

Bildähnlichkeitserkennung von Markenlogos mithilfe von Machine Learning

Untertitel

Masterarbeit

Eingereicht in teilweiser Erfüllung der Anforderungen zur Erlangung des akademischen Grades:

Master of Science in Engineering

an der FH Campus Wien

Studienfach: Software Design and Engineering

Autor:

David Walser

Matrikelnummer:

01609388

Betreuer:

FH-Prof. DI Dr. Igor Miladinovic

Datum:

02.02.2022

Erklärung der Urheberschaft:

Ich erkläre hiermit diese Masterarbeit eigenständig verfasst zu haben. Ich habe keine anderen Quellen, als die in der Arbeit gelisteten verwendet, noch habe ich jegliche unerlaubte Hilfe in Anspruch genommen

Ich versichere diese Masterarbeit in keinerlei Form jemandem Anderen oder einer anderen Institution zur Verfügung gestellt zu haben, weder in Österreich noch im Ausland.

Weiters versichere ich, dass jegliche Kopie (gedruckt oder digital) identisch ist.

Datum:

Unterschrift:

Abstract

(E.g. “This thesis investigates...”)

Kurzfassung

(Z.B. "Diese Arbeit untersucht...")

Abkürzungen

ÖPA	Österreichisches Patentamt
KNN	künstliches neuronales Netzwerk
CNN	convolutional neural network
ARP	Address Resolution Protocol
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
WLAN	Wireless Local Area Network

Schlüsselbegriffe

GSM

Mobilfunk

Zugriffsverfahren

Inhaltsverzeichnis

1	Einführung	1
1.1	Problembeschreibung	1
1.2	Motivation und Ziel	1
2	Machine learning	2
2.1	Supervised Machine Learning	3
2.2	Unsupervised Machine Learning	4
2.3	Semi-supervised Learning	5
2.4	Reinforcement Learning	6
2.5	Neuronale Netzwerke	7
3	Bildbezogenes Machine Learning	13
3.1	Wie computer sehen	13
3.2	Bilderkennung und Klassifizierung	14
3.3	Algorithmen für Bildähnlichkeitserkennung	14
4	Prototyp zur Bildähnlichkeitserkennung von Markenlogos	15
4.1	Daten	15
4.2	Bildvorverarbeitung	15
4.3	Algorithmus	15
5	Diskussion der Ergebnisse	16
6	Conclusio	17
7	Ausblick	18
	Bibliographie	19
	Abbildungen	23
	Tabellen	24
	Appendix	25

1 Einführung

1.1 Problembeschreibung

Für uns Menschen ist es eine ziemlich einfache Aufgabe zu ermitteln ob ein Bild ähnlich zu einem anderen ist oder nicht. Wir erkennen alle Möglichen Merkmale eines Bildes, wie Farben, Texte oder Muster, ohne große Schwierigkeiten. Es stellt jedoch eine Herausforderung dar, wenn ein Bild mit 400.000 anderen Bildern verglichen und auf Ähnlichkeit geprüft werden soll.

Im Jahre 2020 wurden 6260 neue Marken beim österreichischen Patentamt angemeldet [1]. Die meisten dieser Marken werden in Kombination mit einem Bild, auch Logo genannt, registriert. Damit es bei einer Neuanschuldung nicht zu einer unmittelbaren Verwechslungsgefahr mit bereits bestehenden Marken kommt, bietet das österreichische Patentamt einen Dienst an, bei dem Daten zu einer Marke angegeben werden, und überprüft wird, ob es Ähnlichkeiten mit bereits angemeldeten Marken gibt [2]. Ein Hauptbestandteil dieser Ähnlichkeitsrecherche ist die Überprüfung von Ähnlichkeiten der Logos.

1.2 Motivation und Ziel

Machine Learning ist ein aufkommendes und zukunftsweisendes Thema. Laut einer Studie aus 2017 wurde ein Wachstum des machine learning Marktes von 1.03 Milliarden \$ in 2016 auf 8.81 Milliarden \$ im Jahre 2022 erwartet, was einer Wachstumsrate von 44.1% entspricht [3]. Technologisch gesehen ist der mit dieser Masterarbeit verbundene Prototyp eine große Herausforderung, da zur Umsetzung allerneuste Technologien und Frameworks benötigt werden. Eine weitere Herausforderung wird es sein, wie genau die vom österreichischen Patentamt dankenswerter weise zur Verfügung gestellten Daten zu analysieren und kategorisieren sind. Ziel dieser Masterarbeit ist es das Patentamt bei der Ähnlichkeitsrecherche zu unterstützen, in dem ein Prototyp entwickelt wird, welcher Ähnlichkeiten bei Bildern erkennt. Außerdem soll diese Masterarbeit einen Überblick über Machine Learning, mit Fokus auf Bildverarbeitung, enthalten. Daraus ergibt sich die folgende Forschungsfrage

Welche Unterschiede weisen verschiedene ML Algorithmen auf, im Bezug auf Erkennungsrate einer Ähnlichkeitsüberprüfung von Bildern?

zu beantworten.

2 Machine learning

Für viele Firmen ist machine Learning bereits der meist benutzte Bereich aus dem Feld der künstlichen Intelligenz [4]. Machine Learning ist die Wissenschaft und Kunst Computern das Lernen anhand von Daten zu ermöglichen [5] und ist ein Anwendungsgebiet von künstlicher Intelligenz welches bereits seit vielen Jahren die Forschung und Wirtschaft unterstützt [6]. "Machinelles Lernen ist ein Oberbegriff für die "künstliche" Generierung von Wissen aus Erfahrung: Ein künstliches System lernt aus Beispielen und kann diese nach Beendigung der Lernphase verallgemeinern." [7] Muster und Gesetzmäßigkeiten werden aus den Trainingsdaten erkannt, woraus ein statistisches Modell erzeugt wird [7]. Dieser Prozess wird als Modelltraining bezeichnet und ist ein iterativer Prozess, welcher oft mehrfach durchlaufen wird, bis die Qualität des Ergebnis zufriedenstellend ist [6]. Aus dem Erlernen und Analysieren der historischen Daten kann ein Ergebnis für neue und unbekannte Daten prognostiziert werden, ohne explizit darauf programmiert zu sein [8].

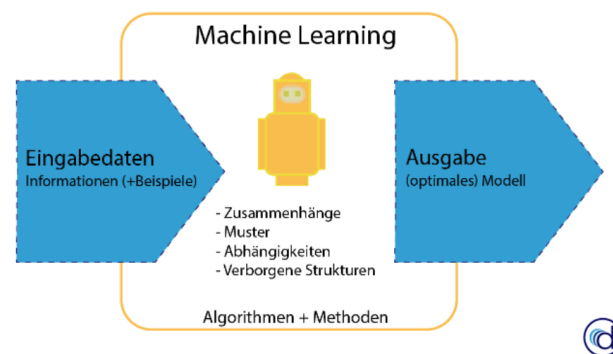


Abbildung 2.1: Machine Learning nimmt Eingabedaten mit Beispielen und lernt daraus, um für die Zukunft Prognosen zu machen [6]

