Fallstudie 27: Maschinelles Lernen

# Was versteht man unter maschinellem Lernen?

Unter maschinellem Lernen werden Algorithmen zusammengefasst, welchen unter Verwendung eines Training-Sets (Lerndaten), also bestehenden Erfahrungswerten, neue Informationen generieren und diese anschließend verallgemeinern können. Dafür wird als Ergebnis des Lernalgorithmus ein Modell erzeugt, welches alle gewonnen und erzeugten Informationen umfasst und dazu genutzt werden kann diese auf weitere Daten anzuwenden und Vorhersagen (bspw. Klassifizierung) über diese zu treffen. Hierbei ist die Menge und Qualität der im Training-Set enthaltenen Daten von großer Bedeutung. Der Algorithmus lernt also mit zunehmender Erfahrung hinzu und kann somit seine Leistung bzw. Genauigkeit der Vorhersagen verbessern.

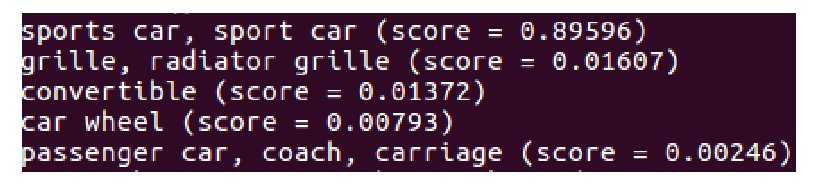


Abbildung - Beispielhafter Vorhersagen-output einer Anwendung zur Bilderkennung

Ein großer Vorteil von maschinellem Lernen ist, dass damit im Vergleich zu linearen Vorgehensweisen deutlich größere Datensätze ausgewertet werden und dabei wesentlich mehr Variablem berücksichtigt werden können.

Der heute wohl meistverwendete Methode bzw. Vorgehensweise von maschinellem Lernen ist Deep Learning, was auf künstlichen neuronalen Netzen basiert.



Abbildung 1 - Typischer Ablauf eines Maschinellen Lernalgorithmus

Diese Algorithmen werden häufig für Klassifizierungs oder Clusterung Aufgaben verwendet. Maschinelle Lernalgorithmen sind hierfür besonders gut geeignet, da sie in der Lage sind neue Muster innerhalb der Lerndaten zu erkennen und diese dann auf unbekannte Daten anzuwenden.

Das besondere an diesen Algorithmen ist, dass sie nicht speziell darauf programmiert werden müssen eine spezifische Aufgabe auszuführen oder ein spezifisches Training-Set auszuwerten. Sie lernen stattdessen selbst dazu und passen sich dementsprechend an die aktuelle Aufgabe an.

## Kriterien zum sinnvollen Einsatz von maschinellem Lernen

Auch wenn auf maschinellem Lernen basierende Lösungsansätze oftmals deutlich bessere Ergebnisse erzielen als vergleichbare lineare Vorgehensweißen sollten und können sie nicht blind auf alle Probleme angewendet werden. Die Problemstellung sollte auf eine Reihe von Kriterien überprüft werden bevor maschinelles Lernen darauf angewendet wird.

### Messbarkeit der Performance

Die Performance bzw. der Erfolg der Berechnungen muss auf eine sinnvolle Art messbar sein. Ziel des Algorithmus ist es eine definierte Erfolgsmetrik zu optimieren. Wenn der Output nicht klar evaluiert werden kann wird der Algorithmus sich auch nicht verbessern können. Der Algorithmus muss wissen wie weit sein Ergebnis von der korrekten Lösung entfernt ist um dahingehend zu optimieren. Dies ist insbesondere bei überwachtem Lernen von Bedeutung, da hier beim Lernprozess bereits die korrekten Lösungswerte mitgeliefert werden.

### Genügend Trainingsdaten

Bevor Aussagen über neue Daten getroffen werden können muss zuerst ein dementsprechendes Modell erzeugt werden. Um ein aussagekräftiges Modell zu generieren wird eine gewisse Menge an Trainingsdaten benötigt. Grundsätzlich lässt sich sagen, dass umso mehr Daten zur Verfügung stehen, umso besser das Modell wird.

### Qualität der Trainingsdaten

Eine große Menge an Daten ist jedoch nicht ausreichend um ein qualitativ hochwertiges Modell zu garantieren. Die Trainingsdaten sollten möglichst unterschiedlich voneinander sein und ein möglichst breites Spektrum der möglichen Ausprägungen darstellen. Wenn sich alle Datensätze sehr ähneln wird das Modell nur mit wenig Erfolg auf Datensätze mit anderen Ausprägungen angewendet werden können.

