfunktion

Quelle: http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/

Definition des künstlichen neuronalen Netzes (KNN)

- Ein KNN besteht aus Knoten (Einheiten oder Neuronen N) und Kanten (Verbindungen V).
  - Die Stärke der Verbindung wird durch ein Gewicht w ausgedrückt.
- Die Knoten verfügen über einen Wert der Aktivierung, der beschreibt ob und wie aktiv er ist.
  - Das ist sozusagen der Schaltzustand.
- Die Struktur des Netzwerkes wird als „Topologie“ bezeichnet.
- Das Wissen des Netzwerks besteht aus der Gesamtheit der Gewichte der einzelnen Kanten.
  - Als Ausgangswerte werden i.d.R. Zufallswerte genommen.
  - Lernen erfolgt durch iteratives Anpassen der Gewichtungen nach unterschiedlichen Lernalgorithmen.

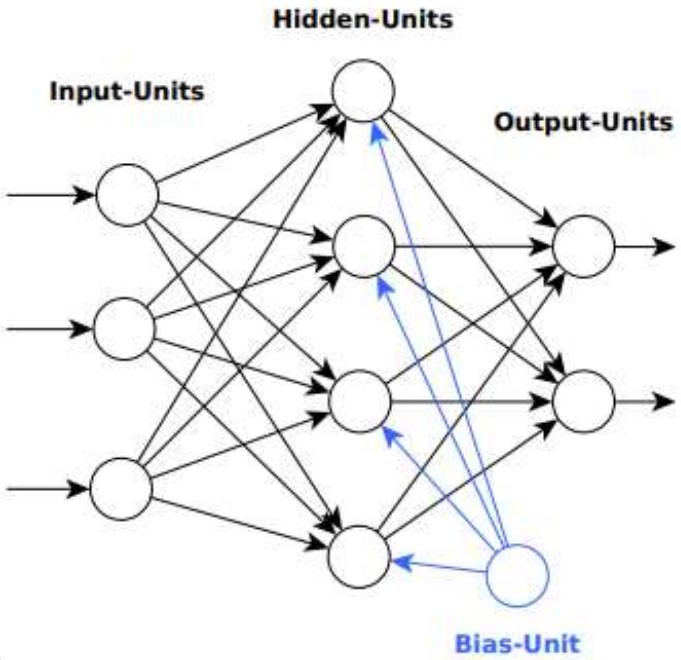
„Ein Neuronales Netz ist ein sortiertes Tripel (N, V, w) mit zwei Mengen N, V sowie einer Funktion w, wobei N die Menge der Neurone bezeichnet und V eine Menge $\{(i, j)|i, j \in N\}$ ist, deren Elemente Verbindungen von Neuron i zu Neuron j heißen. Die Funktion $w : V \rightarrow R$ definiert die Gewichte, wobei w((i, j)), das Gewicht der Verbindung von Neuron i zu Neuron j, kurz mit wi,j bezeichnet wird.“



N: Knoten
V: Kanten
w: Gewichte

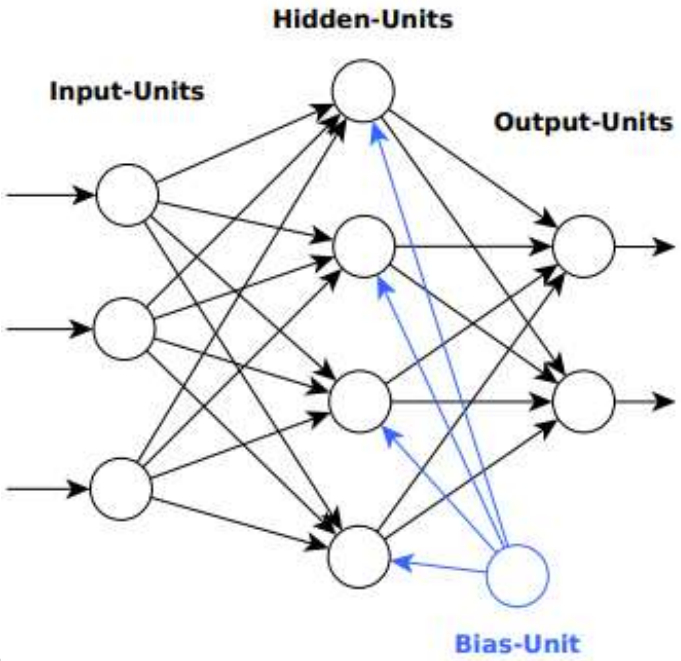
Arten von Neuronen

- Input-Units:
  - nur ein Eingang
  - Aktivierungsfunktion entfällt
- Hidden-Units:
  - Ein- und Ausgabe unbekannt
  - Deshalb „versteckte“ Knoten
- Output-Units:
  - Ausgabe des Systems



<zur weiteren Vertiefung>

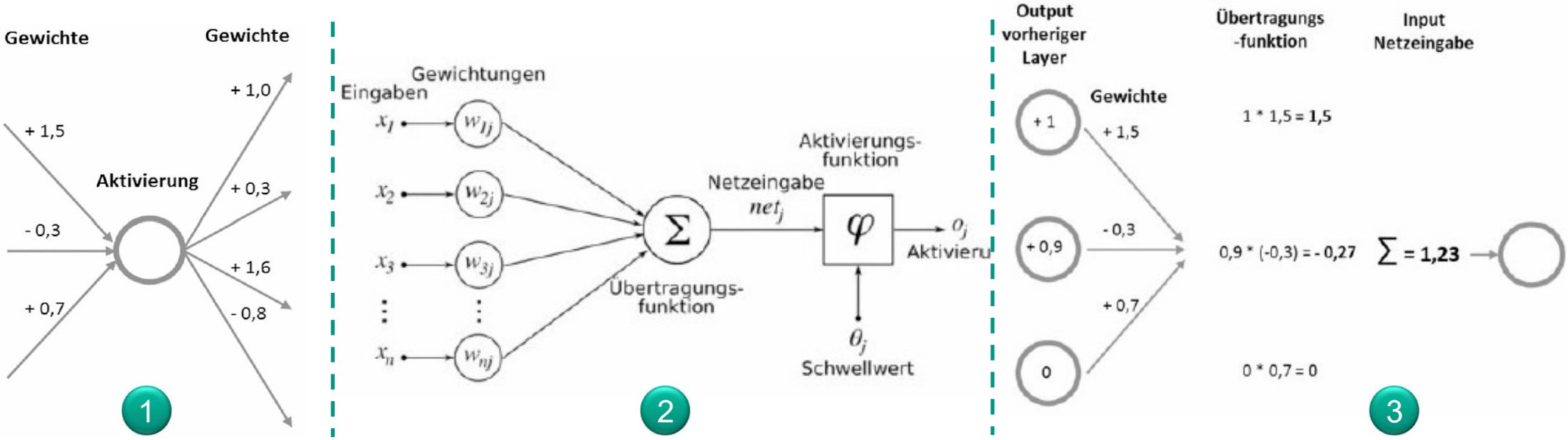
- Bias-Units:
  - Die Bias-Unit erhält selbst keinen Input,
  - ihr Aktivitätslevel beträgt immer +1.
  - Das Gewicht von der Bias-Unit zu einer anderen Unit kann positiv oder negativ sein.
  - Wenn kein starker Input von anderen Einheiten erfolgt, dann stellt die Bias-Unit sicher, dass die Einheit bei positivem Gewicht aktiv bleibt.
  - Bei negativem Gewicht sorgt die Bias-Einheit hingegen dafür, dass die Unit in ihrem inaktiven Zustand verharrt.
  - Funktion der Bias-Unit
    - kann nützlich sein, wenn man eine Schwelle benötigt (beim negativen Bias), die andere Input-Units erst überschreiten müssen.
    - Umgekehrt kann das Ziel auch sein, dass die Einheit sehr häufig feuern, also gewöhnlich aktiv sein soll. Dazu verwendet man einen positiven Bias.



Beispiel

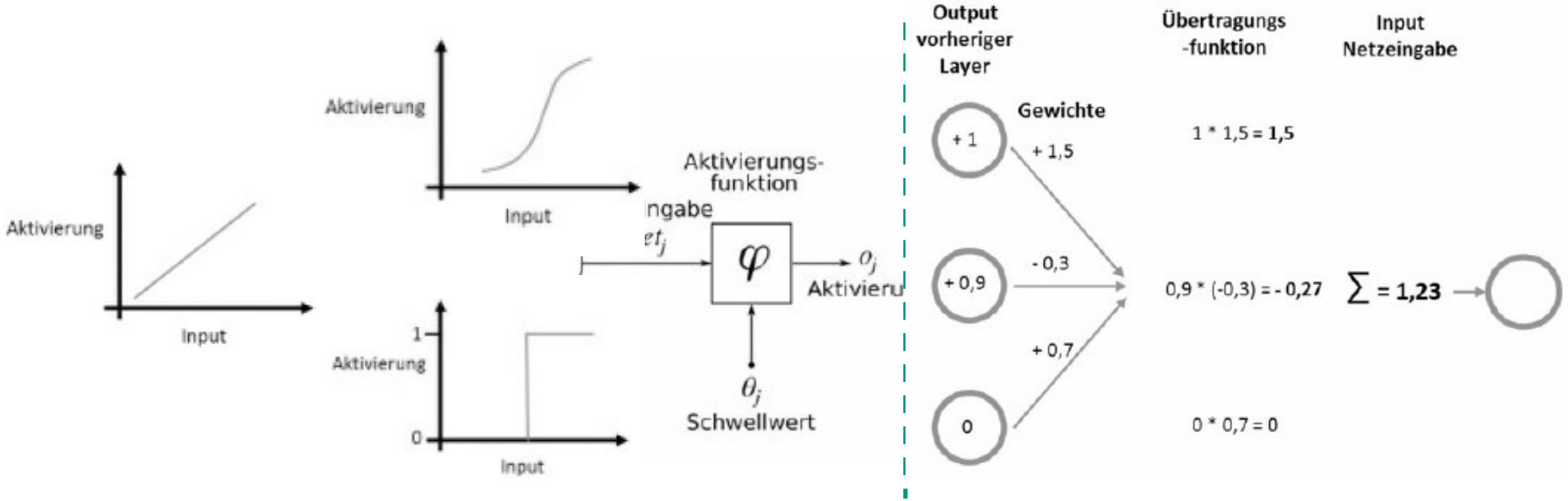
Ausgangspunkt sind gegebene, willkürliche Werte für die Gewichte der Kanten (1).

- Der Wert für die Aktivierung und den Output der Einheit in Abhängigkeit vom Input in das Netzwerk ergibt sich über Zusammenhang (2)
- Der Input x_1 bis x_n entspricht dem Output der entsprechenden vorangegangen Knoten.
- Dieser wird mit den Kantengewichten multipliziert und summiert (3)
  - Die Übertragungsfunktion ist also annahmegemäß die Summe der gewichteten Eingaben x_1 bis x_n.



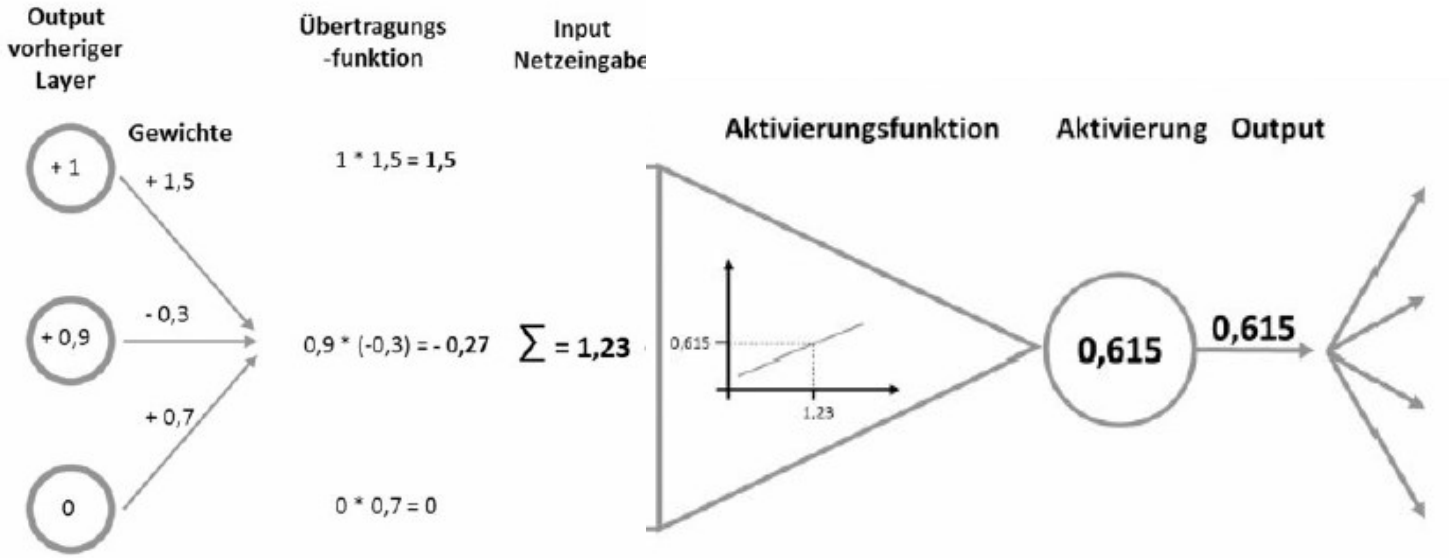
Aktivierungsfunktion

- Aus dem Input wird über die Aktivierungsfunktion der Aktivierungswert berechnet.
- Es handelt sich um eine funktionale Zuordnung des Inputs mit der Aktivierung der Zelle.
- Die Aktivierungsfunktion kann unterschiedliche Ausprägungen einnehmen:
  - Lineare Funktion ohne/mit Schwellenwert
  - Binäre Funktion
  - Sigmoide (z. B. logistische) Funktion



Aktivierungsfunktion

- Aus dem Input wird über die Aktivierungsfunktion der Aktivierungswert berechnet.
- Es handelt sich um eine funktionale Zuordnung des Inputs mit der Aktivierung der Zelle.
- Annahme: die Aktivierungsfunktion ist eine lineare Funktion ohne Schwellenwert mit der „Steigung“ 0,5 und die Aktivierung entspricht dem Output:
  - Aktivierung = 0,5 *
  - Input = Output = 0,615



- Berechnung einer Einheit bzw. eines Neurons!