Ein Alltag ohne dem Interagieren mit Machine Learning ist heutzutage kaum mehr wegzudenken. Bei der Benutzung von sozialen Medien, online Shopping oder Bankdiensten kommt Machine Learning zum Einsatz [9] und bereits seit den 1990er Jahren beeinflusst Machine Learning in Form des Spam Filters das Leben vieler [5]. Netflix bietet mithilfe von Machine Learning personalisierte Film und Serienempfehlungen an, und zusätzlich unterstützt Machine Learning bei der Optimierung der Produktion von Filmen und TV Shows [10]. Facebooks Algorithmus kann bereits mit 100 bis 150 Likes die Persönlichkeit einer Person genauer beschreiben als deren Familienmitglieder oder Freunde [11]. Die Grundlage für Machine Learning bilden Algorithmen, welche sich in folgende Arten aufteilen lassen [9]:

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning

2.1 Supervised Machine Learning

Supervised Machine Learning, im Deutschen übersetzt überwachtes Lernen, ist die am häufigsten angewendete Algorithmusart [9]. Ähnlich wie in der Schule wenn unter Aufsicht des Lehrers oder der Lehrerin geprüft wird, ob ein Problem richtig oder falsch gelöst wird, ist die Situation bei supervised Machine Learning Algorithmen. Dem Algorithmus wird ein gelabelter Datensatz zum Lernen zur Verfügung gestellt, somit weiß der Algorithmus für jeden Datensatz die richtige Lösung [12]. Ein gelabelter Datensatz kann z.B. ein Bild von einem Tier sein, wobei hier zusätzlich auch die Information über ein Feature mitgegeben wird, wie z.B. die Art des Tieres oder das Gewicht des Tieres (siehe Abbildung 2.2). Das Label ist die Information über ein Feature welche der Algorithmus später vorhersagen will. [13] Folgende

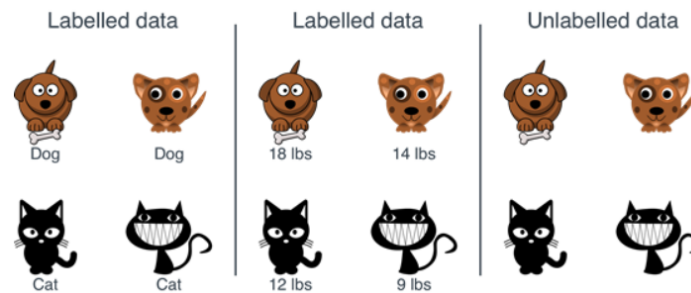


Abbildung 2.2: Labeled vs unlabeled Data [13]

Algorithmen sind Beispiele für supervised Learning [5]:

- k-Nearest Neighbors
- Naive Bayes
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees und Random Forests
- Neuronale Netzwerke (wobei diese auch unsupervised oder semisupervised sein können)

Klassifizierung und Regression sind klassische Anwendungsgebiete für supervised learning [5].

2.1.1 Klassifizierung

Bei Klassifizierungsproblemen geht es darum einen Status vorherzusagen [13], wie z.B. zu welcher Klasse oder Gruppe die Inputdaten gehören [12]. Der Spam Filter ist ein gutes Beispiel für Klassifizierung, hierbei handelt es sich um zwei Klassen: Spam und nicht Spam. Der Algorithmus bekommt E-Mails zum Lernen, welche als Spam E-Mail oder normale E-Mail gelabelt sind, um somit für neue E-Mails herauszufinden, ob diese Spam E-Mails sind [5].

2.1.2 Regression

Regression ist ein Fachgebiet der Statistik und ist eine Methode um die Beziehung zwischen unabhängigen Variablen oder Merkmalen und einer abhängigen Variable oder einem Ergebnis zu verstehen. Sobald die Beziehung zwischen unabhängigen und abhängigen Variablen geschätzt wurde, können die Ergebnisse vorhergesagt werden. [14] Bei Regressionsproblemen

geht es um kontinuierliche Daten [12] und darum eine Zahl vorherzusagen, wie z.B. das Vorhersagen von Aktienkursen [13] oder von Grundstückspreisen [12].

2.2 Unsupervised Machine Learning

Bei unsupervised Machine Learning, im Deutschen übersetzt unüberwachtes Lernen, handelt es sich um den Ansatz mit Datensätzen zu lernen, welche kein Label besitzen [5]. Dies wird auch als selbst organisiertes Lernen bezeichnet [12], da der Algorithmus ohne einen Lehrer oder eine Lehrerin lernt [5]. Da es sich hierbei um ungelabelte Datensätze handelt, besitzen diese nicht die Zielinformation, welche vorhergesagt werden soll [13] und somit werden Zusammenhänge in den Daten [15], versteckte Muster oder Datengruppierungen von unsupervised Algorithmen erkannt, ohne dass menschliches Eingreifen erforderlich ist [14]. Die Fähigkeit,

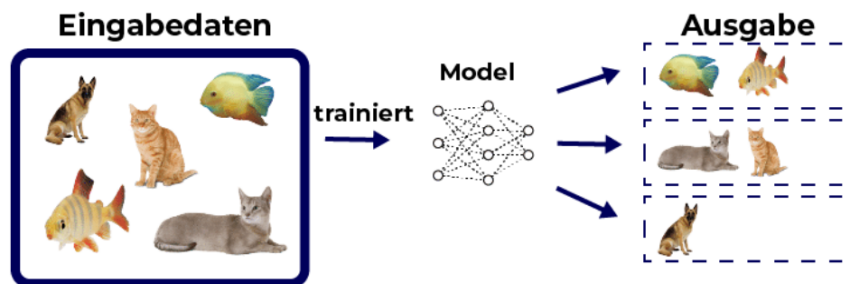


Abbildung 2.3: "Model trainiert ohne Zielvariable und findet eigenständig Muster und Zusammenhänge in den Daten"[15]

Ähnlichkeiten und Unterschiede in Informationen zu entdecken, macht sie zur idealen Lösung für die explorative Datenanalyse, Kundensegmentierung und Bilderkennung. Folgende Algorithmen sind Beispiele für unsupervised Learning [5]:

- K-Means
- DBSCAN
- Hierarchical Cluster Analysis (HCA)
- Principal Component Analysis (PCA)
- Kernel PCA
- Locally-Linear Embedding (LLE)
- Apriori
- Eclat

Clustering, Assoziationsanalyse und Dimensionalitätsreduktion sind die drei Hauptaufgaben von unsupervised Machine Learning [14].

2.2.1 Clustering

Beim Clustering geht es darum, die Population oder die Datenpunkte in eine Reihe von Gruppen aufzuteilen, so dass die Datenpunkte in denselben Gruppen anderen Datenpunkten in derselben Gruppe ähnlicher und den Datenpunkten in anderen Gruppen unähnlicher sind [16]. Es werden ungelabelte Daten auf der Grundlage ihrer Ähnlichkeiten oder Differenzen in verschiedene Gruppen aufgeteilt [14]. Anwendungsgebiete für Clustering sind beispielsweise

Bücher die je nach Titel und Inhalt in verschiedene Gruppen eingeteilt werden oder Pflanzen und Tiere welche in Spezies aufgeteilt werden (siehe Abbildung 2.3) [16].