### Klar definierte Problemstellung

Zuletzt ist es noch wichtig zu beachten, dass das Problem an sich gut durch maschinelle Lernansätze lösbar ist. Diese funktionieren besonders dann gut, wenn Input und Output gut angegeben werden können. Sei X gegeben soll Y ausgegeben werden. Außerdem sind diese Ansätze besonders geeignet wenn Korrelationen in solch großen Datenmengen gesucht werden sollen, dass ein Mensch diese nicht erkennen könnte.

# Geschichtliche Entwicklung

# Grundbegriffe

### Ground truth

Die ground truth im Gebiet des maschinellen Lernens stellt das korrekte/gemessene Ergebnis dar. Bei einem Algorithmus zur Handschriftenerkennen von Zahlen ist die ground truth beispielsweise der Wert der zu erkennende Zahl.



Abbildung 2 - Handgeschriebene Ziffer "7"

In dem Beispiel in Abbildung 1 soll die Ziffer „7“ erkannt werde, somit wäre die ground truth hier ebenfalls die 7.

### Backpropagation

## Überwachtes(supervised) learning:

Überwachtes Lernen wird beispielsweiße für Klassifizierungsprobleme eingesetzt, da hierbei die Klassen bereits vorher bekannt sind und jedem Datensatz genau eine Klasse zugewiesen werden kann.

Beim überwachten lernen erhalten die Datensätze des Training-Sets jeweils ein Label, welches die korrekte Zuordnung des Datensatzes zu einer gegeben Klasse darstellt. Der Datensatz für das Beispiel in Abbildung 1 würde das Label „7“ erhalten, da eben die Ziffer „7“ erkannt werden soll. In den meisten Fällen entspricht das Label der ground truth.

Ziel des Algorithmus ist es, Zusammenhänge und Muster zwischen den restlichen Informationen des Datensatzes und dem Label zu erkennen und daraus ein Modell zu bilden, welches dafür genutzt werden kann um dieses Wissen auf Datensätze zu extrapolieren, welche kein Label enthalten. Dabei ist es wichtig, dass der Algorithmus nicht einfach sämtliche Datensätze „auswendig“ lernt, da er ansonsten bei unbekannten Datensätzen, welche nicht Teil des Training-Sets sind nur wenig erfolgreich ist.

## Unüberwachtes(unsupervised) learning:

Das unüberwachte Lernen hingegen, wird hauptsächlich für Clusteringprobleme eingesetzt bei denen die einzelnen Cluster/Gruppen nicht im Vorfeld bekannt sind. Dies bedeutet, dass hierbei im Gegensatz zum überwachten Lernen kein Label in den Datensätzen enthalten ist. Der Algorithmus muss sich die Gruppen selbst erstellen und die einzelnen Datensätzen selbst dazu zuordnen. Dafür muss der Algorithmus selbst, oft für Menschen nicht offensichtliche, Muster und Zusammenhänge zwischen den einzelnen Datensätzen erkennen. Es gibt also keine „richtige“ Lösung und das Ergebnis des Algorithmus kann im Vorfeld nicht bestimmt bzw. vorhergesagt werden.

Eine weit verbreitete Implementierung dieser Vorgehensweise zum Clustering ist beispielsweise der k-means Algorithmus.

# Anwendungsgebiete

Weit verbreitete Umsetzungen von maschinellem Lernansätzen sind beispielsweise Support-Vector-Machines(SVM), künstliche neuronale Netze (KNN) mit dem Untergebiet Deep Learning oder genetische Algorithmen.

## Support Vector Machines

## Künstliche neuronale Netze

### Grundbegriffe

#### Neuronen

Neuronen sind der wichtigste Bestandteil eines neuronalen Netzes. Ihre Aufgabe ist es eine Reihe von Eingabewerten zu verarbeiten und auf einen einzelnen Ausgabewert abzubilden. Neuronen werden nach der Art der verwendetet activation function (Siehe 3.4) unterschieden.

ine spezielle Art von Neuron ist der Perzeptron, dieser bildet binäre Eingabedaten auf einen binären Ausgabewert ab. Der Nachteil hiervon ist, dass eine kleine Veränderung der Eingabedaten das Ergebnis komplett von 0 zu 1 ändern kann. Bei neuronalen Netzen ist dieses Verhalten nicht erwünscht, weshalb Neuronen verwendet werden die mehrere Ausgabewerte liefern können.