Fragen bzw. Handlungsoptionen

Neuronale Netze, Einstellgrößen (*Hyperparameter*)

- Soll das KNN für ein überwachtes oder unüberwachtes Lernen eingesetzt werden, oder anders ausgedrückt:
  Liegen ge-labelte Lerndaten vor (also Trainingsdaten, für die Input- und Output-Werte) oder soll mit dem KNN eine Datenmenge mit nicht vorhandenen Output-Werten zur Erkennung von Mustern verwendet werden?
  - Im ersten Fall müssen über einen Lernalgorithmus die Gewichtungen der Kanten solange angepasst werden, bis das Netz die Trainingsdaten hinreichend gut wiedergibt.
  - Im zweiten Fall muss der Lernalgorithmus die Gewichtungen im Netz solange anpassen, bis das Lernziel (z. B. die Mustererkennung oder die Clusterbildung) hinreichend gut erfüllt ist.

    ► Es müssen also die für das Ziel des Lernens geeigneten Lernregeln ausgewählt werden.

Fragen bzw. Handlungsoptionen

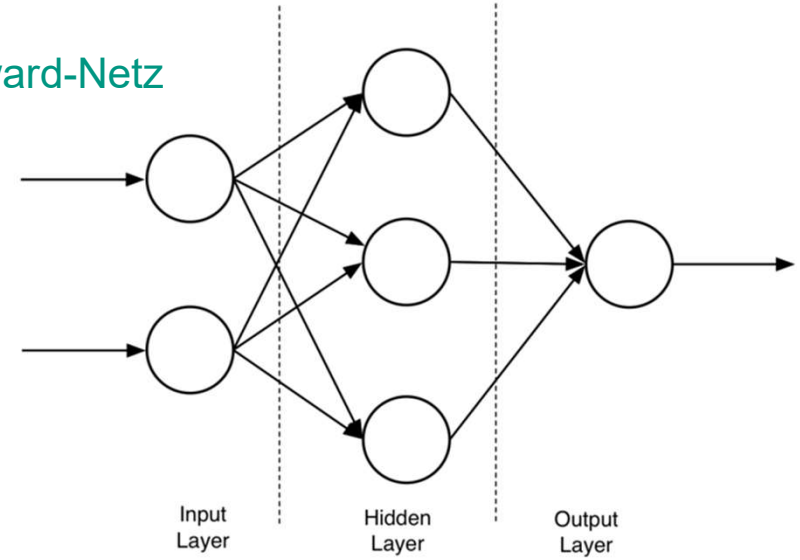
Neuronale Netze, Einstellgrößen (*Hyperparameter*)

- Soll das KNN für ein überwachtes oder unüberwachtes Lernen eingesetzt werden, oder anders ausgedrückt:
  Liegen ge-labelte Lerndaten vor (also Trainingsdaten, für die Input- und Output-Werte) oder soll mit dem KNN eine Datenmenge mit nicht vorhandenen Output-Werten zur Erkennung von Mustern verwendet werden?
  - Im ersten Fall müssen über einen Lernalgorithmus die Gewichtungen der Kanten solange angepasst werden, bis das Netz die Trainingsdaten hinreichend gut wiedergibt.
  - Im zweiten Fall muss der Lernalgorithmus die Gewichtungen im Netz solange anpassen, bis das Lernziel (z. B. die Mustererkennung oder die Clusterbildung) hinreichend gut erfüllt ist.
    ► Es müssen also die für das Ziel des Lernens geeigneten Lernregeln ausgewählt werden.
- Welche Topologie soll mein KNN haben?
  - Wie viele Knoten und wie viele Hidden Layers soll das Netz haben?
- Wie ist die Richtung der Verbindungen?
  - Die Verbindungen können ausschließlich in eine Richtung (von „links nach rechts“) und immer nur zwischen zwei benachbarten Layers vorkommen (*Feedforward Netze*), oder aber es sind Rückkopplungen in die andere Richtung (indirekte Rückkopplung), im selben Layer (seitliche Rückkopplung), oder sogar mit derselben Unit (*direkte Rückkopplung*) möglich.
- Welche Aktivierungsfunktion soll ausgewählt werden?

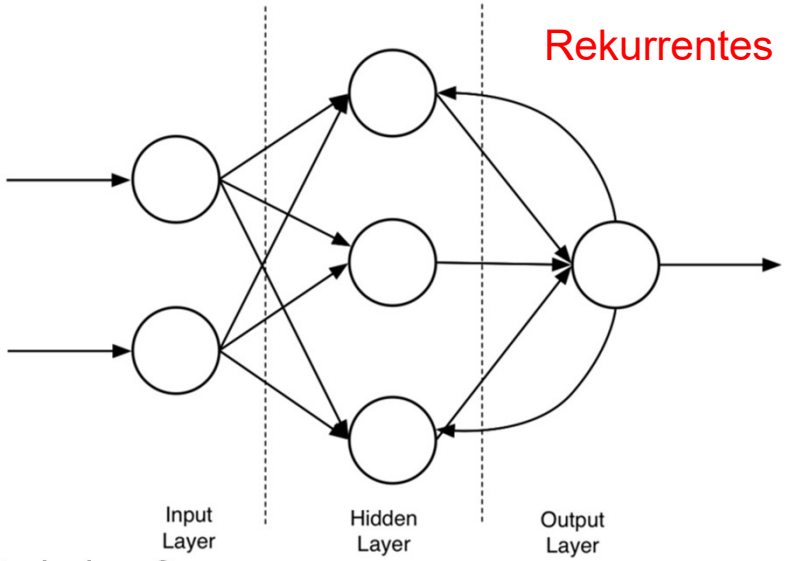
Fragen bzw. Handlungsoptionen

Neuronale Netze, Einstellgrößen (*Hyperparameter*)

Feedforward-Netz



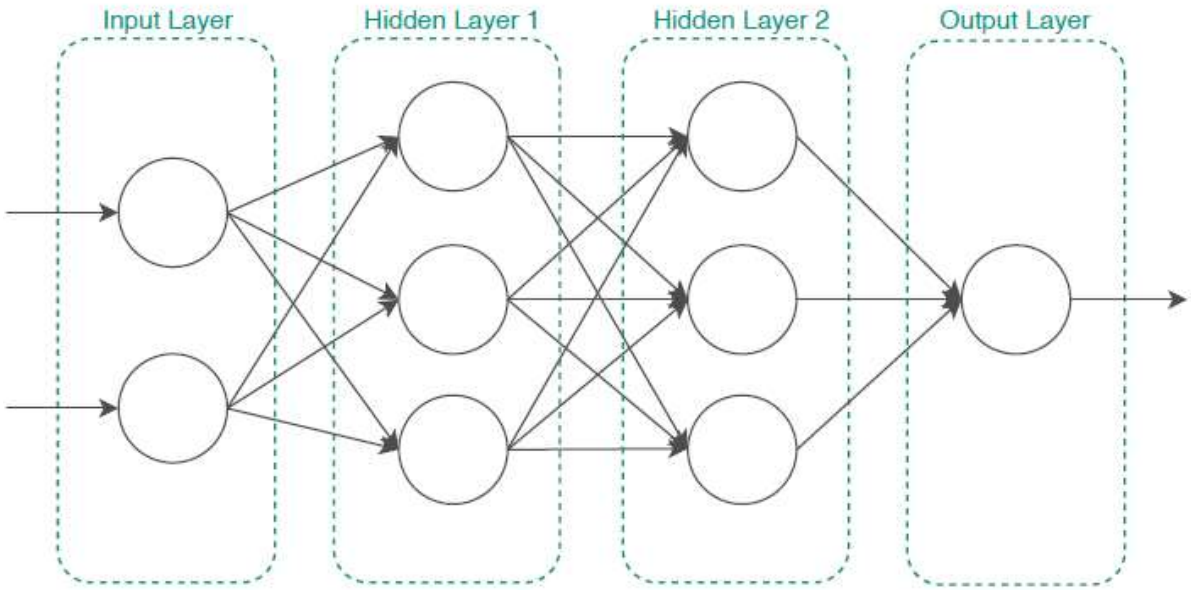
Rekurrentes Netze



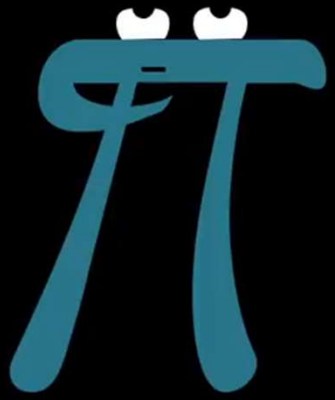
- Wie viele Knoten und wie viele Hidden Layers soll das Netz haben?
- Wie ist die Richtung der Verbindungen?
  - Die Verbindungen können ausschließlich in eine Richtung (von „links nach rechts“) und immer nur zwischen zwei benachbarten Layers vorkommen (*Feedforward Netze*), oder aber es sind Rückkopplungen in die andere Richtung (*indirekte Rückkopplung*), im selben Layer (seitliche Rückkopplung), oder sogar mit derselben Unit (*direkte Rückkopplung*) möglich.
- Welche Aktivierungsfunktion soll ausgewählt werden?

Deep Learning

- Es gibt unzählige Ausprägungen an Netzwerken.
- Ein wichtiges Merkmal ist die Frage, ob und wieviele „Hidden Layers“ vorhanden sind.
- Bei mehr als einem Hidden Layer spricht man von **Deep Learning**
- Die am weitesten verbreiteten sind:
  - Convolutional Neural Network (CNN)
  - Stacked Auto-Encoders

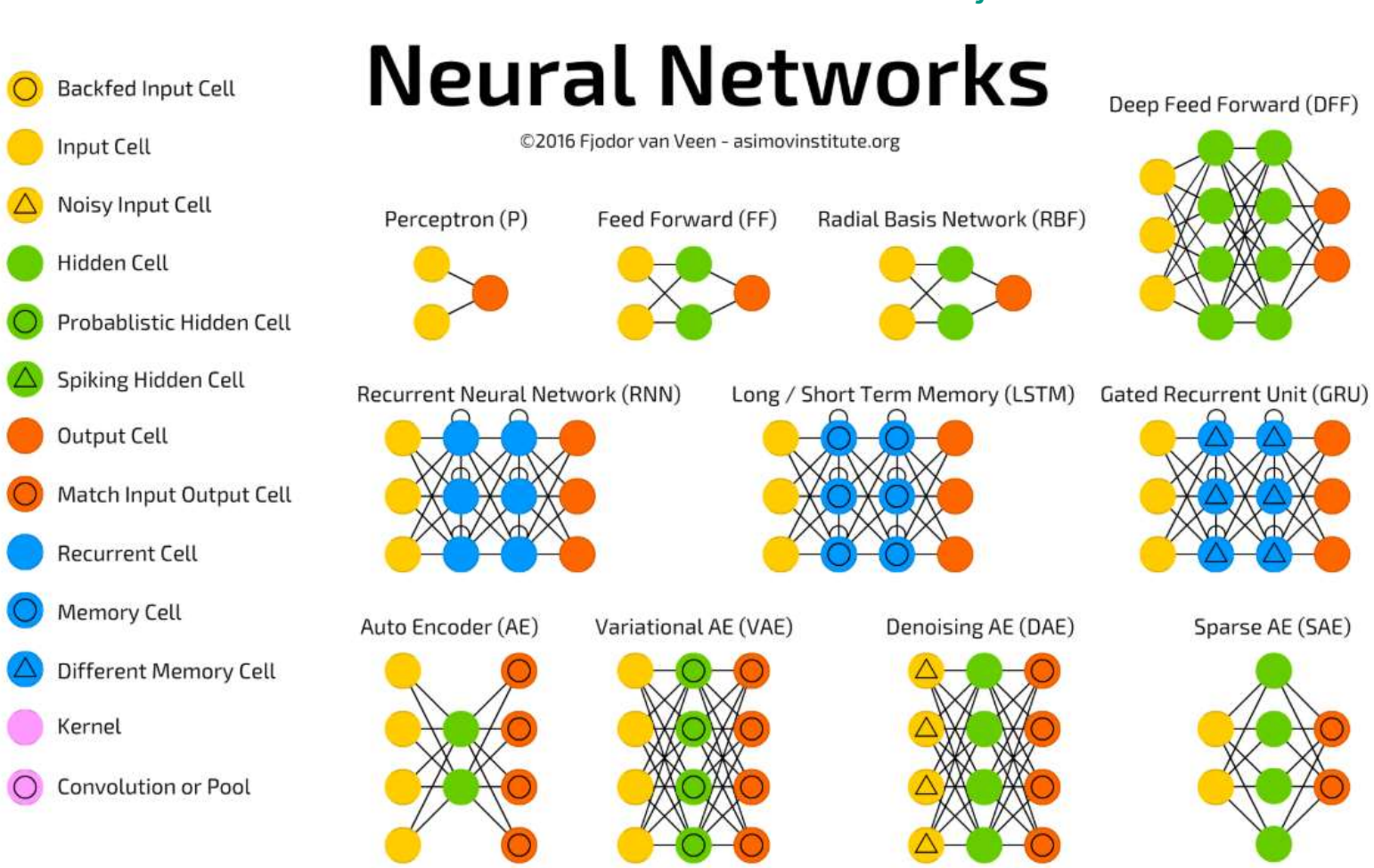


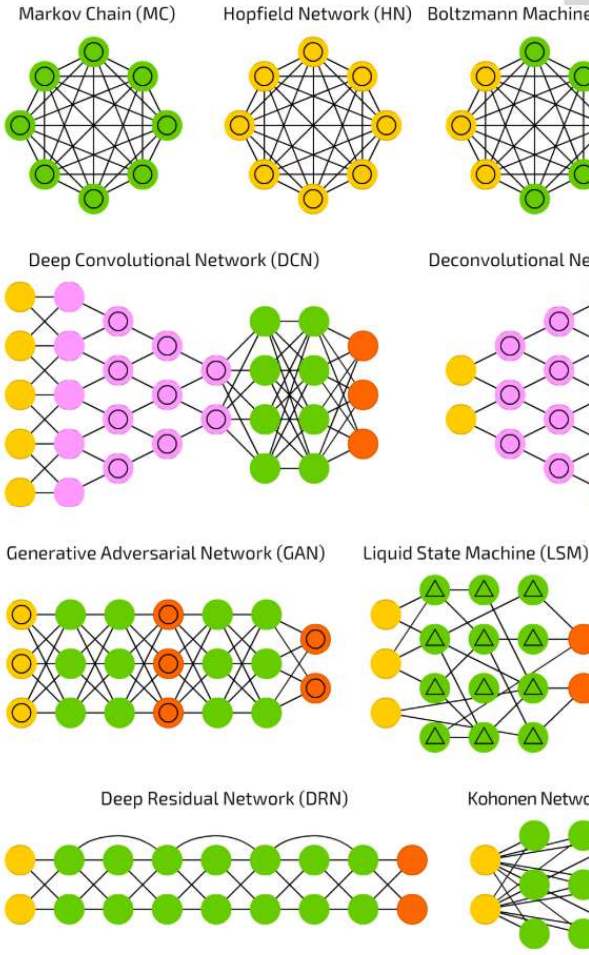
Neural network



Übersicht Architekturen

Verschiedene Arten von neuronalen Netzen nach Fjodor van Veen





Convolutional Neural Networks (CNN)