2.2.2 Assoziationsanalyse

Die Assoziationsanalyse ist eine Methode um herauszufinden welche Muster (Beziehungen, Korrelationen, Strukturen, ...) es in den Datensätzen gibt [17]. Diese Methode wird häufig für Warenkorbanalysen verwendet, um ein besseres Verständnis über die Beziehung zwischen den Produkten zu bekommen. Beispiele davon sind z.B. auf Amazon die "Kunden, die diesen Artikel gekauft haben, kauften auch:Änzeige oder die Spotify "Discover Weekly"Playlist. [14]

2.2.3 Dimensionalitätsreduktion

Während mehr Daten grundsätzlich zu genaueren Ergebnissen führen können, so kann dies auch die Leistung von Algorithmen für machine Learning beeinträchtigen (z.B. overfitting) und die Visualisierung von Datensätzen erschweren [14]. Bei der Dimensionalitätsreduktion geht es darum, die Variablen in den Daten auf die wesentlichen und zielführenden zu beschränken (z.B. werden redundante oder noise Features entfernt) [15], wobei die Integrität der Daten so weit wie möglich erhalten bleiben soll [14]. Es kann für das bereinigen von Daten oder auch für das Hervorheben von Features verwendet werden, da die Daten dabei von hochdimensionalem Feature Raum in niedrigdimensionalem Feature Raum transformiert werden [18].

2.3 Semi-supervised Learning

Um die Schwierigkeiten vom Erstellen von großen gelabelten Datensätzen entgegenzuwirken gibt es eine Methode bei der ein kleiner Anteil gelabelte Daten, der Großteil jedoch ungelabelte Daten sind. Diese Methode wird als semi-supervised Learning bezeichnet, welche die Benefits von unsupervised und supervised kombiniert. [19] Der Algorithmus wird initial mit den wenig gelabelten Daten trainiert und kann somit iterativ mehr und mehr an die ungelabelten Daten angewandt werden. Self-training und Co-training sind zwei Ansätze für semi-supervised learning. [20]

2.3.1 Self-training

Self-training ist eines der einfachsten Beispiele für semi-supervised learning und ist ein Prozedere einen supervised learning Ansatz in semi-supervised umzuwandeln [20]. Der straight-forward Ansatz wird anhand folgendem Beispiel erklärt:

1. Die gelabelten Daten werden für das Trainieren des Models herangenommen [21].
2. Dieses Model wird dann für die Vorhersage von ungelabelten Daten verwendet [21].
3. Es werden Ergebnisse, welche dem zuvor bestimmten Kriterium entsprechen (z.B. >90% accuracy), mit pseudo-labels verzehrt und mit den bereits gelabelten Daten kombiniert [21].
4. Das Model wird nun mit dem neuen Pool an gelabelten Daten trainiert [21].
5. Der ganze Prozess wird nun durchiteriert bis entweder alle Daten gelabelt sind oder die spezifizierte Maximalanzahl an Iterationen erreicht wird [21].

Die Performance von dem Self-training Ansatz variiert von Datensatz zu Datensatz und es gilt abzuwägen ob sie im Vergleich zu dem supervised Ansatz eine Verbesserung erzielt [20].

2.3.2 Co-training

Co-training ist eine verbesserte Version von Self-Training, die zum Einsatz kommt, wenn nur wenig gelabelte Daten verfügbar sind [20]. Bei dieser Methode benötigt es zwei Sichten auf die Daten [22] und basierend darauf werden zwei individuelle Klassifizierer trainiert [20]. Die zwei Ansichten sind verschiedene Gruppen von Features, welche jeweils Informationen für jede Instanz beinhalten [20]. Dies bewirkt dass beide Ansichten unabhängig sind und sie alleinstehend die Klasse einer Instanz vorhersagen können[22]. Ein Anwendungsgebiet für Co-Training könnte die Klassifizierung von Webinhalten sein. Der Inhalt einer Webseite kann in zwei Ansichten aufgeteilt werden: eine mit Wörtern, und die andere mit Verweistexten der Links, welche zu dieser Seite führen. [20] Der Ablauf eines Co-Trainings sieht wie folgt aus:

1. Für jede Ansicht werden die separaten Models mit den gelabelten Daten trainiert [20].
2. Die ungelabelten Daten werden hinzugefügt und mit pseudo-labels versehen [20].
3. Die beiden Klassifizierer trainieren sich gegenseitig mit den pseudo-labels und es wird das höchste Konfidenzniveau angestrebt. Wenn z.B. der erste Klassifizierer das Label für einen Datensatz vorhersagt, während der andere einen Fehler macht, so werden die vom ersten Klassifizierer zugewiesenen pseudo-labels jene vom zweiten überschreiben und vice-versa. [20]
4. Zuletzt werden die beiden geupdaten Klassifizierer in zu einem kombiniert [20].

2.4 Reinforcement Learning

Reinforcement Learning, im Deutschen übersetzt Bestärkendes oder Verstärkendes Lernen, ist eine immer beliebter werdende Methode und befasst sich damit intelligente Lösungen für komplexe Steuerungsprobleme zu finden [23]. Das lernende System, welches in diesem Kontext *Agent* genannt wird, beobachtet das Umfeld und führt dann Aktionen aus, für die es entweder Belohnungen oder Bestrafungen bekommt [5].

Hierbei erlernt der Agent eigenständig welche Aktion für welche Situation am besten geeignet ist, um die Belohnungsfunktion zu maximieren [24]. Im Gegensatz zu supervised oder unsupervised learning werden beim Reinforcement Learning vorab keine Daten benötigt. Die Daten werden während den Trial-and-Error Trainingsdurchläufen der Simulationsumgebung generiert und gelabelt. Das Ziel von Reinforcement Learning ist es eine bestmögliche Policy zu erreichen. [23] Als Policy wird die Strategie bezeichnet, welche der Agent ausführt und gibt an welche Aktion als nächstes ausgeführt werden soll[24] um die Belohnung zu maximieren [23]. Reinforcement Learning anhand eines Beispiels: Die Katze (siehe Abbildung , welche in diesem Beispiel der Agent ist, befindet

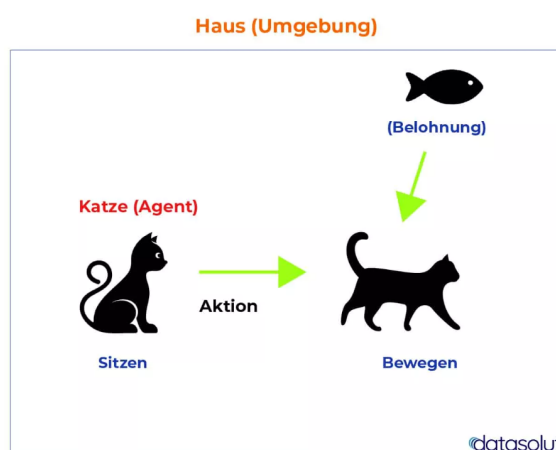


Abbildung 2.4: Beispiel von Reinforcement Learning [24]

sich im Haus (Umgebung) und kann die zwei Zustände sitzen oder bewegen haben. Wenn die Katze nun eine Aktion ausführt, so kann sie ihren Zustand ändern, wofür sie eine Belohnung oder eine Strafe erhalten kann.