#### Gewichte & Bias

Bei den Gewichten handelt es sich um Stellschrauben eines jeden Neurons, die das Netz zum Einlernen von Aufgaben benötigt. Jedes Neuron besitzt für jeden Eingabewert ein separates Gewicht. Durch den Wert dieser Gewichte können nutzvolle von nutzlosen Informationen unterschieden werden. Sollte beispielsweiße die Aufgabe eines Neurons sein ein Stoppschild zu erkennen und die Eingabewerte stellen unterschiedliche Farben dar, so ist nur die Information relevant ob das Schild rot ist. Das Neuron, das angibt ob das Schild rot ist oder nicht würde eine höhere Gewichtung bekommen als die anderen Neuronen.

Der Bias ist eine weitere Stellschraube eines jeden Neurons, im Gegensatz zu den Gewichten jedoch besitzt jedes Neuron nur einen Bias. Der Bias ist eine Konstante, die auf alle Eingabewerte aufaddiert wird.

Sollte von einem Neuron beispielsweiße der Ausgabewert ~1 erwünscht sein falls alle Eingabewerte schwarz sind (bei RGB-Kodierung = 0) so ist dies ohne die Verwendung eines Bias nicht möglich (siehe Abbildung 3.1-1). Der Ausgabewert würde immer 0.5 sein, da alle Eingabewerte 0 sind. Durch das addieren eines Bias können wir den Ausgabewert näher an die geforderte 1 bringen.

#### Aktivierungsfunktion

Die activation function ist ein zentraler Bestandteil eines jeden Neurons des hidden layers und die Wahl der richtigen Funktionen ist von großer Bedeutung für die Leistung des Netzes.

Die activation function dient dazu, eine Reihe von Eingabewerten auf einen Zielbereich abzubilden. Die Eingabewerten bestehen aus den Werten der Neuronen des vorherigen Layers inkl. der Gewichtungen und dem Bias.

Das Besondere daran ist, dass es sich um eine nichtlineare Funktion handelt, dadurch wird das gesamte Netz nichtlinear was zu einer höheren Vielzahl an möglichen Abbildungen führt. Ohne die Funktion würden die Neuronen lediglich lineare Transformationen durch das weitere Multiplizieren mit den Gewichten und dem Bias durchführen. Dies ist jedoch nicht ausreichend um komplexere Aufgaben zu lösen.

### Deep Learning

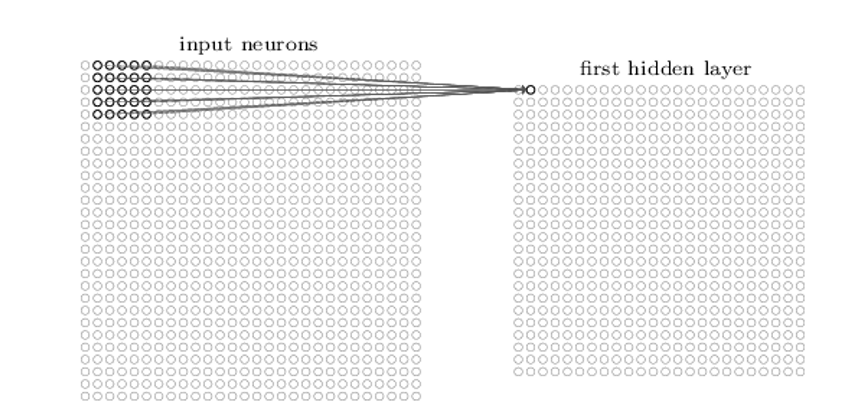
Deep Learning ist eine Teildisziplin von machine learning und basiert auf tiefen neuronalen Netzen. Diese sind dem menschlichen Gehirn nachempfunden und deren Aufbau kann in 3 Schichten, wobei jede davon aus einer Vielzahl von Neuronen besteht, unterteilt werden. Tiefe neuronale Netze zeichnen sich durch ein mehrschichtiges hidden layer aus.

In der ersten Schicht, dem input layer, wird die Dateneingabe, beispielsweiße durch ein Neuron für jeden Pixel eines Bildes vorgenommen. Die Werte der einzelnen Pixel werden ohne Kontext zueinander flach als Linie dargestellt (Siehe Abbildung 1-1). Anschließend werden die Daten an das hidden layer weitergegeben, in welchem die Daten weiterverarbeitet werden. Das hidden layer besteht bei tiefen Netzen aus mehreren Ebenen, welche aus den Eingabedaten immer weitere Informationen erkennen bis der output layer erreicht ist, welcher das Ergebnis der Auswertung liefert und eine Vorhersage trifft. Jedes Neuron im output layer liefert einen Wert zwischen 0 und 1, welcher die Wahrscheinlichkeit darstellt, dass dieses Neuron die richtige Antwort darstellt. Die Summe aller Ausgabewerte ist 1.