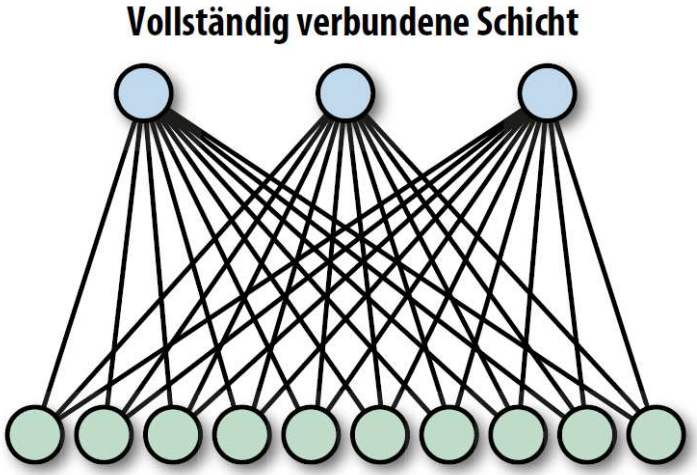
„faltendes Neuronales Netz“ → Stand der Technik für Bilderkennung

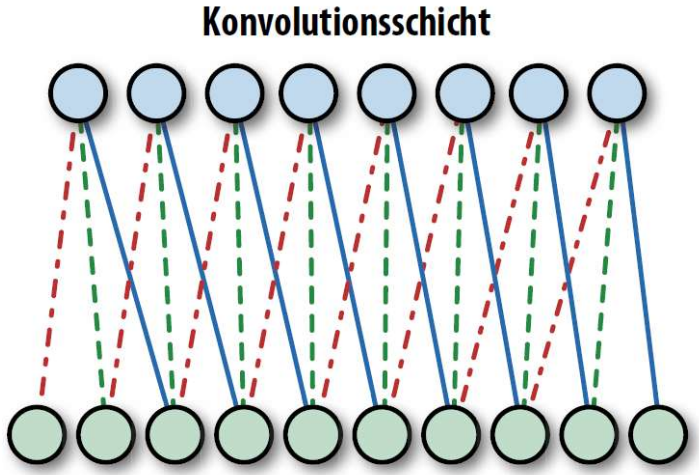
- Der grundlegende Unterschied zwischen *vollständig verbundenen Netzen* und *Konvolutionsnetzen* ist das Verbindungsmuster aufeinanderfolgender Schichten.
  - In einer vollständig verbundenen Schicht ist jede Einheit mit allen Einheiten der vorigen Schicht verbunden
- In der Konvolutionsschicht eines CNN dagegen ist jede Einheit mit wenigen räumlich nahen Einheiten der vorigen Schicht verknüpft (passend z.B. bei Bildern, wo Zusammenhänge zwischen weit entfernten Bildpunkten nicht in Betracht gezogen werden)
  - Außerdem sind sämtliche Einheiten mit der vorigen Schicht auf dieselbe Weise, mit genau denselben Gewichten und derselben Struktur verbunden.

Convolutional Neural Networks (CNN)

„faltendes Neuronales Netz“

- Der grundlegende Unterschied zwischen *vollständig verbundenen Netzen* und *Konvolutionsnetzen* ist das Verbindungsmuster aufeinanderfolgender Schichten.
  - In einer vollständig verbundenen Schicht ist jede Einheit mit allen Einheiten der vorigen Schicht verbunden.
- In der Konvolutionsschicht eines CNN dagegen ist jede Einheit mit wenigen räumlich nahen Einheiten der vorigen Schicht verknüpft (passend z.B. bei Bildern, wo Zusammenhänge zwischen weit entfernten Bildpunkten nicht in Betracht gezogen werden)
  - Außerdem sind sämtliche Einheiten mit der vorigen Schicht auf dieselbe Weise, mit genau denselben Gewichten und derselben Struktur verbunden.
- Es werden kleine »Fenster« von Gewichten (sogenannte *Filter)* über Eingangsdaten (z.B. ein Bild) hinweg angewendet.
  - Dadurch wird der als *Konvolution* bezeichnete Vorgang ermöglicht.

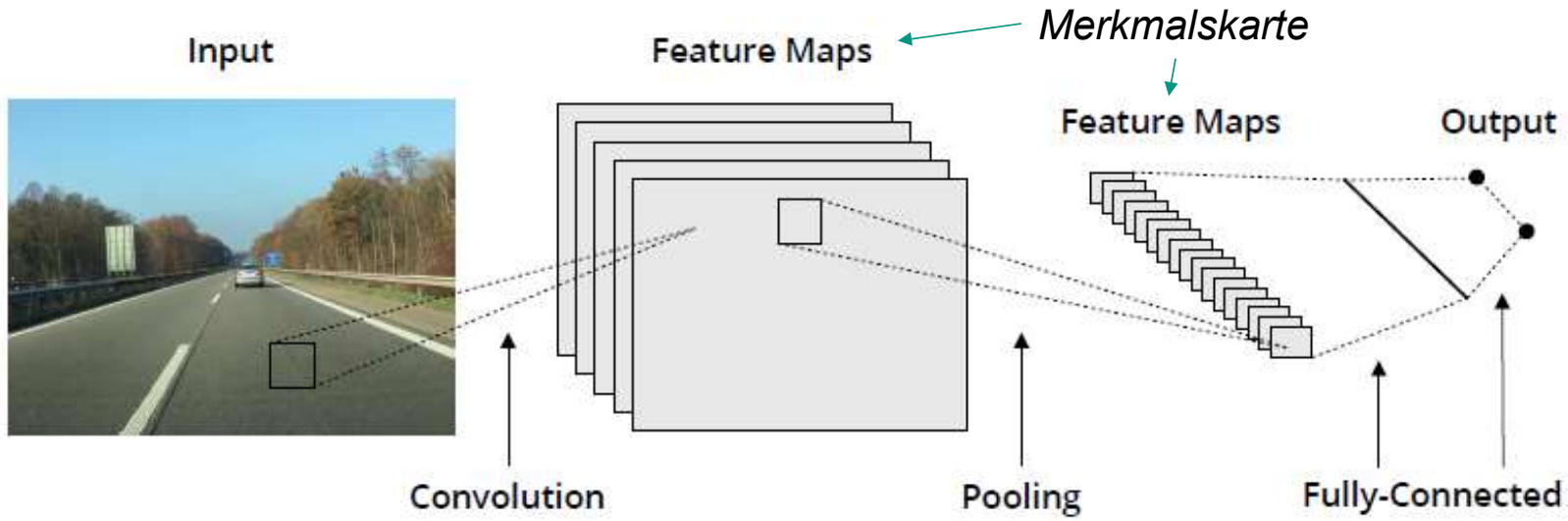




Convolutional Neural Networks (CNN)

Stand der Technik für Bilderkennung

- I.d.R.liegt die Eingabe als zwei- oder dreidimensionale Matrix (z.B. die Pixel eines Bildes) vor.
  - Dementsprechend sind die Neuronen im Convolutional Layer angeordnet.
- Die Aktivität jedes Neurons wird über eine diskrete Faltung (Konvolution von lateinisch „convolvere“ → „zusammenrollen“) berechnet.
  - Convolution-Operation (Faltung) für die Extraktion von räumlichen Merkmalen
  - Pooling-Operationen für die Dimensionsreduzierung
  - Fully-Connected-Schichten für die Klassifizierung auf Basis der extrahierten Merkmalen



Faltung

Erste Iteration

- Filter zu entwerfen geschieht über die Faltung unter Verwendung einer Faltungsmatrix (als Maske oder Kern bezeichnet).
- Das Filter betrachtet das Bild, auf welches es angewendet wird, als zweidimensionale Matrix, auf die der Kern angewendet wird.
  - Beschaffenheit und Inhalt des Kernes bestimmen, welcher Effekt bei der Anwendung erreicht wird.

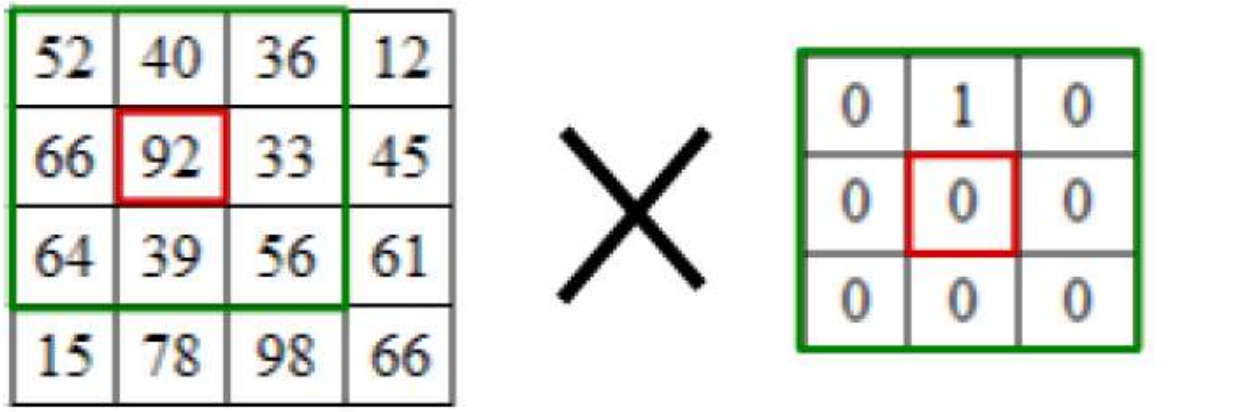
52	40	36	12
66	92	33	45
64	39	56	61
15	78	98	66

Originalmatrix
(z.B. Bild mit Grautönen)

Faltung

Erste Iteration

- Hier werden nur 3x3-Matrizen betrachtet, da diese weiter verbreitet sind als zum Beispiel 5x5 Matrizen und die wesentlichen Effekte mit ihnen realisiert werden können.
- Das Filter betrachtet nacheinander jedes Pixel des Bildes.
  - Für das aktive Pixel und seine 8 Nachbarpixel wird eine Multiplikation mit den entsprechenden Elementen des Kernes durchgeführt.
  - Die Ergebnisse dieser Multiplikationen werden als neuer Wert des aktuellen Pixels übernommen.



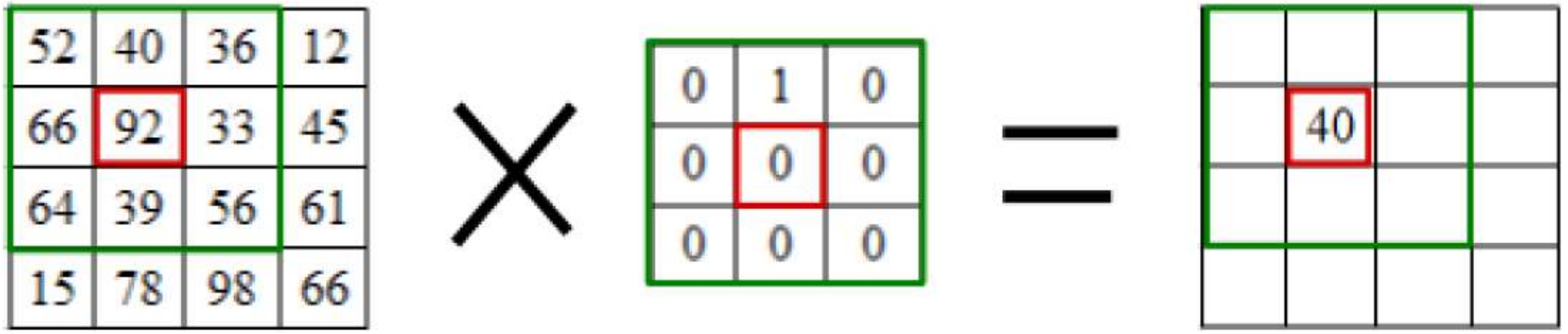
Originalmatrix
(z.B. Bild mit Grautönen)

Faltungsmaske (Kernel)

Faltung

Erste Iteration

- Hier werden nur 3x3-Matrizen betrachtet, da diese weiter verbreitet sind als zum Beispiel 5x5 Matrizen und die wesentlichen Effekte mit ihnen realisiert werden können.
- Das Filter betrachtet nacheinander jedes Pixel des Bildes.
  - Für das aktive Pixel und seine 8 Nachbarpixel wird eine Multiplikation mit den entsprechenden Elementen des Kernes durchgeführt.
  - Die Ergebnisse dieser Multiplikationen werden als neuer Wert des aktuellen Pixels übernommen.