2.5 Neuronale Netzwerke

Die Anfänge von künstlichen neuronalen Netzwerken machten 1943 der Neuropsychologe Warren MCCulloch und der Mathematiker Walter Pitts. Sie stellten ein vereinfachtes Computermodell vor, welches wie biologische Neuronen in Tiergehirnen zusammenarbeitet [5], um komplexe Aufgaben aus den Bereichen Statistik, Wirtschaft und Informatik lösen zu können [25]. Dies gilt als erste Architektur für künstliche neuronale Netzwerke und seither wurden viele weitere Architekturen entwickelt. [5] "Künstliche neuronale Netze, auch künstliche neuronale Netzwerke, kurz: KNN (englisch artificial neural network, ANN), sind Netze aus künstlichen Neuronen." [26] Der Ursprung von künstlichen Neuronen sind die biologische Neuronen, welche in tierischen Gehirnen zu finden sind. Neuronen kommunizieren in dem sie kurze elektrische Impulse über Synapsen versenden, bzw. empfangen. Erhält ein Neuron eine ausreichende Anzahl an Signalen innerhalb weniger Millisekunden, so feuert dies eigene Signale weiter. Die einzelnen Neuronen sind in einem Netzwerk mit Milliarden von anderen Neuronen verbunden, in dem sie in fortlaufenden Layern eingegliedert sind (siehe Abbildung 2.5). [5]

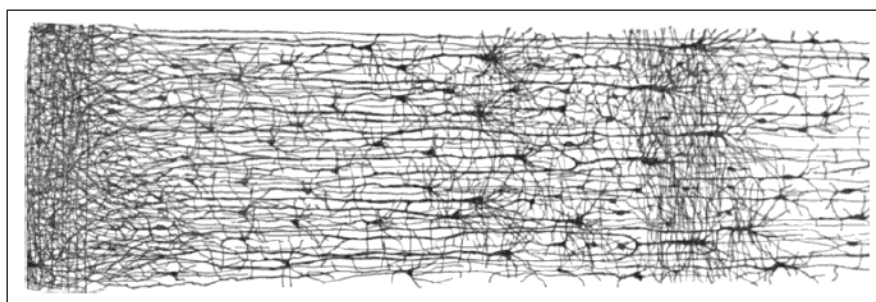


Abbildung 2.5: Multiple Layer in einem biologischen neuronalen Netzwerk [5]

Künstliche neuronale Netzwerke spiegeln das Verhalten des menschlichen Gehirns wieder und ermöglichen es Computerprogrammen selbständig Muster zu erkennen und allgemeine Probleme zu lösen [27]. Künstliche neuronale Netzwerke können verschiedene Datenquellen wie Geräusche, Bilder, Texte, Tabellen oder Zeitreihen interpretieren, woraus sie Informationen extrahieren und Muster erkennen um später auf unbekannte Daten Vorhersagen treffen zu können. Grundsätzlich weisen künstliche neuronale Netzwerke die Strukturen gerichteter Graphen auf, können aber je nach Komplexität unterschiedlich aufgebaut sein. [25] Angelehnt an die Struktur eines biologischen neuronalen Netzwerkes besteht ein künstliches neuronales Netzwerk ebenfalls aus Neuronen, welche in Layern (Schichten) angeordnet sind. Ein solches Netzwerk besteht aus einem Input-Layer, einem (oder mehreren) Hidden-Layer und einem Output-Layer (veranschaulicht in Abbildung ??), wobei jeder dieser Layer eine Vielzahl an Neuronen beinhaltet. Desto komplexer das zu lösende Problem ist, umso mehr Layer werden benötigt. [28] Der Input-Layer nimmt die eingegebenen Daten entgegen, verarbeitet und gewichtet diese, bevor sie an den Hidden-Layer weiter gegeben wird [25]. Der Hidden-Layer befindet sich zwischen Input und Output-Layer [25] und kann aus einem oder mehreren Layern bestehen und ist der Layer, in dem die meisten Berechnungen stattfinden [29]. Auch in

diesem Layer finden wieder Gewichtungen statt. Sind mehrere Layer im Hidden-Layer vorhanden, so geschieht das Gewichten pro Layer. [25] Die Ausgabewerte des letzten Hidden-Layers werden dem Output-Layer als Eingabewerte übergeben [29]. Der Output-Layer ist die letzte Schicht und beinhaltet die Entscheidung oder Vorhersagung des neuronalen Netzwerkes [25]. Eine angemessene Anzahl an Neuronen und eine passende Aktivierungsfunktion sollte je nach

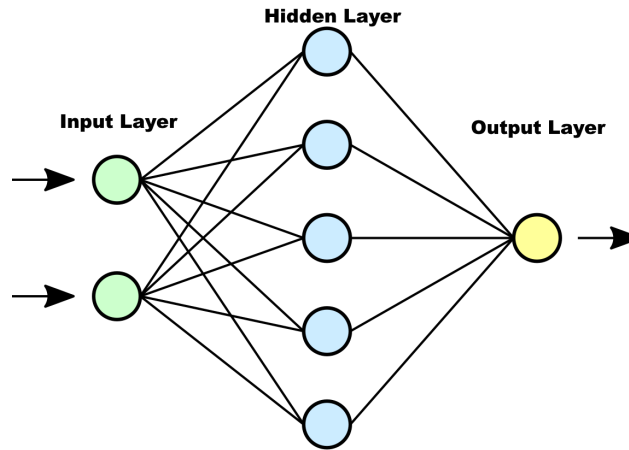


Abbildung 2.6: Einfache Veranschaulichung eines neuronalen Netzwerkes [26]

Aufgabe gewählt werden. In den Neuronen werden Berechnungen getätigt [30] und jedes Neuron hat ein zugeordnetes Gewicht, wodurch sie unterschiedliche Wichtigkeiten erhalten [28]. Im biologischen Kontext können Gewichte mit Synapsen verglichen werden [31]. Kombiniert mit einer Übertragungsfunktion entscheidet das Gewicht über den Input eines Neurons [28]. Bei den künstlichen neuronalen Netzwerken kombiniert ein Neuron die Eingangssignale mit zugeordneten Gewichten, welche die Bedeutung des Signals schwächen oder verstärken. Nach dem Aufsummieren wird das Ergebnis durch eine Aktivierungsfunktion gegeben um zu berechnen ob, und mit welchem Ausmaß das Ausgangssignal weitergesendet wird. Das Neuron wird als aktiviert bezeichnet, wenn es ein Ausgangssignal versendet. [30]

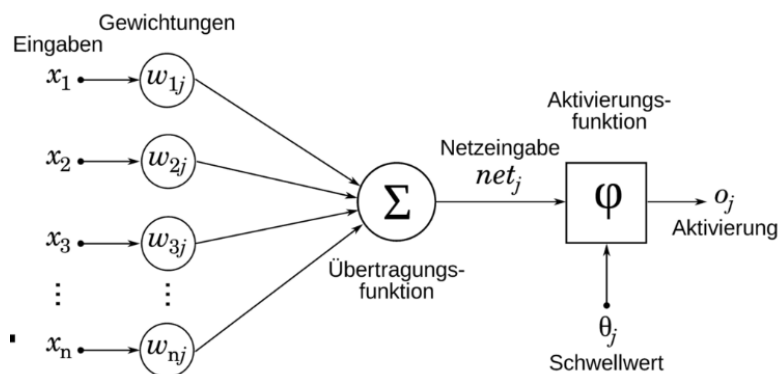


Abbildung 2.7: Aufbau eines Neurons [25]

2.5.1 Arten von neuronalen Netzwerken

Es gibt viele verschiedene Arten von künstlichen neuronalen Netzwerken und es werden nun ein paar davon etwas näher erläutert.

Preceptron

Preceptron ist ein neuronales Netzwerk mit nur einem Layer [32], welches im Bereich von supervised Learning Anwendung findet [33]. Es ist ein linearer oder binärer Klassifizierer und wird verwendet um Daten in zwei Gruppen zu klassifizieren [32]. Ein Preceptron funktioniert folgendermaßen:

1. Alle Eingangssignale werden mit ihren Gewichten multipliziert [32]
2. Nach dem multiplizieren wird alles aufsummiert [32].
3. Abschließend wird eine Aktivierungsfunktion angewendet, welche den Wert auf den gewünschten Wert, wie z.B. (0,1) mapt [32].

