Fallstudie 27: Maschinelles Lernen

Mike Wieder

17.05.2018

# Corporate Performance Management

Inhalt

[Corporate Performance Management 1](#_Toc513987048)

[Was versteht man unter maschinellem Lernen? 3](#_Toc513987049)

[Kriterien zum sinnvollen Einsatz von maschinellem Lernen 4](#_Toc513987050)

[Messbarkeit der Performance 4](#_Toc513987051)

[Genügend Trainingsdaten 5](#_Toc513987052)

[Qualität der Trainingsdaten 5](#_Toc513987053)

[Klar definierte Problemstellung 5](#_Toc513987054)

[Geschichtliche Entwicklung 5](#_Toc513987055)

[Grundbegriffe 6](#_Toc513987056)

[Ground truth 6](#_Toc513987057)

[Backpropagation 6](#_Toc513987058)

[Überwachtes(supervised) learning: 6](#_Toc513987059)

[Unüberwachtes(unsupervised) learning: 6](#_Toc513987060)

[Anwendungsgebiete 7](#_Toc513987061)

[Künstliche neuronale Netze 7](#_Toc513987062)

[Grundbegriffe 7](#_Toc513987063)

[Deep Learning 9](#_Toc513987064)

[Einsatz maschineller Lernverfahren im Unternehmen 11](#_Toc513987065)

[Konkrete Anwendungsbeispiele 11](#_Toc513987066)

[Prototypische Konzepte und Implementierung 12](#_Toc513987067)

[Prototyp Supervised 12](#_Toc513987068)

[Tensorflow 12](#_Toc513987069)

[Prototyp Unsupervised 12](#_Toc513987070)

[Literaturverzeichnis 13](#_Toc513987071)

# 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 Lernvorgangs 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 mit zunehmender Erfahrung, also mehr zur Verfügung stehender Daten, verbessern.

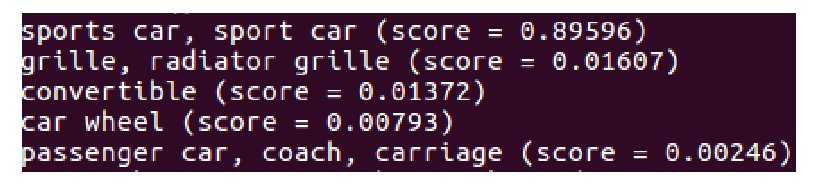


Abbildung ‑ - Beispielhafter Vorhersagen-output einer Anwendung zur Bilderkennung

Abbildung 1 zeigt beispielhaft den Output eines auf künstlichen neuronalen Netzen basierendem Algorithmus zur Bilderkennung. Hier wurde dem bereits eingelerntem Modell ein unbekanntes Bild eines Sportwagens vorgelegt, welches er mit einer „Sicherheit“ von 89,596% auch als solches klassifiziert hat.

Der große Vorteil von maschinellem Lernen ist, dass damit sehr große Datenmengen ausgewertet werden können und die Qualität der Ergebnisse mit zunehmender Größe des Datensatzes sogar zunimmt. Dabei können wesentlich mehr Variablem berücksichtigt werden als mit klassischen Ansätzen.

Die 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.

Besonders lohnenswert sind maschinelle Lernverfahren dann, wenn die auszuwertenden Daten sehr umfangreich sind und viele Dimensionen besitzen, sodass die Komplexität der Daten so hoch ist, dass ein menschlicher Anwender keine Verbindungen und Zusammenhänge mehr erkennen kann.

Grundsätzlich können maschinelle Lernverfahren wie in Abbildung 2 gezeigt kategorisiert werden. In dieser Ausarbeitung werden überwachte und unüberwachte Lernverfahren näher betrachtet, während das Bestärkendes Lernen nur kurz angeschnitten wird.  
Unter dem Bereich überwachtes Lernen werden hauptsächlich Klassifikations- und Regressionsalgorithmen zusammengefasst, während beim unüberwachten Lernen größtenteils Clustering-Algorithmen verwendet werden.



Abbildung - Überblick unterschiedlicher Verfahren

## Kriterien zum sinnvollen Einsatz von maschinellem Lernen

Auch wenn auf maschinellem Lernen basierende Lösungsansätze oftmals deutlich bessere Ergebnisse erzielen als vergleichbare klassische 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. Jedoch steigt gleichzeitig die Komplexität und somit der Rechenaufwand mit zunehmender Datenmenge an, sodass es nicht immer sinnvoll sein muss alle zur Verfügung stehenden Daten zu verwenden.