Originalmatrix
(z.B. Bild mit Grautönen)

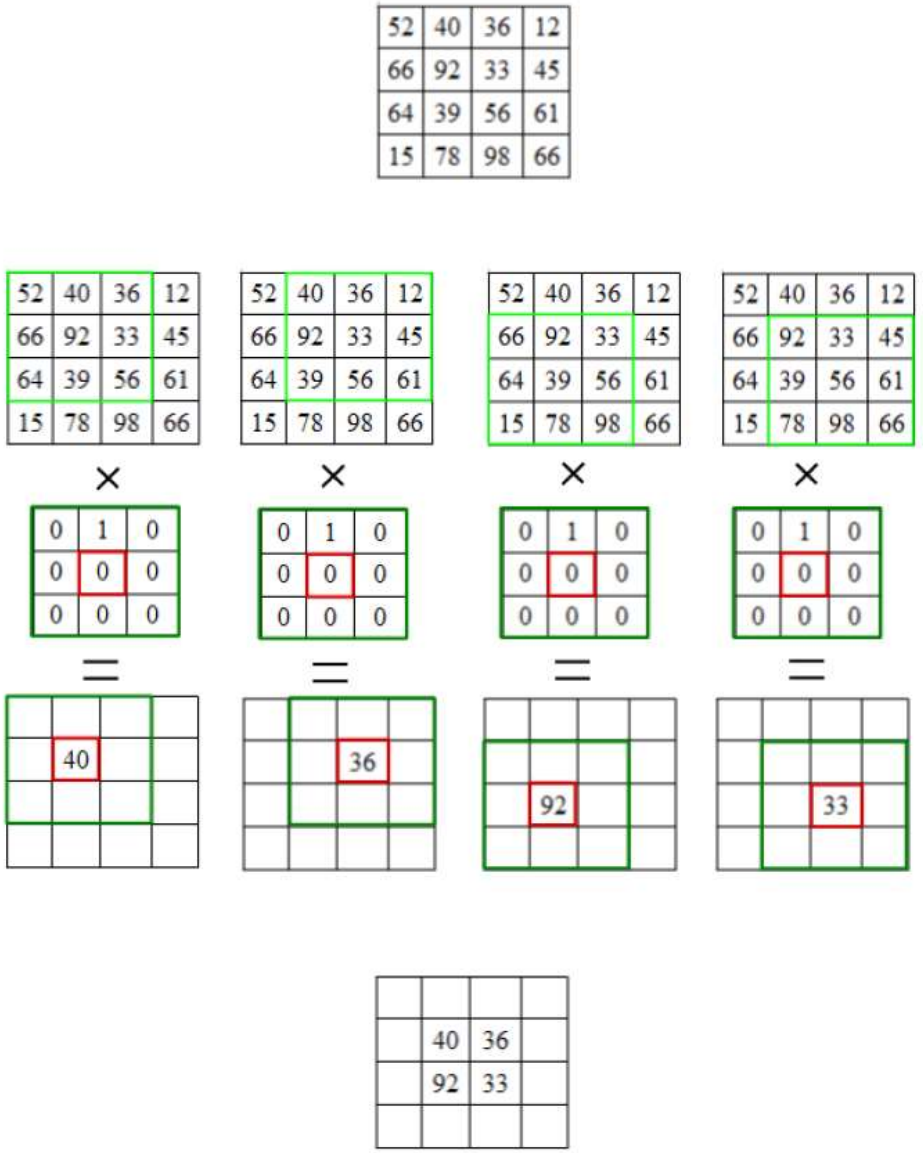
Faltungsmaske (Kernel)

Resultierende Matrix

Faltung

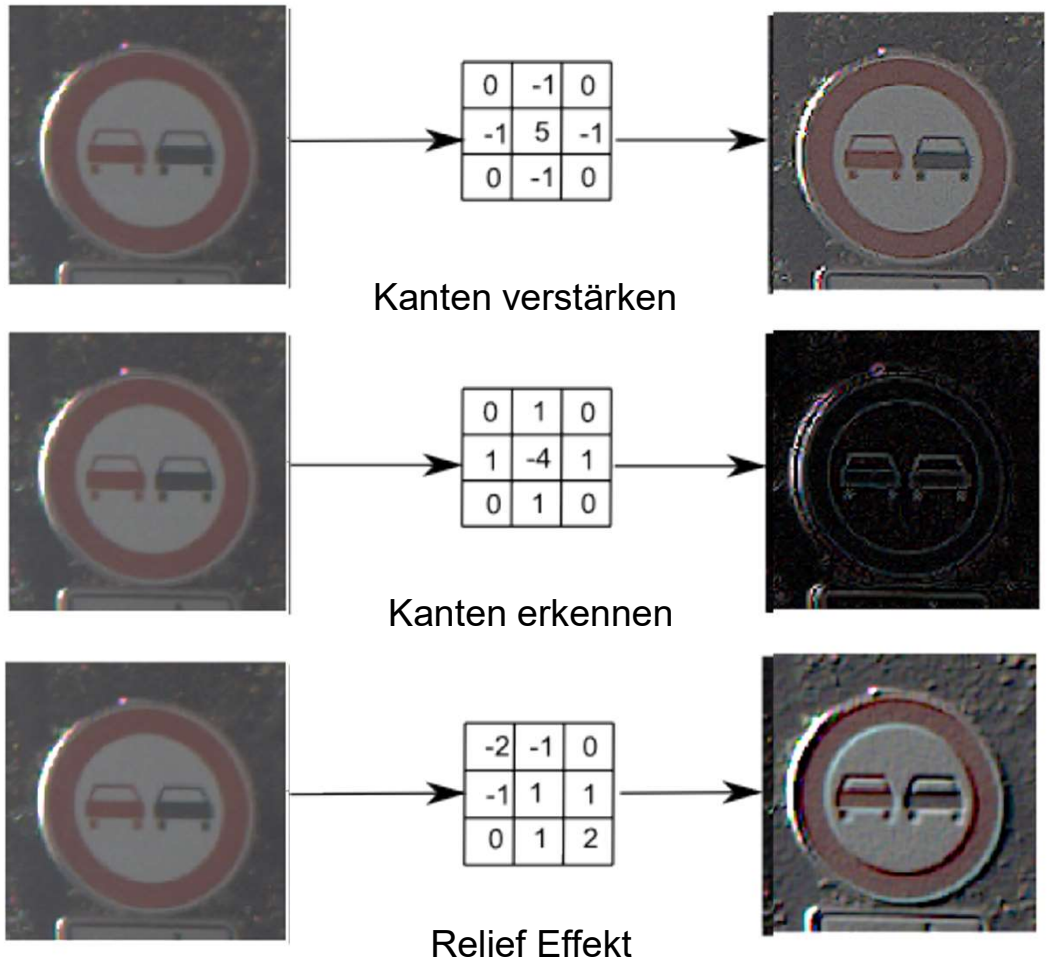
Filterung des kompletten Bildes

- Die Maske wird über alle Bildpositionen platziert.
  - Abhängig von der Kernelgröße besteht ein Bildrand von mindestens einer Pixelbreite
  - Diesen Pixeln am Rand kann kein Wert zugewiesen werden.
    - ► Dieser Rand kann mit verschiedenen Methoden behandelt werden, z.B kann der Rand mit Nullen oder mit den Werten benachbarter Pixel ausgefüllt werden (Padding).
- Es entsteht 1 Merkmalskarte gleicher Größe wie das Ausgangsbild (inkl. Padding)
- Mit verschiedenen Filtern entstehen verschiedene Merkmalskarten.



Beispiele von Konvolutions-Effekten

- Durch das Verwenden unterschiedlicher Masken entstehen mehrere Bildvarianten mit verschiedenen Effekten.

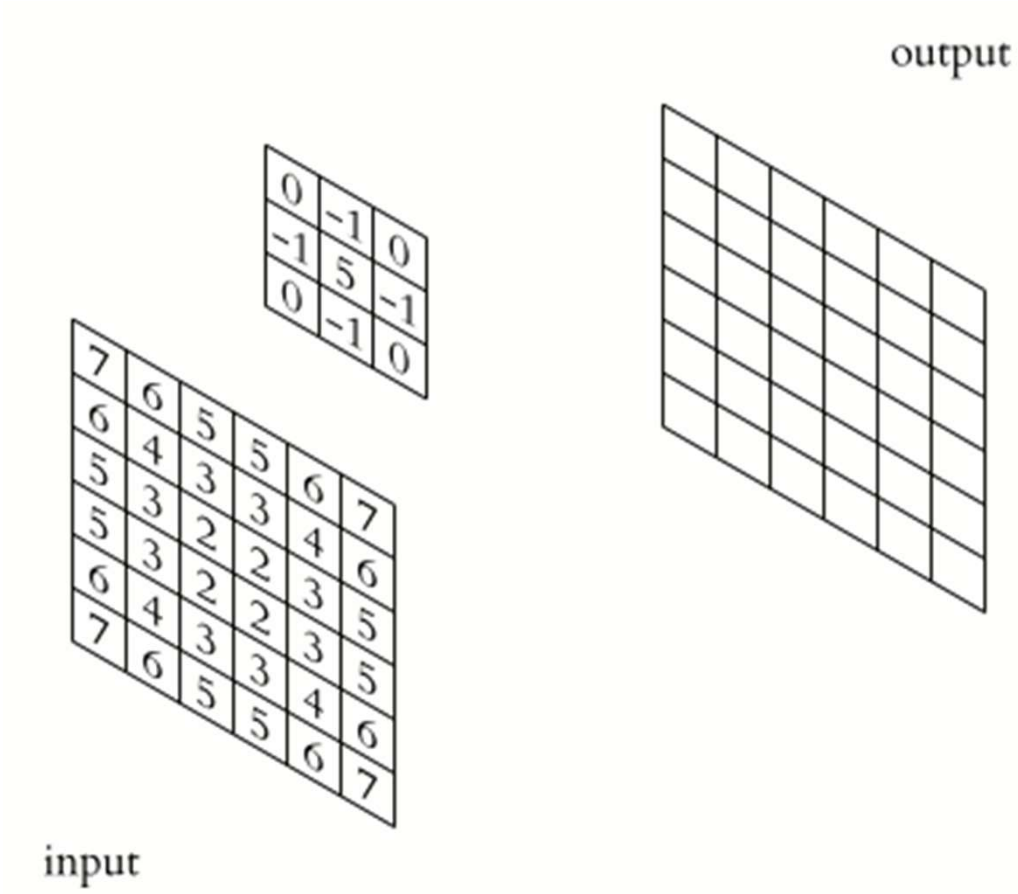


Beispiele von Konvolutions-Effekten

- Durch das Verwenden unterschiedlicher Masken entstehen mehrere Bildvarianten mit verschiedenen Effekten.
  - Kantendetektion
  - Scharfzeichnung
  - etc.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

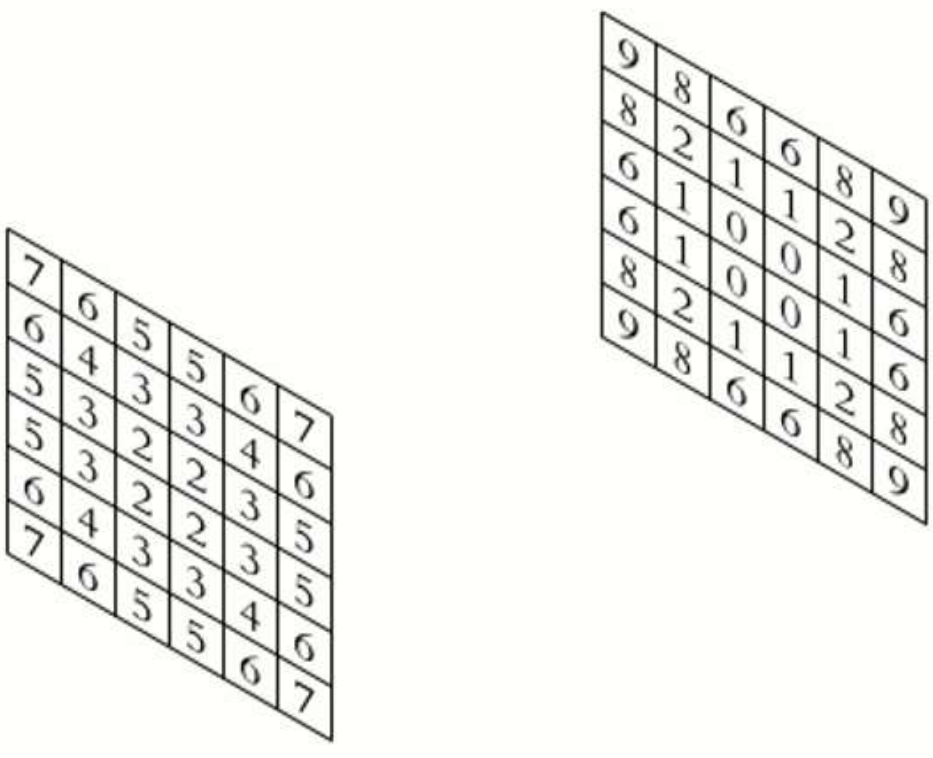
Padding
Inklusive Sharpening



- Zur Behandlung der Randregionen des Inputs existieren verschiedene Padding-Methoden.
- *S. 3 Folien zuvor: „Dieser Rand kann mit verschiedenen Methoden behandelt werden, z.B kann der Rand mit Nullen oder mit den Werten benachbarter Pixel ausgefüllt werden.“*
- Unter Padding versteht man den Innenabstand des Elements.
  - Das ist der Zwischenraum zwischen dem Inhalt eines Elements und der Elementgrenze.