Feed forward neuronale Netzwerke

Ein Preceptron ist eine Form von Feed forward neuronalen Netzwerken, welche die einfachste Form von neuronalen Netzwerken sind [34]. Informationen werden nur in eine Richtung, von Input-Layer über Hidden-Layer zum Ausgangs-Layer verarbeitet. Abbildung 2.6 veranschaulicht ein feed forward neuronales Netzwerk und die funktionsweise ist gleich wie bei Preceptrons. Wenn es viele Layer zwischen dem Input und Output-Layer gibt, so wird es auch als deep feed forward neural network bezeichnet. [35]

Convolutional neuronale Netzwerke

Convolutional neural networks sind eines der beliebtesten deep neural networks. Der Name kommt von der mathematischen Linearfunktion convolution, welche auf Matrixen angewandt wird. [36] CNNs sind effizient mit 2D und 3D Eingabedaten und werden unter anderem für Objektdetektion bei Bildern verwendet [25]. Die Architektur ist unterschiedlich zu klassischen neuronalen Netzwerken, mehr dazu in Kapitel 3.3.2.

Recurrent neuronale Netzwerke

Wenn es um sequentielle oder zeitliche Daten geht so können feed forward neuronale Netzwerke nicht verwendet werden. Es wird ein Mechanismus benötigt um historische Daten zu speichern und deshalb wurden Recurrent neuronale Netzwerke entwickelt. In klassischen feed forward neuronalen Netzwerken sind alle Datenpunkte unabhängig von einander, was mit sequentiellen Daten, welche von vorherigen Daten abhängig sind, nicht funktioniert. [37] Recurrent neuronale Netzwerke haben ein Konzept mit dem sie Informationen über vorherige Daten speichern können. Dieses Konzept wird Memory genannt. [38]

2.5.2 Training bei Neuronalen Netzwerken

Ein Vorteil von künstlichen neuronalen Netzwerken ist, dass beim Erstellen nicht alle Parameter spezifiziert werden müssen, da der Algorithmus anhand von Beispielen von selbst lernt [39]. Neuronale Netzwerke können grundsätzlich für jedes komplexe Problem verwendet werden, bei dem es sich um funktionale Beziehungen handelt. Es ist hierbei nicht notwendig die Art der Beziehung zu kennen, da das neuronale Netzwerk durch das Beobachten und iterativem anpassen der Parameter lernt [40].

Um zu verstehen wie gut ein solches Netzwerk für eine bestimmte Aufgabe performt kann die Verlustfunktion verwendet werden. Zu Beginn wird ein neuronales Netzwerk mit zufälligen

Gewichten versehen, was in einem schlecht performenden neuronalem Netzwerk resultiert. Das Ziel ist es die hohe Verlustfunktion durch das Trainieren des neuronalen Netzwerkes in eine niedrige zu verwandeln. [41] Eine Methode für das fine-tuning von neuronalen Netzwerken ist Backpropagation. Dabei handelt es sich um eine Methode zur Anpassung der Gewichte eines neuronalen Netzwerkes, welche als Grundlage die Fehlerquote der vorangegangenen Iterationen (bei neuronalen Netzwerken Epochen genannt) hernimmt. Durch das richtige adaptieren der Gewichte kann die Fehlerquote verringert und die Genauigkeit des Models erhöht werden. [42] Gewichte verstärken oder schwächen die Signale der Neuronen und geben die Wichtigkeit eines bestimmten Bereiches der Daten an. Ein Neuron kann ein bestimmtes Muster erkennen und die Bedeutung dieses Muster für das Gesamtkonstrukt entscheidet sich je nach Gewicht [43].

2.5.3 Anwendungsbeispiel eines neuronalen Netzwerkes

Um ein besseres Verständnis für neuronale Netzwerke zu bekommen folgt nun ein Anwendungsbeispiel mit kleinen Codesnipets. Ziel ist es ein neuronales Netzwerk aufzubauen, welches ein Bild einer handgeschriebenen Zahl richtig klassifizieren kann. Hierfür wird der MNIST Datensatz ¹ verwendet, welcher 60000 28 mal 28 Pixel große Graustufenbilder von handgeschriebenen Zahlen von 0 bis 9 beinhaltet. Um den MNIST Datensatz zu laden wird Tensorflow und Keras verwendet (siehe Codesnippet 2.1).

```
from tensorflow.keras.datasets import mnist
# load dataset
(trainX, trainy), (testX, testy) = mnist.load_data()
```

Listing 2.1: MNIST Datensatz in Python laden

Die Bilder aus dem MNIST Datensatz sind in Abbildung 2.8 zu sehen.

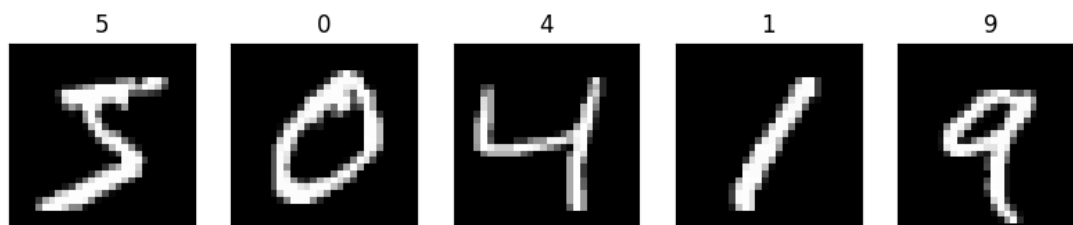


Abbildung 2.8: Beispielsbilder aus dem MNIST Datensatz

Die Architektur des neuronalen Netzwerkes hat 784 (28*28) Neuronen im Input-Layer. Somit kann jeder Pixel eines Bildes an ein Neuron als Eingabedaten übergeben werden. Der Hidden-Layer beinhaltet 15 Neuronen, wobei die Anzahl hier variieren kann. Abgeleitet von den vorhandenen Klassen (0 bis 9) ergeben sich 10 Neuronen im Output-Layer. Um diese Architektur mithilfe von Keras und Tensorflow in Python nachzubilden können die Codezeilen aus dem Codesnippet 2.2 verwendet werden.

```
from keras.layers import Dense
from keras.models import Sequential
image_size = 784 # 28*28 Pixel
```

¹MNIST Datensatz erhältlich unter <http://yann.lecun.com/exdb/mnist/>

```
num_classes = 10 # Zahlen von 0 bis 9
model = Sequential()
model.add(Dense(units=15, activation='sigmoid', input_shape=(
    image_size,)))
model.add(Dense(units=num_classes, activation='softmax'))
model.summary()
```

Listing 2.2: Architektur des Neuronalen Netzwerkes

Dense bedeutet dass es sich um ein vollständig verbundenes Netzwerk handelt, was bedeutet dass jedes Neuron des Input-Layers mit jedem Neuron aus dem nächsten Layer verbunden ist. Dies ist in Abbildung 2.9 zu sehen, welche die Beschriebene Architektur des neuronalen Netzwerkes veranschaulicht. Vor dem Compilieren und Trainieren des neuronalen Netzwerkes