### 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

Der erste Deep Learning Algorithmus wurde bereits 1965 von Ivakhenko und Lapa vorgestellt. 1979 wurde von Fukushima das erste convolutional neural network vorgestellt, welches die Eingabewerte zuerst nach nützlichen Informationen filtert und so Muster erkennen kann.

In den 80er Jahren wurde weiter an dem Verbessern des Einlernens der Netze, vor allem an der Verbesserung des Gradiantenverfahrens, geforscht.

Die erste praktische Anwendung wurde 1989 von LeCun entwickelt und wurde dazu benutzt um handgeschriebene Ziffern zu erkennen. Sie basierte auf einem convolutional neural network. Das Trainieren des Netzes dauerte jedoch 3 Tage und die Gewichte der Neuronen mussten teilweise manuell gestellt werden, weshalb es nie praktisch eingesetzt wurde.

Durch die hohen Rechenkosten und der fehlende Erfolg der bisherigen Forschung wurde Deep Learning in den 1990er und 2000er nur wenig gefördert und verwendet.

Durch die Weiterentwicklung von GPUs und dadurch die Möglichkeit wesentlich größere Trainingsdatensätze zu verwenden ist Deep Learning seit 2009 eines der wichtigsten Forschungsthemen im Gebiet der künstlichen Intelligenz und des maschinellen Lernens. Fortschritte in der Forschung des menschlichen Gehirns helfen ebenfalls bei der Entwicklung neuronaler Netze. [1]

# Gängige Konzepte

### 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 3 - 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. Ziel des Lernvorgangs ist es also, eine Generalisierungs- bzw. Assoziationsfähigkeit zu erlangen. Damit können ähnliche Probleme, welche nicht explizit trainiert wurden besser gelöst werden. [2]

## 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

Eine der heute am weitesten verbreitete Umsetzung von maschinellem Lernen sind künstliche neuronale Netze mit dem Untergebiet Deep Learning.

## Künstliche neuronale Netze

Künstliche neuronale Netze sind vom Aufbau dem menschlichen Gehirn nachempfunden. Durch den Einsatz einer Vielzahl von Neuronen parallelisieren sie die Berechnungen und erlangen eine Lernfähigkeit. Dies bedeutet, dass sie nicht auf eine Aufgabe explizit programmiert werden müssen sondern durch Trainingsdaten darauf eingelernt werden. [2]

Es gibt viele unterschiedliche Paradigmen zur Erzeugung und Verwendung neuronaler Netze. Hier möchte ich einige klassische Beispiele näher beschreiben. [2]

Zunächst werden die Grundbegriffe und Bausteine von künstlichen neuronalen Netze erklärt

### 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.

Eine 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.

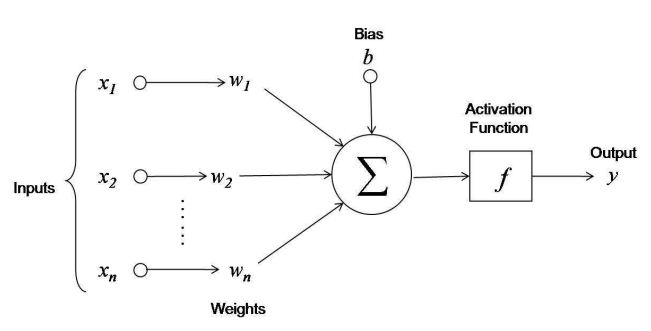


Abbildung - Funktionsweise eines Neurons [3]

#### 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.

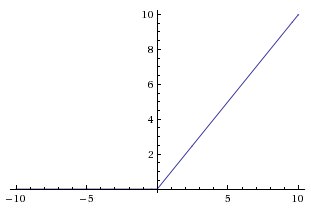


Abbildung - RELU-function [4]

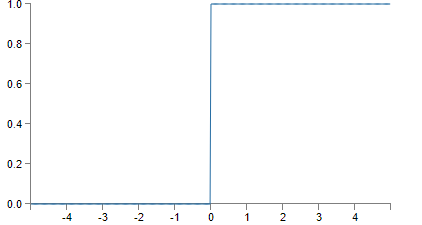


Abbildung 4 - Step-function, welche bei einem Perzeptron eingesetzt wird [5]