Padding
Inklusive Sharpening

- Scharfzeichnen

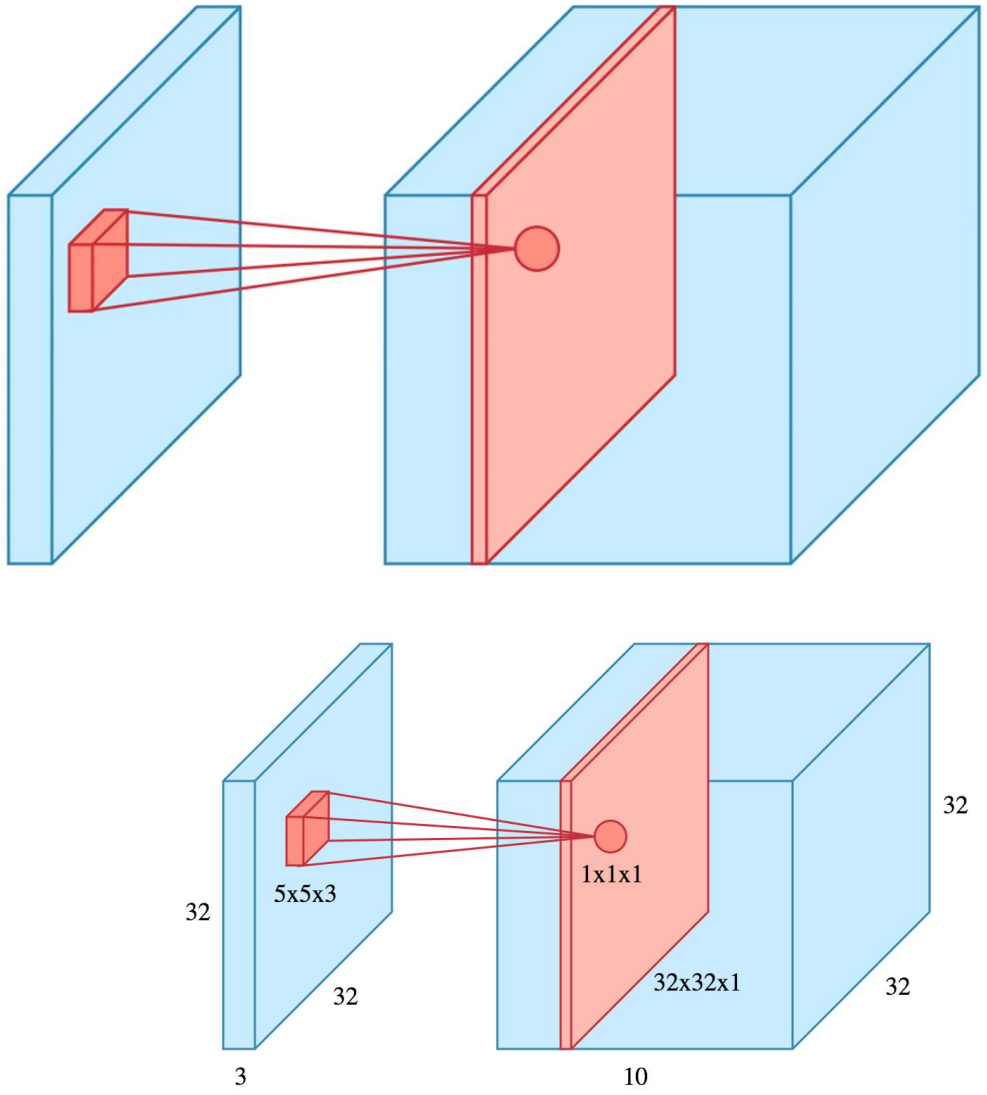


- Zur Behandlung der Randregionen des Inputs existieren verschiedene Padding-Methoden.
- *S. 3 Folien zuvor: „Dieser Rand kann mit verschiedenen Methoden behandelt werden, z.B kann der Rand mit Nullen oder mit den Werten benachbarter Pixel ausgefüllt werden.“*
- Unter Padding versteht man den Innenabstand des Elements.
  - Das ist der Zwischenraum zwischen dem Inhalt eines Elements und der Elementgrenze.

3D vs. 2D

- In der Praxis werden Bilder mit 3D Werten für Höhe, Breite und Farbtiefe (RGB → x3) dargestellt.
- Der Unterschied ist, dass man jetzt in der Matrixkalkulation 3D an Stelle von 2D verwendet.
  - Das Ergebnis ist allerdings weiter Skalar.
- D.f. der Filter wird über die Matrixflächen verschoben.
  - Es entsteht jeweils 1 Merkmalskarte (hier 32x32x1) aus der Matrix 32x32x3
- Bei 10 Filtern entsteht ein Quader von 32x32x10, die große blaue Box rechts.
- Dass hier wieder Merkmalskarten 32x32 entstehen, hängt wieder mit dem Padding zusammen.

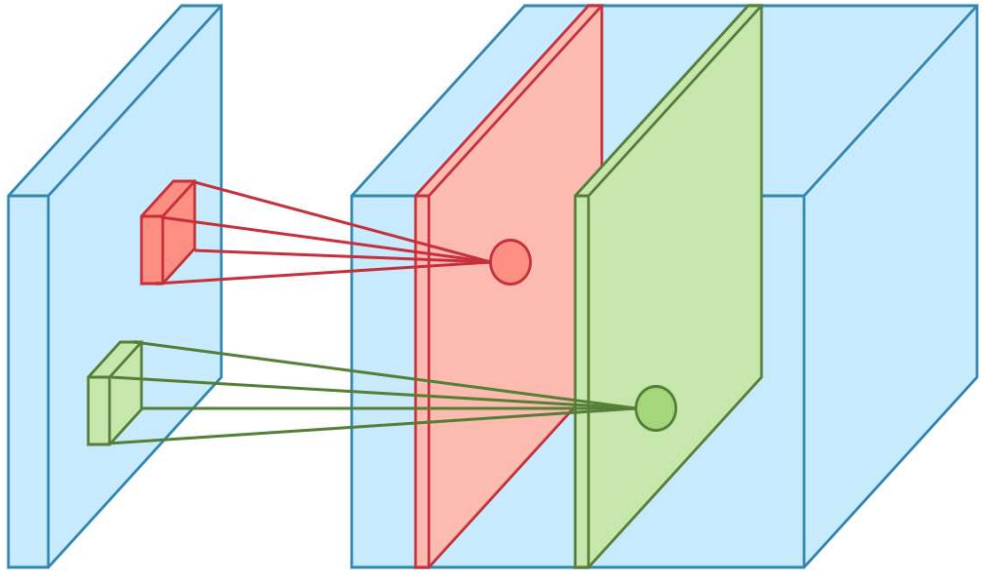
- Remark: *Die Animation findet repräsentativ für 4 Felder der Matrix statt. Real natürlich für alle Felder.*

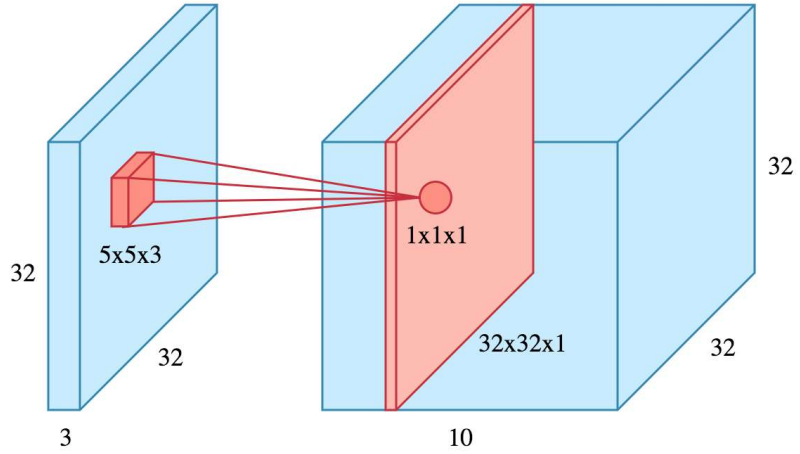


3D vs. 2D

- In der Praxis werden Bilder mit 3D Werten für Höhe, Breite und Farbtiefe (RGB → x3) dargestellt.
- Der Unterschied ist, dass man jetzt in der Matrixkalkulation 3D an Stelle von 2D verwendet.
  - Das Ergebnis ist allerdings weiter Skalar.
- D.f. der Filter wird über die Matrixflächen verschoben.
  - Es entsteht jeweils 1 Merkmalskarte (hier 32x32x1) aus der Matrix 32x32x3
- Bei 10 Filtern entsteht ein Quader von 32x32x10, die große blaue Box rechts.
- Dass hier wieder Merkmalskarten 32x32 entstehen, hängt wieder mit dem Padding zusammen.

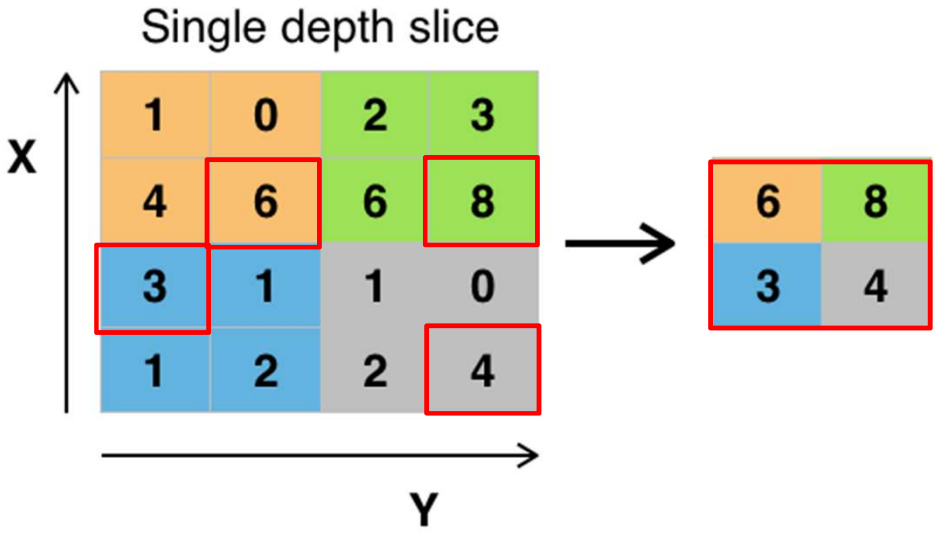
- Remark: *Die Animation findet repräsentativ für 4 Felder der Matrix statt. Real natürlich für alle Felder.*





Pooling

Verwerfen überflüssiger Informationen



- Zur Objekterkennung in Bildern ist die *exakte* Position einer Kante im Bild von vernachlässigbarem Interesse → die ungefähre Lokalisierung ist hinreichend.

- Es gibt verschiedene Arten des Poolings.
  - Am stärksten verbreitet ist das Max-Pooling, wobei aus jedem 2 × 2 Quadrat aus Neuronen des Convolutional Layers nur die Aktivität des aktivsten (daher "Max") Neurons für die weiteren Berechnungsschritte beibehalten wird.
  - Die Aktivität der übrigen Neuronen wird verworfen.
- Vorteile:
  - Verringerter Platzbedarf und erhöhte Berechnungsgeschwindigkeit
  - Daraus resultierende Möglichkeit zur Erzeugung tieferer Netzwerke, die komplexere Aufgaben lösen können.
  - Präventionsmaßnahme gegen Overfitting

Grenzen

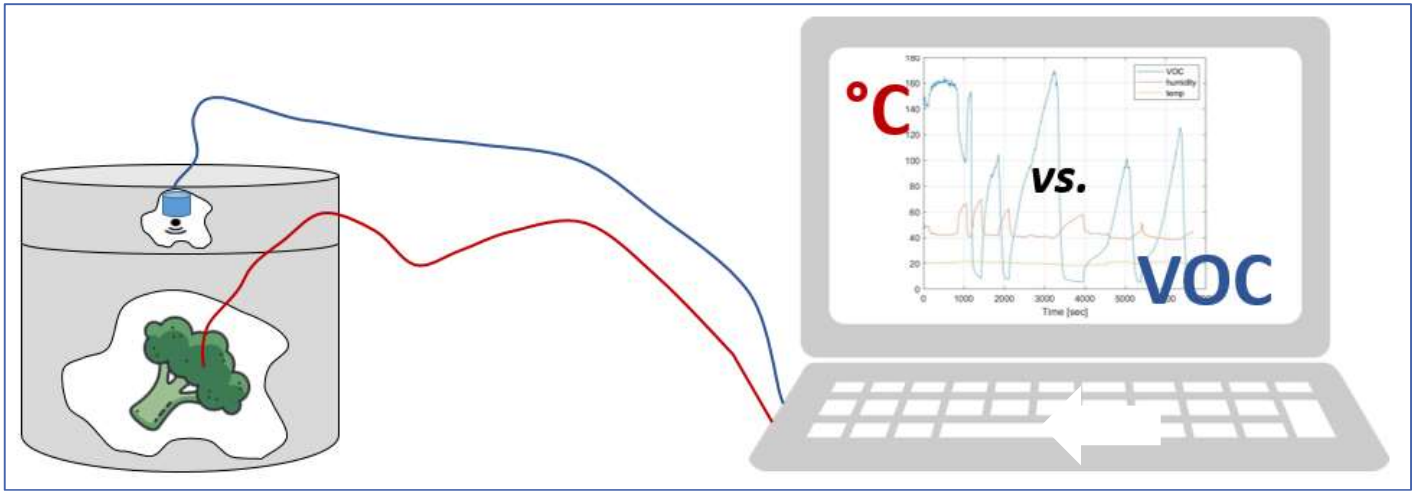
- Neuronale Netze sind in ihrem Einsatzgebiet und in ihrer Ausgestaltung sehr flexibel.
- Sie werden für Klassifizierungen, Prognosen und Clustering-Aufgaben angewendet.
- Insbesondere im Bereich der Bild-, Schrift- und Spracherkennung werden sie erfolgreich eingesetzt.
- Kritisch gesehen wird, dass neuronale Netze eine Black Box darstellen, deren Ergebnisse manchmal nur schwer zu erklären sind.
  - Die Gestaltung eines Netzwerkes wird oft als willkürlich angesehen und es besteht die Gefahr der Überanpassung (Overfitting) des Netzes an die Trainingsdaten.
  - Das Netz liefert dann perfekte Ergebnisse für die Trainingsdaten, stellt sich aber ungeeignet für „neue“ Daten heraus.
- Je komplizierter (tiefer) die Struktur der Netze ist, umso rechenaufwendiger werden die Verfahren.
- Dank der gestiegenen Rechenleistung aktueller Computer und immer neuer Deep-Learning-Bibliotheken erfahren neuronale Netze aber derzeit geradezu eine Renaissance in ihrer Anwendung.
  - *Bei aller Euphorie sollte man in der Erwartungshaltung aber auf dem Boden bleiben.*
- Man darf nicht vergessen:
  - Ein menschliches Gehirn besteht aus 80 bis 100 Milliarden Neuronen. Davon sind künstliche neuronale Netze noch „meilenweit“ entfernt.