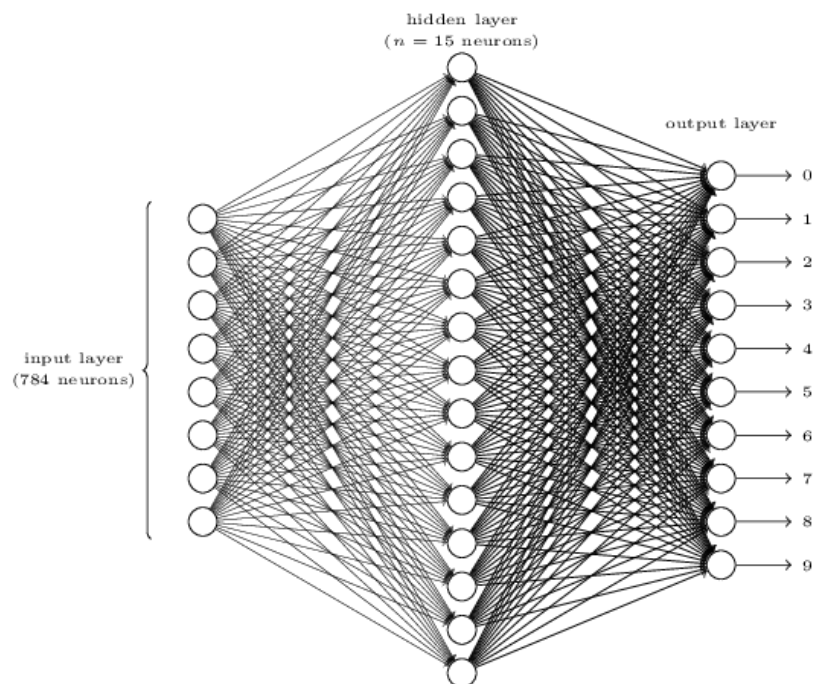


Abbildung 2.9: Architektur des Neuronalen Netzwerkes [44]

werden die Testdaten noch modifiziert, was im kompletten Codebeispiel im Anhang zu finden ist.

```
model.compile(loss='categorical_crossentropy', optimizer='sgd',
    metrics=['acc'])
history = model.fit(x_train, y_train, batch_size=128, epochs=5,
    verbose=False, validation_split=.1)
loss, accuracy = model.evaluate(x_test, y_test, verbose=False)
```

Listing 2.3: Model compilen und erlernen

Als Verlustfunktion wird hier die categorical crossentropy verwendet, da diese für Mehrklassenklassifizierung, bei der ein Datensatz genau einer Klasse angehört, sehr geeignet ist [45]. Stochastic Gradient Descent gehören zu den einfachsten Optimierer, weshalb sie in diesem Beispiel für das Model gewählt wurden. Für das Trainieren des Algorithmus wurden zum Vergleich einmal 5 und einmal 50 Epochen ausgewählt. Wie in Abbildung 2.10 zu sehen ist

2 Machine learning

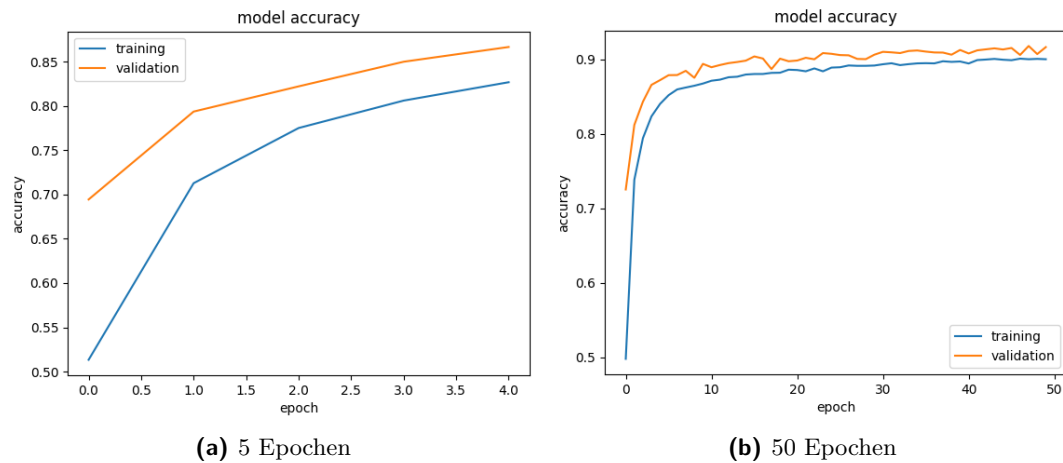


Abbildung 2.10: Vergleich 5 und 50 Epochen

erreicht das Model nach 5 Epochen eine Genauigkeit von etwas mehr als 80%, während es nach 50 Epochen eine Genauigkeit von fast 90% erreicht. Die Architektur und der Aufbau des neuronalen Netzwerkes könnte noch deutlich verändert und optimiert werden, jedoch dient dieses Beispiel zum Verständnis.

3 Bildbezogenes Machine Learning

Die Digitalisierung und Automatisierung von Abläufen nimmt auf der ganzen Welt rasant zu. Bilverarbeitung im Kontext von Computern und machine Learning bezieht sich darauf wichtige und bedeutende Informationen aus Bildern zu extrahieren. Da Bilder mittels unstrukturierten Daten repräsentiert werden, ist es weitaus schwieriger diese im Gegensatz zu strukturierten Daten wie Tabellen zu interpretieren. Je nach Anwendung gibt es unterschiedliche Methoden und Algorithmen, welche verwendet werden können. Die große Branche, welche zudem stetig wächst, wird bereits in verschiedenen Bereichen wie Video Verarbeitung, Gesichtserkennung, Medizinisches Umfeld, Robotic Vision, Selbstfahrende Autos und vieles mehr.[46] Während Tesla bereits autonome und selbständig fahrende Autos verkauft, hat Apple Ende 2021 einen Techniker von Tesla angeheuert und plant 2024 selbstfahrende Autos auf den Markt zu bringen [47]. Bildverarbeitung und Bilderkennung spielen hierbei eine große Rolle, da Objekte wie Fahrspur, Ampel und andere Verkehrsteilnehmer richtig und zeitgerecht erkannt werden müssen [46].

Auch im medizinischen Bereich hat machine Learning einen großen Einfluss. Neuronale Netzwerke, insbesondere deep learning Methoden wie Convolutional neural Networks (siehe Kapitel 3.3.2), werden bereits für medizinische Bildanalyse wie Bildsegmentierung, Klassifizierung und Problemvorhersagungen verwendet und unterstützten dabei medizinische Experten [48]. Durch das Analysieren von Röntgenbildern der Brust können mithilfe von machine Learning COVID-19 Patienten, mit einer Genauigkeit von 96.78%, erkannt werden [49].

3.1 Wie computer sehen

Seit über 60 Jahren versuchen Wissenschaftler auf der ganzen Welt Maschinen beizubringen, wie sie bedeutungsvolle Informationen aus visuellen Daten extrahieren können [50]. Bereits im Jahre 1959 beschäftigten sich die Neurophysiologen David Hubel und Torsten Wiesel mit dem Thema Computer Vision, welches ein Teilbereich von künstlicher Intelligenz ist, und den Zweck verfolgt Systemen das erkennen von bedeutsamen Informationen aus Bildern oder Videos zu extrahieren [52, 50]. Ihre Publikation "Receptive fields of single neurons in the cat's striate cortex"[51] gilt als eines der Papers, welches am meisten Einfluss auf die Thematik genommen hat [52, 50]. Dabei beschreiben sie ihre Experimente, bei denen sie Elektroden an den primären Bereich des visuellen Kortex von anesthierten Katzen angeschlossen haben, und diese dabei beobachtet haben, während den Versuchstieren verschiedene Bilder gezeigt wurden. Durch einen glücklichen Zufall sind sie zur Erkenntnis gekommen, dass es sowohl einfache, als auch komplexe Neuronen im primären visuellen Kortex gibt, und dass visuelle Verarbeitung immer mit simplen Strukturen wie Kanten und Rändern beginnt. Dieser Ansatz findet Anwendung bei convolutional neural networks und ist im Wesentlichen das Grundprinzip von deep Learning. [50] Bevor es tiefer in diese Materie geht, wird im folgenden Unterkapitel kurz definiert und veranschaulicht was ein Bild ist und wie ein Computer solch eines wahrnimmt.