#### Gradientenverfahren

Das Gradientenverfahren ist ein wichtiger Bestandteil bei dem Lernen eines neuronalen Netzes. Hierfür müssen die vom Netz erzeugten Vorhersagen mit dem Label verglichen werden indem der Abstand (z.B. Cross-Entropy, Manhatten-Distanz, Euklidische-Distanz, …) dazwischen berechnet wird. Das Ziel des Gradientenverfahrens hierbei ist es diesen Abstand zu minimieren. Der Abstand ist von den Gewichten, dem Bias, sowie den Eingabewerten abängig, also müssen diese dementsprechend angepasst bzw. optimiert werden.  
Der Gradient zeigt „bergab“, bis ein (lokales) Minimum erreicht wird und somit in die Richtung des geringsten Abstandes zwischen der erzeugten und der richtigen Lösung zeigt. Ein globales Minimum wird mit diesem Verfahren häufig nicht erreicht.

Die Gewichte und der Bias werden daraufhin entsprechend dem erreichten Minimum angepasst. Dabei werden sie entsprechend ihrem Einfluss auf das Ergebnis stark verändert. Die Gewichte in den späteren Schichten des Netzes werden also stärker verändert als die Gewichte in den früheren Schichten. Dies wird auch als vanishing-gradient-problem bezeichnet.

Es wird jedoch immer nur ein gewisses Stück weit dem Gradienten gefolgt (Lernrate), da man ansonsten häufig die Werte über das Minimum hinaus anpasst und es somit nie erreicht. Über die Lernrate kann somit die Lerndauer des Netzes und die erreichte Präzision eingstellt werden.

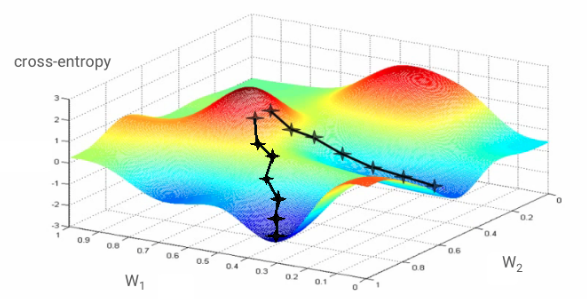


Abbildung - Gradiantenverfahren mit cross-entropy [6]

### Deep Learning

Deep Learning ist eine Teildisziplin von machine learning und basiert auf tiefen neuronalen Netzen. 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.

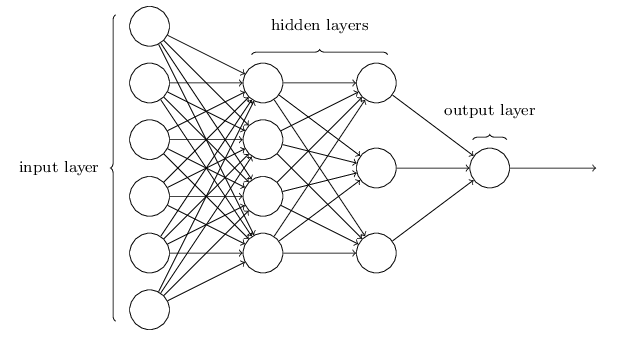


Abbildung - Schematischer Aufbau eines KNN

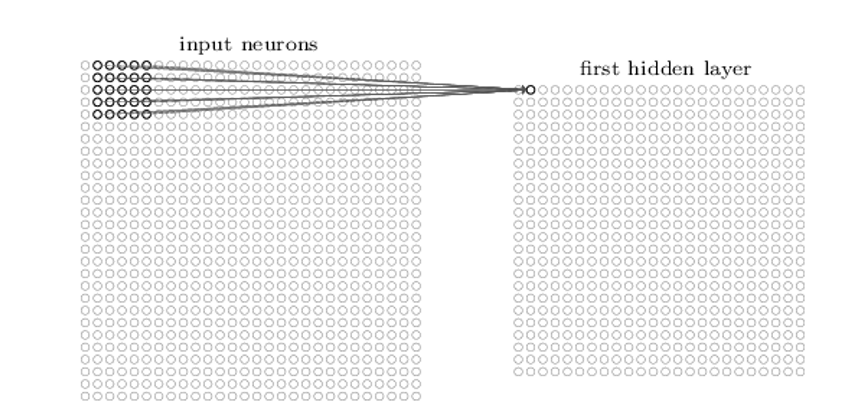
Bei dieser Art von neuronalen Netzen handelt es sich um ein Deep feedforward neural network, da die Neuronen immer nur nach vorne kommunizieren.