Bei dieser Art von neuronalen Netzen handelt es sich um ein Deep feedforward neural network, da die Neuronen immer nur nach vorne kommunizieren.

#### Konvolutionelle neuronale Netze

Ein Convolutional neural network funktioniert vom Aufbau her wie der visuelle Cortex des Menschen. Hierbei liegt an den Neuronen des ersten ersten hidden Layers nichtmehr jedes einzelne input Neuron an, sondern ein gewisser Bereich. Die Eingabewerte behalten hierbei auch ihre ursprüngliche Form bei und werden nichtmehr umgeformt dargestellt.



Der Vorteil hiervon ist, dass gewisse Strukturen und Zusammenhänge der Eingabewerte nichtmehr verloren gehen. Die Gewichte und der Bias der hidden Neuronen werden hier geteilt. Jedes hidden Neuron besitzt für jede Verbindung zu den Input Neuronen dieselben Gewichte und denselben Bias. Sollten die Neuronen 25 Inputs erfassen besitzen sie somit 25 Gewichte.

Alle Neuronen werden somit darauf eingestimmt bestimmte Muster (z.B eine Kante in einem Bild) zu erkennen. Um mehrere Unterschiedliche Muster zu erkennen können mehrere Schichten verwendet werden. Diese Schichten bezeichnet man als convolutional feature Layer.

Daraufhin folgen ein oder mehrere Pooling Layer, welche die Informationen aus dem feature Layer auf einen kleineren Bereich zusammenfassen, sie fassen beispielsweiße ein 2x2 großen Bereich in einem einzelnen Neuron zusammen.

Am Ende werden alle Neuronen des Pooling Layers mit allen Neuronen des Output Layers verbunden und es wird wie gewohnt eine Vorhersage getroffen.

## Genetische Algorithmen

# Konkreter Einsatz im Betrieb

Ein Anwendungsfall von maschinellem Lernen ist beispielsweise die Betrugsaufdeckung bei PayPal. Das Hauptziel hierbei ist es die Anzahl der Fehlalarme und damit das Blockieren der Accounts ordnungsgemäßer Kunden zu verhindern. Klassische Analysetools würden beispielsweise einen Accounts sperren nachdem er innerhalb kurzer Zeit von mehreren unterschiedlichen IP-Adressen aufgerufen wurde. Da maschinelle Verfahren mehr Inputvariablen verarbeiten können, bieten diese die Möglichkeit die einzelnen Fälle deutlich präziser zu untersuchen. In diesem Fall wäre es beispielsweise möglich zu erkennen, dass alle IP-Adressen aus einem Flughafen W-LAN stammen und es sich beim Kunden um einen Piloten handelt. <https://www.americanbanker.com/news/how-paypal-is-taking-a-chance-on-ai-to-fight-fraud>

Ein weiterer Anwendungsbeispiel liefert Amazon mit seinem gezielten Produktvorschlägen für seine Kunden. Durch das Verwenden von neuronalen Netzen kann der extrem umfassende Produktkatalog mit den bereits gekauften bzw. betrachteten Produkten des einzelnen Kunden abgeglichen und neue Vorschläge generiert werden. <https://aws.amazon.com/de/blogs/big-data/generating-recommendations-at-amazon-scale-with-apache-spark-and-amazon-dsstne/>

# Prototypische Konzepte und Implementierung

## Prototyp Supervised

### Tensorflow

Zur Implementierung des Prototyps eignet sich aufgrund von einer umfangreicher Dokumentation, sowie einfacher Handhabung die Bibliothek Tensorflow

Tensorflow ist eine von Google entwickle Bibliothek für maschinelles Lernen mit dem Schwerpunkt auf Sprach- und Bilderkennung durch (tiefe) neuronale Netze. Sie basiert auf Python und C++, bietet jedoch auch eine API für Java, Go und einige weitere Sprachen.

Zusätzlich wird mit Tensorboard eine einfache Visualisierung der erzeugten Zwischen- und Endergebnisse angeboten.

/ANN,

## Prototyp Unsupervised

//K-means , Autoencoder etc….