3.1.1 Was ist ein Bild?

tbd.

3.2 Bildererkennung und Klassifizierung

tbd.

3.3 Algorithmen für Bildähnlichkeitserkennung

3.3.1 SIFT

3.3.2 Convolutional Neuronale Netzwerke

Seit den 1980er Jahren werden convolutional neuronale Netzwerke für visuelle Aufgaben verwendet [53].

4 Prototyp zur Bildähnlichkeitserkennung von Markenlogos

4.1 Daten

4.2 Bildvorverarbeitung

4.3 Algorithmus

5 Diskussion der Ergebnisse

6 Conclusio

7 Ausblick

Literaturverzeichnis

- [1] Österreichisches Patentamt, “Statistische Übersicht über geschäftsumfang und geschäftstätigkeit in patentangelegenheiten gebrauchsmusterangelegenheiten markenangelegenheiten musterangelegenheiten,” Website, 2020, online erhältlich unter https://www.patentamt.at/fileadmin/root_oepa/Dateien/Allgemein/Statistiken/Stat2020_v1_1.pdf; abgerufen am 27. Februar 2022. 1
- [2] —, “MarkenÄhnlichkeitsrecherche,” Website, 2022, online erhältlich unter <https://www.patentamt.at/markenaehnlichkeitsrecherche/>; abgerufen am 27. Februar 2022. 1
- [3] Research and Markets, “Machine learning market by vertical bfsi, healthcare and life sciences, retail, telecommunication, government and defense, manufacturing, energy and utilities, deployment mode, service, organization size, and region - global forecast to 2022,” Website, 2017, online erhältlich unter <https://www.researchandmarkets.com/reports/4395173/machine-learning-market-by-vertical-bfsi/>; abgerufen am 18. März 2022. 1
- [4] V. Kurama, “MI-based image processing,” Website, 2021, online erhältlich unter <https://nanonets.com/blog/machine-learning-image-processing/>; abgerufen am 13. Mai 2022. 2
- [5] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems*, 2nd ed. O’Reilly, 2019. 2, 3, 4, 6, 7, 23
- [6] L. Wuttke, “Machine learning: Definition, algorithmen, methoden und beispiele,” Website, 2022, online erhältlich unter <https://datasolut.com/was-ist-machine-learning/>; abgerufen am 18. März 2022. 2, 23
- [7] Wikipedia, “Maschinelles lernen,” Website, 2022, online erhältlich unter https://de.wikipedia.org/wiki/Maschinelles_Lernen; abgerufen am 04. April 2022. 2
- [8] E. Burns, “machine learning,” Website, 2021, online erhältlich unter <https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML>; abgerufen am 04. April 2022. 2
- [9] Oracle, “Was ist machine learning?” Website, 2022, online erhältlich unter <https://www.oracle.com/at/data-science/machine-learning/what-is-machine-learning/>; abgerufen am 26. März 2022. 2, 3
- [10] Netflix, “Machine learning learning how to entertain the world,” Website, 2022, online erhältlich unter <https://research.netflix.com/research-area/machine-learning>; abgerufen am 26. März 2022. 2
- [11] H. Hodson, “What you ‘like’ on facebook gives away your personality,” Website, 2015, online erhältlich unter <https://www.newscientist.com/article/dn26781-what-you-like-on-facebook-gives-away-your-personality/>; abgerufen am 26. März 2022. 2
- [12] Intellipaat, “Supervised learning vs unsupervised learning vs reinforcement learning,” Website, 2022, online erhältlich unter <https://intellipaat.com/blog/>

- supervised-learning-vs-unsupervised-learning-vs-reinforcement-learning/; abgerufen am 09. April 2022. 3, 4
- [13] L. G. Serrano, *Grokking Machine Learning*, 4th ed. Manning, 2021. [Online]. Available: <https://livebook.manning.com/book/grokking-machine-learning/chapter-1/> 3, 4, 23
- [14] IBM, “Unsupervised learning,” Website, 2022, online erhältlich unter <https://www.ibm.com/cloud/learn/unsupervised-learning>; abgerufen am 02. Mai 2022. 3, 4, 5
- [15] L. Wuttke, “Was ist unsupervised learning (unüberwachtes lernen)?” Website, 2020, online erhältlich unter <https://datasolut.com/wiki/unsupervised-learning/>; abgerufen am 02. Mai 2022. 4, 5, 23
- [16] S. Priy, “Clustering in machine learning,” Website, 2021, online erhältlich unter <https://www.geeksforgeeks.org/clustering-in-machine-learning/>; abgerufen am 02. Mai 2022. 4, 5
- [17] Baysan, “What is association rule learning? an applied example in python: Basket analysis and product offering,” Website, 2021, online erhältlich unter <https://medium.com/codex/what-is-association-rule-learning-abd4a76144d8>; abgerufen am 02. Mai 2022. 5
- [18] N. Barla, “Dimensionality reduction for machine learning,” Website, 2021, online erhältlich unter <https://neptune.ai/blog/dimensionality-reduction>; abgerufen am 02. Mai 2022. 5
- [19] DataRobot, “Semi-supervised learning,” Website, 2021, online erhältlich unter <https://www.datarobot.com/blog/semi-supervised-learning/>; abgerufen am 03. Mai 2022. 5
- [20] Alexsoft, “Semi-supervised learning, explained with examples,” Website, 2022, online erhältlich unter <https://www.altexsoft.com/blog/semi-supervised-learning/>; abgerufen am 03. Mai 2022. 5, 6
- [21] S. Dobilas, “Self-training classifier: How to make any algorithm behave like a semi-supervised one,” Website, 2021, online erhältlich unter <https://towardsdatascience.com/self-training-classifier-how-to-make-any-algorithm-behave-like-a-semi-supervised-one-2958e7b54ab7>; abgerufen am 03. Mai 2022. 5
- [22] Wikipedia, “Co-training,” Website, 2021, online erhältlich unter <https://en.wikipedia.org/wiki/Co-training>; abgerufen am 04. Mai 2022. 6
- [23] C. Lemke, “So funktioniert reinforcement learning,” Website, 2019, online erhältlich unter <https://www.alexanderthamm.com/de/blog/einfach-erklart-so-funktioniert-reinforcement-learning/>; abgerufen am 04. Mai 2022. 6
- [24] L. Wuttke, “So funktioniert reinforcement learning,” Website, 2019, online erhältlich unter <https://datasolut.com/reinforcement-learning/>; abgerufen am 04. Mai 2022. 6, 23
- [25] —, “Künstliche neuronale netzwerke: Definition, einföhrung, arten und funktion,” Website, 2022, online erhältlich unter <https://datasolut.com/neuronale-netzwerke-einfuehrung/>; abgerufen am 10. Mai 2022. 7, 8, 9, 23
- [26] Wikipedia, “Künstliches neuronales netz,” Website, 2022, online erhältlich unter https://de.wikipedia.org/wiki/Künstliches_neuronales_Netz; abgerufen am 05. Mai 2022. 7, 8, 23
- [27] IBM, “What are neural networks?” Website, 2020, online erhältlich unter <https://www.ibm.com/cloud/learn/neural-networks>; abgerufen am 05. Mai 2022. 7
- [28] D. G. Inonos, “Neural networks: Was können künstliche neuronale netze?” Website, 2020, online erhältlich unter <https://www.ionos.at/digitalguide/online-marketing/suchmaschinenmarketing/was-ist-ein-neural-network/>; abgerufen am 05. Mai 2022. 7, 8