WMF Group – a Groupe SEB company

EXPERIMENTAL SETUP

Recognition of the state of food

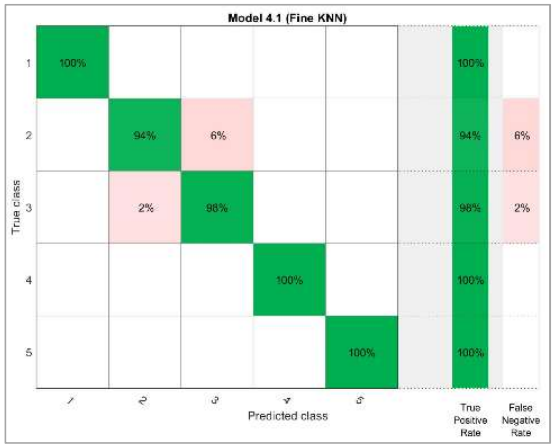
				
Tomato	**Pepper**	**Celery**	**Carrot**	**Apple**
2-3min	*3-5min*	*4-6min*	*5-8min*	*2-5min*
Washed & cut	*Washed & cut*	*Peeled & cut*	*Peeled & cut*	*Washed & cut*
				

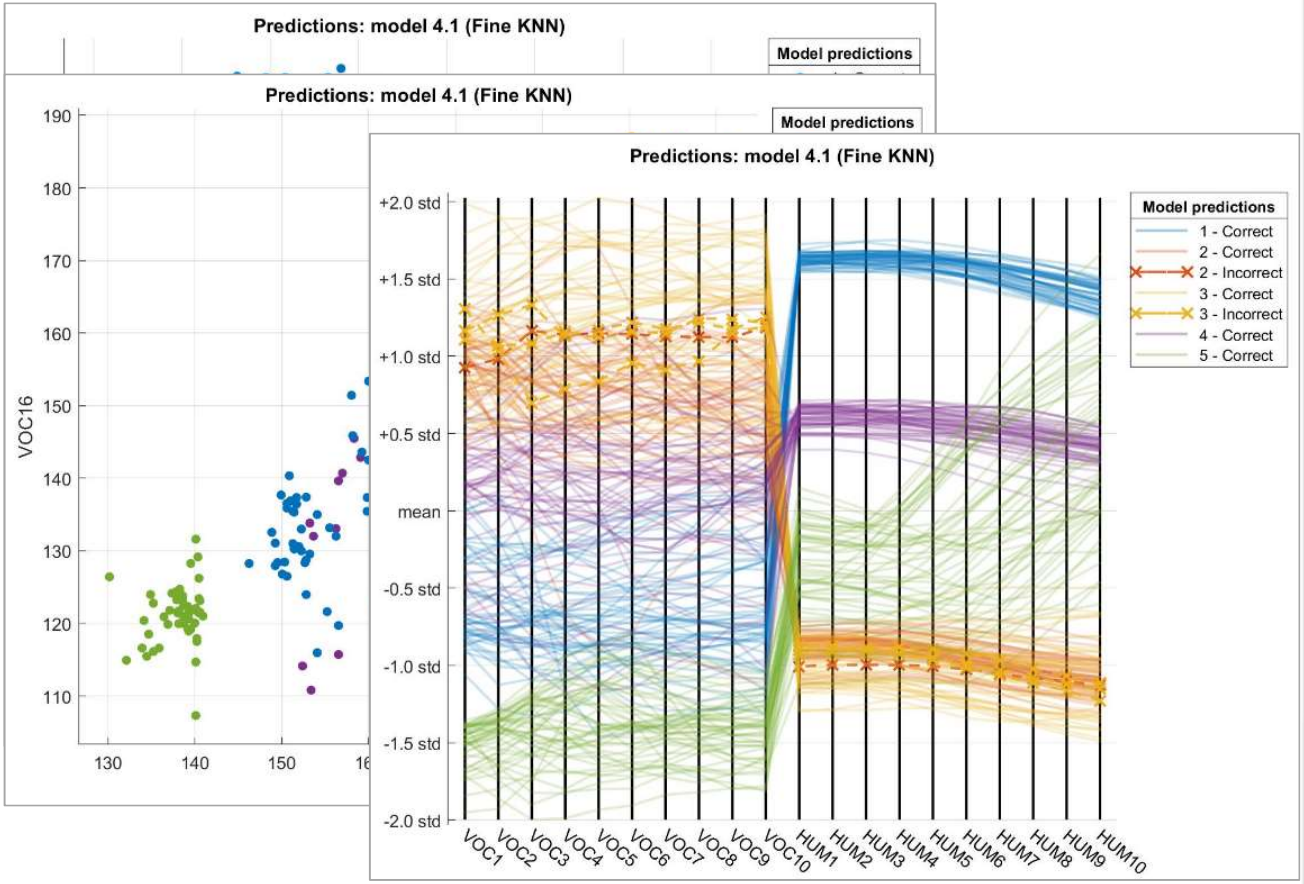


MODELLING OF ML-CONCEPT

Prediction results of classification learner

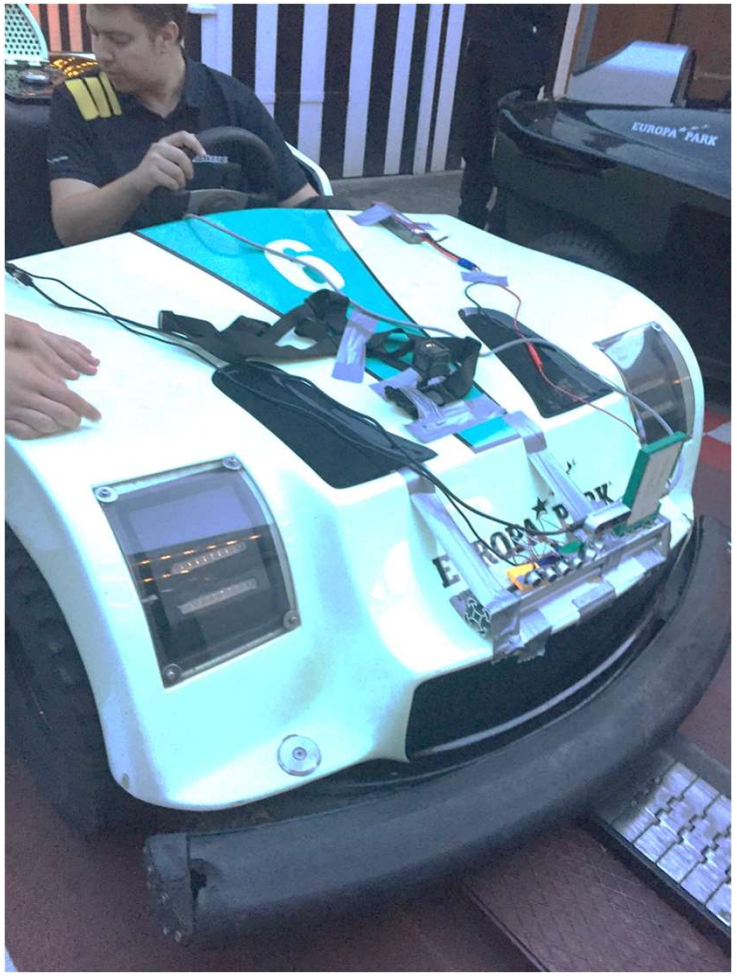
Model	Accuracy (PCA off)	Accuracy (PCA on)
Fine Tree	96.4%	88.4%
Medium Tree	96.4%	88.4%
Linear Discriminant	100%	90.4%
Quadratic Discriminant	99.6%	91.2%
Fine KNN	98.4%	92.4%
Medium KNN	96%	89.2%
Weighted KNN	98.4%	91.6%
Linear SVM	99.6%	90.8%
Quadratic SVM	100%	91.6%
Cubic SVM	99.6%	90.4%
Fine Gaussian SVM	96%	91.2%
Medium Gaussian SVM	99.2%	91.2%
Coarse Gaussian SVM	98.8%	71.6%
Subspace Discriminant	100%	87.6%






6


EUROPA PARK


EUROPA PARK

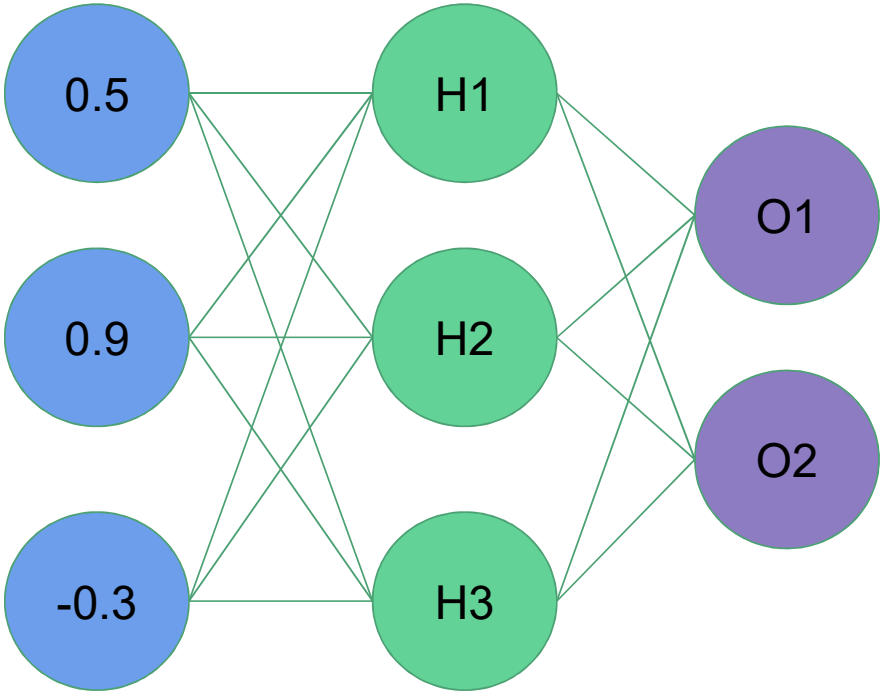
Backup

Beispiele

- Im Haushalt können Neuronale Netze bei Regelungsaufgaben wie der Benutzung einer Waschmaschine oder bei der Steuerung eines Staubsaugerroboters zum Einsatz kommen. Dieser kann mit Hilfe eines solchen Netzes seine Umgebung erlernen und kann bei jedem Saugvorgang seine abgefahrenen Pfade verbessern. Natürlich bleibt Ihre Wohnung nicht vollständig gleich. Hier wird einmal ein Tisch verrückt. Dort bekommt das Sofa einen neuen Standort. Da sich jedoch meist nicht alles gleichzeitig verändern wird, kann die Robotersteuerung durch ein Neuronales Netz die alten Kenntnisse, die sich noch verwenden lassen, weiter behalten und die neuen Kenntnisse zusätzlich lernen. Das heißt, mit Hilfe eines Neuronalen Netzes verblasst seine Erinnerung mit der Zeit und neue Informationen stellen sich in den Vordergrund, so dass er stets das momentan optimale Ergebnis liefert.

- Neuronale Netze können zur Optimierung einer Robotersteuerung eingesetzt werden. Industrieroboter z.B. bei der Produktion von Automobilen haben die Eigenschaft, dass sie aus einer Vielzahl von Gelenken bestehen, mit denen auch ein kompliziertes hineinreichen z.B. in eine Karosserie durchführbar ist. Die Gelenke können jedoch schnell unter Materialermüdung leiden, besonders wenn sie ruckartig bewegt werden und hohe Fliehkräfte wirken. Es geht darum, den Roboterarm möglichst schnell und möglich schonend zu bewegen, so dass alle Arbeiten geleistet werden können. Hierfür können neuronale Netze genutzt werden. Während des Betriebes verändert aufgrund von Abnutzungen der Roboterarm seine Eigenschaften. Eine Korrektur in dem Computerprogramm muss anhand verschiedener Parameter nachgehalten werden. Da sich die Parameter nur langsam ändern, kann auch hierfür ein Neuronales Netz genutzt werden.