Jedes Neuron des hidden layers führt grundsätzlich dieselbe Aufgabe durch und summiert alle anliegenden gewichteten Werte der vorherigen Neuronen auf und addiert eine Konstante(Bias), was dann an eine nichtlineare Funktion, auch activation function genannt (z.B. Softmax, Sigmoid), weitergegen wird.

Das Resultat wird wieder an die darauffolgenden Neuronen weitergegeben und der Vorgang wiederholt sich bis das output Layer erreicht ist

#### 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.

# Einsatz maschineller Lernverfahren im Unternehmen

Durch die heutzutage relativ günstig zu beschaffende Rechenleistung (vor allem durch Cloud-Computing) und die immer größeren zur Verfügung stehenden Datenmengen öffnen sich viele wirtschaftlich rentable Einsatzzwecke für maschinelle Lernverfahren. Einer Befragung von 264 IT-Entscheidern aus unterschiedlichen Unternehmen in dem Jahr 2016 zufolge beschäftigten sich in Deutschland bereits 64% der Unternehmen aktiv mit dem Thema maschinelles Lernen, wobei ein Fünftel aller deutscher Unternehmen solche Technologien bereits produktiv einsetzt. [7]

Die Gründe für den Einsatz dieser Verfahren sind hauptsächlich die gewünschte Optimierung Unternehmensinterner Prozesse um beispielsweise die Fertigung zu verbessern und die Verbesserung der Customer-Experience durch gezieltere Kundenansprache mit speziellen Produkten oder Dienstleistungen. [7]

Die häufigste konkrete Zielsetzungen für den Einsatz ist die stärkere Bindung des Kunden gefolgt von einer Steigerung der Verkäufe von Digital Services. Die Fehlerreduktion in der Produktion sowie eine verstärkte Automatisierung ist ebenfalls von Bedeutung. [7]

## Konkrete Anwendungsbeispiele

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. [8]

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. [9]  
Dadurch können dem Kunden Produkte angezeigt werden, welche wahrscheinlicher von ihm benötigt und somit gekauft werden.

[7]

# 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….

# Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | T. Dettmers, „devblogs.nvidia.com,“ 16 Dezember 2015. [Online]. Available: https://devblogs.nvidia.com/deep-learning-nutshell-history-training/. |
| [2] | D. Kriesel, Ein kleiner Überblick über Neuronale Netze, 2007. |
| [3] | „blogs.cornell.edu,“ Cornell University, 8 September 2015. [Online]. Available: http://blogs.cornell.edu/info2040/2015/09/08/neural-networks-and-machine-learning/. [Zugriff am 12 Mai 2018]. |
| [4] | D. Becker, „kaggle.com,“ kaggle, [Online]. Available: https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning/code. [Zugriff am 12 Mai 2018]. |
| [5] | M. Nielsen, „http://neuralnetworksanddeeplearning.com,“ Dezember 2017. [Online]. Available: http://neuralnetworksanddeeplearning.com/chap1.html. [Zugriff am 12 Mai 2018]. |
| [6] | M. Görner, „codelabs.developers.google.com,“ 2016. [Online]. Available: https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#4. [Zugriff am 12 Mai 2018]. |
| [7] | B. Böttcher, D. Klemm und C. Velten, Januar 2017. [Online]. Available: https://www.unbelievable-machine.com/downloads/studie-machine-learning.pdf. |
| [8] | P. Crosman, 1 September 2016. [Online]. Available: https://www.americanbanker.com/news/how-paypal-is-taking-a-chance-on-ai-to-fight-fraud. [Zugriff am 12 Mai 2018]. |
| [9] | K. Chung, „aws.amazon.com,“ Amazon, 9 Juli 2016. [Online]. Available: https://aws.amazon.com/de/blogs/big-data/generating-recommendations-at-amazon-scale-with-apache-spark-and-amazon-dsstne/. [Zugriff am 12 Mai 2018]. |