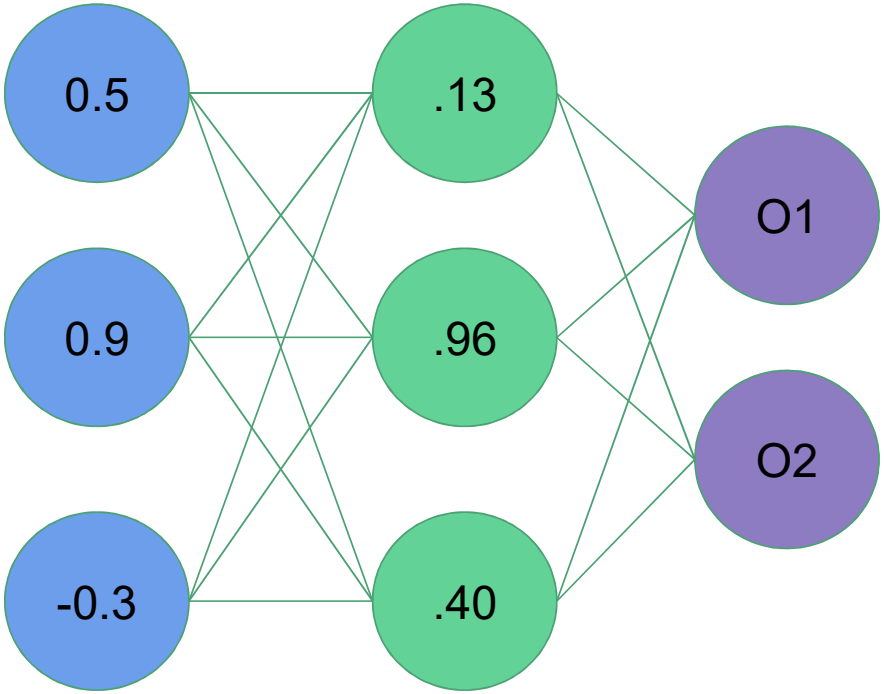
Inferenz
Neuronale Netze



H1 Weights = (1.0, -2.0, 2.0)
H2 Weights = (2.0, 1.0, -4.0)
H3 Weights = (1.0, -1.0, 0.0)

O1 Weights = (-3.0, 1.0, -3.0)
O2 Weights = (0.0, 1.0, 2.0)

Inferenz
Neuronale Netze

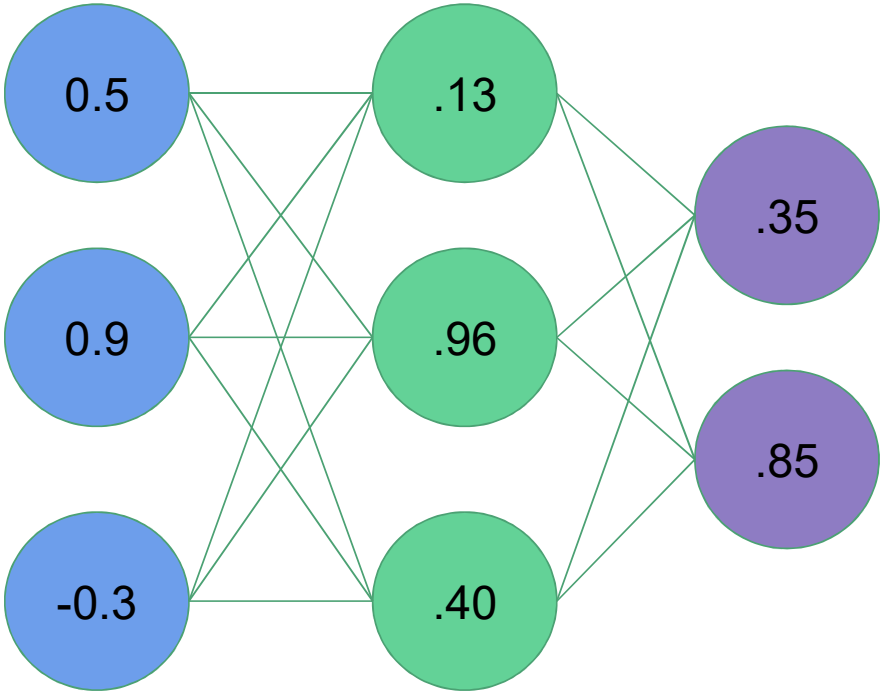


H1 Weights = (1.0, -2.0, 2.0)
H2 Weights = (2.0, 1.0, -4.0)
H3 Weights = (1.0, -1.0, 0.0)

O1 Weights = (-3.0, 1.0, -3.0)
O2 Weights = (0.0, 1.0, 2.0)

H1 = S(0.5 * 1.0 + 0.9 * -2.0 + -0.3 * 2.0) = S(-1.9) = .13
H2 = S(0.5 * 2.0 + 0.9 * 1.0 + -0.3 * -4.0) = S(3.1) = .96
H3 = S(0.5 * 1.0 + 0.9 * -1.0 + -0.3 * 0.0) = S(-0.4) = .40

Neuronale Netze
Inferenz



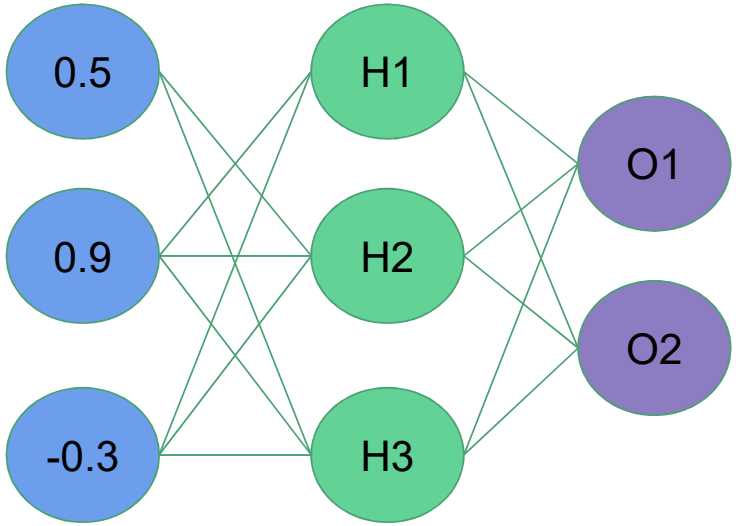
H1 Weights = (1.0, -2.0, 2.0)
H2 Weights = (2.0, 1.0, -4.0)
H3 Weights = (1.0, -1.0, 0.0)

O1 Weights = (-3.0, 1.0, -3.0)
O2 Weights = (0.0, 1.0, 2.0)

O1 = S(.13 * -3.0 + .96 * 1.0 + .40 * -3.0) = S(-.63) = .35
O1 = S(.13 * 0.0 + .96 * 1.0 + .40 * 2.0) = S(1.76) = .85

Matrix-Formulierung der Inferenz

Neuronale Netze



H1 Weights = (1.0, -2.0, 2.0)
H2 Weights = (2.0, 1.0, -4.0)
H3 Weights = (1.0, -1.0, 0.0)

Hidden Layer Weights | Inputs

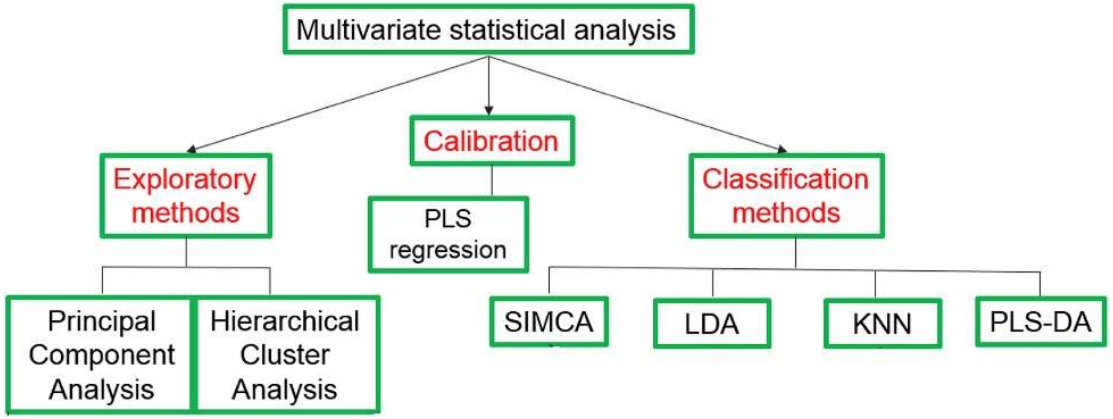
$$S\left(\begin{bmatrix} 1.0 & -2.0 & 2.0 \\ 2.0 & 1.0 & -4.0 \\ 1.0 & -1.0 & 0.0 \end{bmatrix} * \begin{bmatrix} 0.5 \\ 0.9 \\ -0.3 \end{bmatrix}\right) = S\left(\begin{bmatrix} -1.9 & 3.1 & -0.4 \end{bmatrix}\right) = \begin{bmatrix} .13 & .96 & 0.4 \end{bmatrix}$$

Hidden Layer Outputs

Unüberwachtes Lernen

„unsupervised learning“

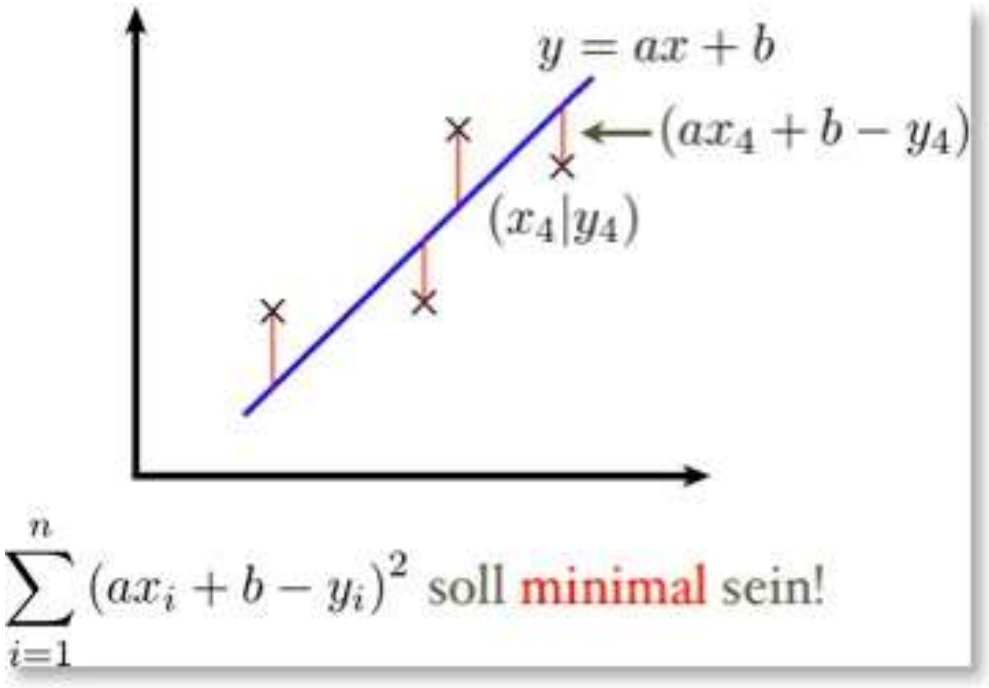
- Klassische Ballungen
  - k-means-Clustering
  - Agglomerative Hierarchical Clustering
- Begriffliche Ballungen („Conceptual Clustering“)
  - CLUSTER/2
  - Bildung von Begriffshierarchien („Concept Formation“)
- COBWEB
- CLASSIT
- Lernen durch Entdeckung
- BACON
- ABACUS
- ...
- Ohne Belohnung
- Ohne Vorwissen



Gauß'schen Methode der kleinsten Quadrate

Am Beispiel von 4 Messwerten

- Es soll eine passende Gerade (blau) gesucht werden.
- Nach einer Idee von **Carl-Friedrich Gauß** verwendet man als Maß für die Anpassung zunächst die vertikalen Abstände zwischen den tatsächlichen Messwerten und den entsprechenden Punkten auf der blauen Geraden.
- Für den vierten Punkt ist dieser Abstand angegeben.
- Diese Abstände werden quadriert und dann für alle Messwerte aufsummiert.
- Nach Gauß ist die beste Ausgleichslinie die, für die **die Summe der Quadrate der Abweichungen minimal ist**.
- Diese Idee heißt:
  - **Methode der kleinsten Fehlerquadrate**.
  - Das Stichwort „minimal“ deutet darauf hin, dass es sich um eine **Extremwertaufgabe** handelt.
  - Die Parameter a und b der Geraden müssen so gewählt werden, dass die bewusste Summe minimal wird.



Gütemaße für die Evaluation von Modellen

- Es existiert eine Reihe von Verfahren zum Testen der Güte statistischer Modelle.
- Die meisten solcher Verfahren liefern Kennzahlen, deren Quantität eine Aussage über die Modellgüte treffen. Ein Beispiel hierfür ist das Bestimmtheitsmaß R^2.
  - Dieses Maß erklärt den Anteil einer Variablen Y an der durch ein statistisches Modell erklärten Varianz.
  - Andere Gütetests beziehen sich auf die Beurteilung binärer Klassifikatoren.
  - Als Beispiel sei die Verifikation numerischer Wettervorhersagemodelle genannt.
  - An ihnen wird u.a. untersucht, wie genau ein vorhergesagter Parameter mit tatsächlich eingetretenen Ereignissen übereinstimmt (z.B. Niederschlag ja/nein).
  - Es werden Häufigkeitszahlen bestimmt, die angeben, wie oft eine Vorhersage bei eingetretenen Ereignissen korrekt war und wie oft inkorrekt.
  - Äquivalent dazu das Ganze für Nicht-Ereignisse.
  - Aus den erhaltenen vier Häufigkeiten lassen sich kategorische Gütemaße definieren.
  - Als Beispiele seien die Kennzahlen
    - Probability Of Detection
    - False Alarm Rate
    - True Skill Statisticsgenannt
  - Kontinuierliche Gütemaße beziehen sich dagegen auf kontinuierliche Parameter wie z.B. die Temperatur.
  - Hier findenübliche statistische Maße wiemittlerer Fehler,mittlerer absoluter Fehler,RMSE[1]unddieStandardabweichung Anwendung
  - Die Methodik, von der einige Verfahren erwähnt wurden, bezieht sich u.a. immer auf die Analyse eines kompletten Datensatzes, auf dem ein statistisches Modell beruht.
- Einen anderen Ansatzverfolgt dagegen die Kreuzvalidierung.
  - Ein Datensatz wird in mehrere Teile aufgeteilt.
  - Auf Basis eines Teils des Datensatzes wird ein statistisches Modellabgeleitet.
  - Das Modell wird auf den Rest des Datensatzes angewendet und die Abweichung zu den tatsächlichen Werten untersucht.
  - Das Verfahren dient also in erster Linieder Überprüfung derPrognosegüteeines Modells.

Gütemaße für die Evaluation von Modellen

Regression

- **R^2- Metric (R-Squared)**
  - Der Wert der R^2 kann Werte zwischen 0 und 1 annehmen
  - Nahe 1: Vorhersagen des Models sind gut
  - Nahe 0: Vorhersagen sind nicht besser als raten

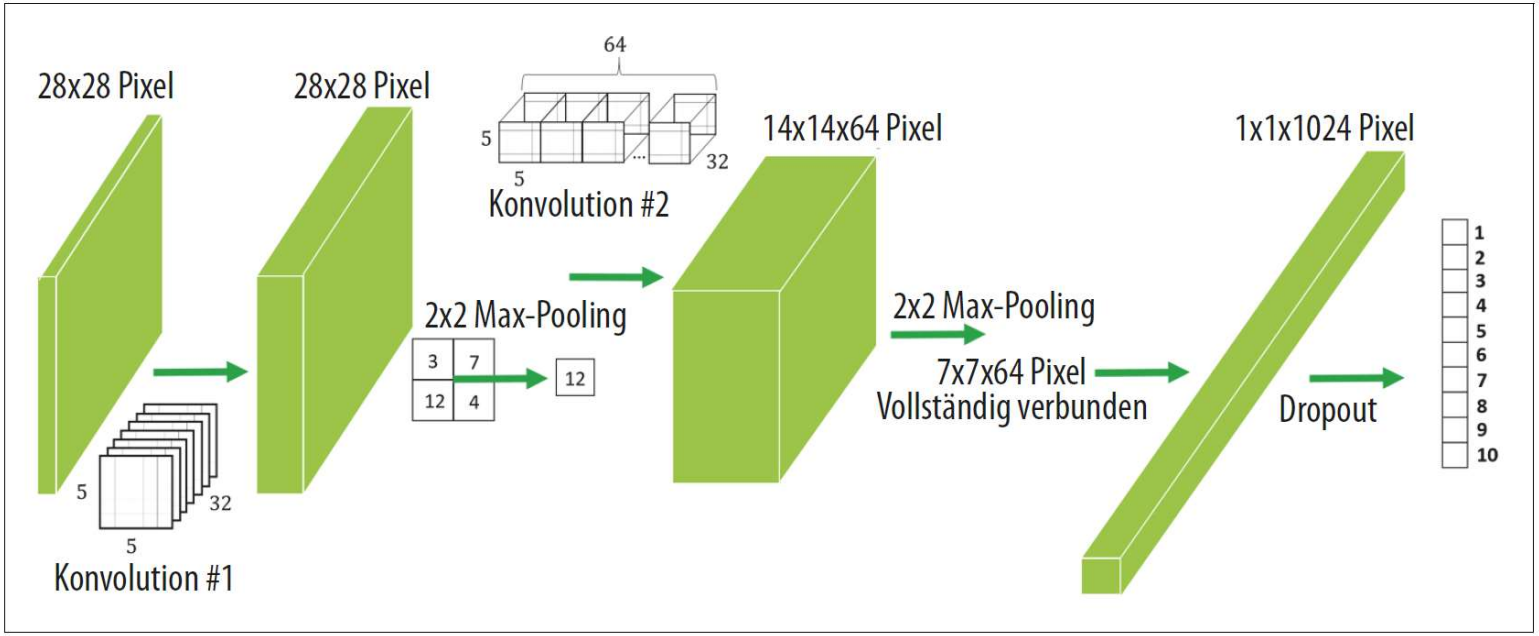
- $R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$

$y_i = vorhergesagte\ Werte$

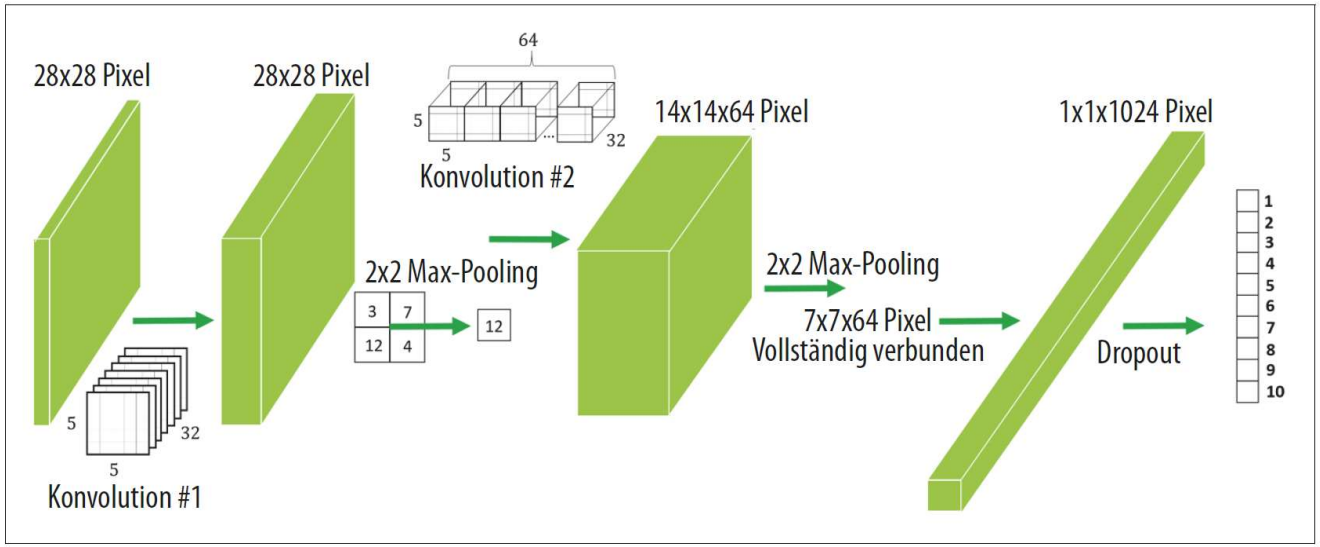
$\tilde{y}_i = korrekte\ Werte$

$\bar{y} = Median\ der\ Werte$

- Bilddaten in einem 2-D-Format mit der Größe 28 × 28 × 1 liegen vor.
- Als nächstes haben wir zwei aufeinanderfolgende Schichten für Konvolution und Pooling, jede davon mit 5 × 5 Konvolutionen und 64 Merkmalskarten, gefolgt von einer einzelnen vollständig verbundenen Schicht mit 1024 Einheiten.
- Vor dem Anwenden der vollständig verbundenen Schicht reihen wir alle Bildpunkte wieder zu einem einzelnen Vektor auf, da ihre räumliche Bedeutung von der vollständig verbundenen Schicht nicht mehr benötigt wird.

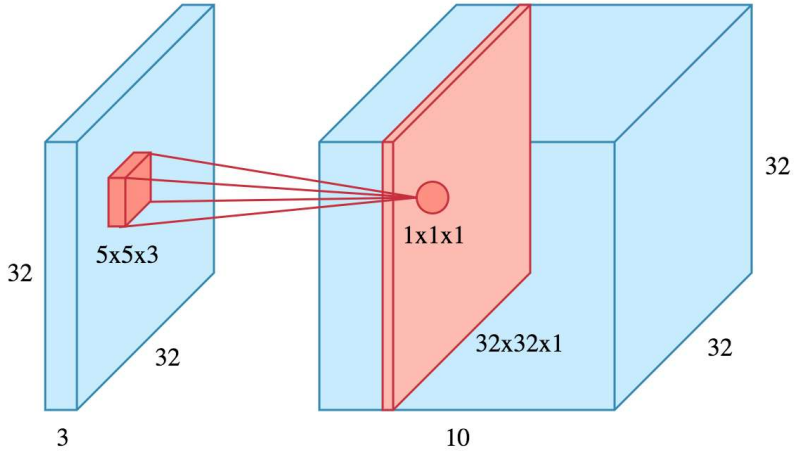


- Beachten Sie, dass die Größe des Bilds nach den zwei Konvolutions- und Pooling-Schichten 7 × 7 × 64 beträgt. Das ursprüngliche Bild mit 28 × 28 Pixeln wird zuerst auf die Größe 14 × 14 reduziert und anschließend von den Pooling-Operationen auf 7 × 7 verdichtet. Die 64 ist die Anzahl der in der zweiten Konvolutionsschicht erzeugten Merkmalskarten. Die Gesamtzahl der im Modell angelernten Parameter, von der sich ein Großteil in der vollständig verbundenen Schicht befindet (von 7 × 7 × 64 auf 1024), beträgt 3,2 Millionen. Ohne Max-Pooling wäre diese Zahl 16 Mal so groß (28 × 28 × 64 × 1024 oder etwa 51 Millionen).
- Die Ausgabe schließlich ist eine vollständig verbundene Schicht mit 10 Einheiten, was der Anzahl der Kategorien entspricht.



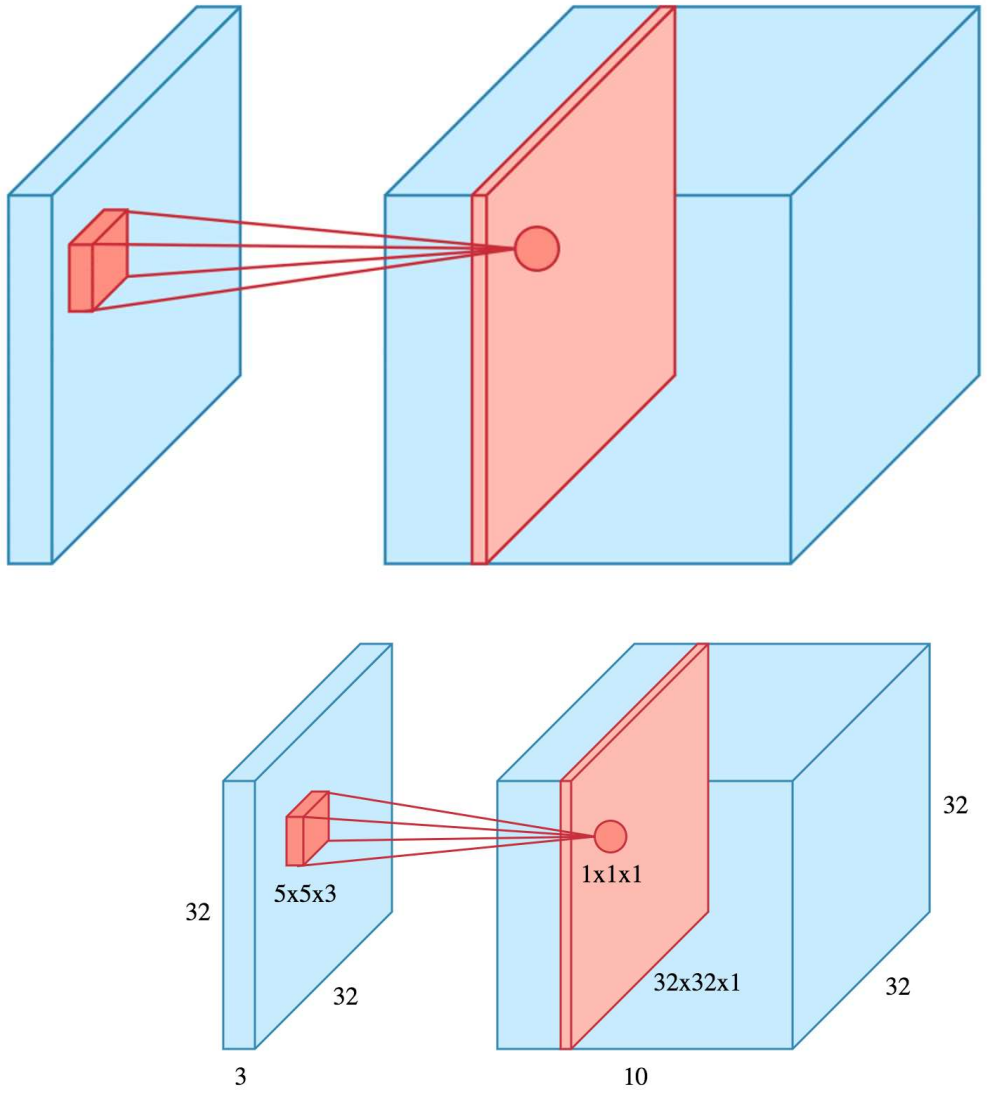
- But in reality these convolutions are performed in 3D.
- In reality an image is represented as a 3D matrix with dimensions of height, width and depth, where depth corresponds to color channels (RGB).
- A convolution filter has a specific height and width, like 3x3 or 5x5, and by design it covers the entire depth of its input so it needs to be 3D as well.
- One more important point before we visualize the actual convolution operation.
  - We perform multiple convolutions on an input, each using a different filter and resulting in a distinct feature map.
  - We then stack all these feature maps together and that becomes the final output of the convolution layer.
  - But first let's start simple and visualize a convolution using a single filter.

- Let's say we have a 32x32x3 image and we use a filter of size 5x5x3 (note that the depth of the convolution filter matches the depth of the image, both being 3).
- When the filter is at a particular location it covers a small volume of the input, and we perform the convolution operation described above.



3D vs. 2D

- The only difference is this time we do the sum of matrix multiply in 3D instead of 2D, but the result is still a scalar.
- We slide the filter over the input like above and perform the convolution at every location aggregating the result in a feature map.
- This feature map is of size 32x32x1, shown as the red slice on the right.
- If we used 10 different filters we would have 10 feature maps of size 32x32x1 and stacking them along the depth dimension would give us the final output of the convolution layer: a volume of size 32x32x10, shown as the large blue box on the right.
- Note that the height and width of the feature map are unchanged and still 32, it's due to padding and we will elaborate on that shortly.
- To help with visualization, we slide the filter over the input as follows. At each location we get a scalar and we collect them in the feature map. The animation shows the sliding operation at 4 locations, but in reality it's performed over the entire input.



Filter („kleine Fenster von Gewichten“)

Konvolution einer 5 x 5 Bildmatrix mit einer 3x3 Filtermatrix