- [29] I. D. Baruah, “Understanding the basics of neural networks (for beginners),” Website, 2021, online erhältlich unter <https://medium.com/geekculture/understanding-the-basics-of-neural-networks-for-beginners-9c26630d08>; abgerufen am 07. Mai 2022. 7, 8
- [30] C. Nicholson, “A beginner’s guide to neural networks and deep learning,” Website, 2022, online erhältlich unter <https://wiki.pathmind.com/neural-network>; abgerufen am 05. Mai 2022. 8
- [31] E. Guresen and G. Kayakutlu, “Definition of artificial neural networks with comparison to other networks,” *Procedia Computer Science*, vol. 3, pp. 426–433, 2011, world Conference on Information Technology. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050910004461> 8
- [32] S. Sharma, “What the hell is perceptron?” Website, 2017, online erhältlich unter <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>; abgerufen am 11. Mai 2022. 9
- [33] Simplilearn, “What is perceptron: A beginners guide for perceptron,” Website, 2022, online erhältlich unter <https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron>; abgerufen am 11. Mai 2022. 9
- [34] A. Thamm, “Feedforward neural network,” Website, 2022, online erhältlich unter <https://www.alexanderthamm.com/de/data-science-glossar/feedforward-neural-network/>; abgerufen am 11. Mai 2022. 9
- [35] DeepAi, “Feed forward neural network,” Website, 2022, online erhältlich unter <https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network>; abgerufen am 11. Mai 2022. 9
- [36] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. 9
- [37] M. Saeed, “An introduction to recurrent neural networks and the math that powers them,” Website, 2021, online erhältlich unter <https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/>; abgerufen am 11. Mai 2022. 9
- [38] IBM, “Recurrent neural networks,” Website, 2022, online erhältlich unter <https://www.ibm.com/cloud/learn/recurrent-neural-networks>; abgerufen am 11. Mai 2022. 9
- [39] S. K. Sarvepalli, “Deep learning in neural networks: The science behind an artificial brain,” 10 2015. 9
- [40] S. F. Frauke Günther, “neuralnet: Training of neural networks,” 6 2010. 9
- [41] V. Bushaev, “How do we ‘train’ neural networks ?” Website, 2017, online erhältlich unter <https://towardsdatascience.com/how-do-we-train-neural-networks-edd985562b73>; abgerufen am 07. Mai 2022. 10
- [42] D. Johnson, “Back propagation neural network: What is backpropagation algorithm in machine learning?” Website, 2022, online erhältlich unter <https://www.guru99.com/backpropagation-neural-network.html>; abgerufen am 10. Mai 2022. 10
- [43] J.-L. Queguiner, “What does training neural networks mean?” Website, 2020, online erhältlich unter <https://blog.ovhcloud.com/what-does-training-neural-networks-mean/>; abgerufen am 10. Mai 2022. 10
- [44] W. Rowe, “What is a neural network? an introduction with examples,” Website, 2020,

- online erhältlich unter <https://www.bmc.com/blogs/neural-network-introduction/>; abgerufen am 11. Mai 2022. 11, 23
- [45] Peltarion, “Categorical crossentropy,” Website, 2022, online erhältlich unter <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy>; abgerufen am 12. Mai 2022. 11
 - [46] S. Fernandes, D. Duseja, and R. Muthalagu, “Application of image processing techniques for autonomous cars,” *Proceedings of Engineering and Technology Innovation*, vol. 17, pp. 01–12, Jan. 2021. [Online]. Available: <https://ojs.imeti.org/index.php/PETI/article/view/6074> 13
 - [47] Reuters, “Apple hires former tesla engineer to boost self-driving car effort,” Website, 2021, online erhältlich unter <https://www.reuters.com/technology/apple-hires-former-tesla-engineer-boost-car-effort-bloomberg-news-2021-11-05/>; abgerufen am 12. Mai 2022. 13
 - [48] Q. Ke, J. Zhang, W. Wei, D. Połap, M. Woźniak, L. Kośmider, and R. Damaševičius, “A neuro-heuristic approach for recognition of lung diseases from x-ray images,” *Expert Systems with Applications*, vol. 126, pp. 218–232, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417419300776> 13
 - [49] M. A. Elaziz, K. M. HosnyID, A. Salah, M. M. Darwish, S. Lu, and A. T. Sahlol, “New machine learning method for image-based diagnosis of covid-19,” May. 2020. 13
 - [50] R. Demush, “A brief history of computer vision (and convolutional neural networks),” Website, 2019, online erhältlich unter <https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3>; abgerufen am 12. Mai 2022. 13
 - [51] H. David and W. Torsten, “Receptive fields of single neurones in the cat’s striate cortex,” vol. 148, May. 1959. 13
 - [52] IBM, “What is computer vision?” Website, 2022, online erhältlich unter <https://www.ibm.com/topics/computer-vision>; abgerufen am 12. Mai 2022. 13
 - [53] W. Rawat and Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 09 2017. [Online]. Available: https://doi.org/10.1162/neco_a_00990 14

Abbildungsverzeichnis

2.1	Machine Learning nimmt Eingabedaten mit Beispielen und lernt daraus, um für die Zukunft Prognosen zu machen [6]	2
2.2	Labeled vs unlabeled Data [13]	3
2.3	"Model trainiert ohne Zielvariable und findet eigenständig Muster und Zusammenhänge in den Daten"[15]	4
2.4	Beispiel von Reinforcement Learning [24]	6
2.5	Multiple Layer in einem biologischen neuronalen Netzwerk [5]	7
2.6	Einfache Veranschaulichung eines neuronalen Netzwerkes [26]	8
2.7	Aufbau eines Neurons [25]	8
2.8	Beispielbilder aus dem MNIST Datensatz	10
2.9	Architektur des Neuronalen Netzwerks [44]	11
2.10	Vergleich 5 und 50 Epochen	12

Tabellenverzeichnis

Appendix

```
import keras
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot as plt
from keras.layers import Dense
from keras.models import Sequential

# load dataset
(pic_train, label_train), (pic_test, label_test) = mnist.load_data
()
# summarize loaded dataset
print('Train: X=%s, y=%s' % (pic_train.shape, label_train.shape))
print('Test: X=%s, y=%s' % (pic_test.shape, label_test.shape))
# plot first few images
fig, axes = plt.subplots(ncols=5, sharex=False,
    sharey=True, figsize=(10, 4))
for i in range(5):
    axes[i].set_title(label_train[i])
    axes[i].imshow(pic_train[i], cmap='gray')
    axes[i].get_xaxis().set_visible(False)
    axes[i].get_yaxis().set_visible(False)
plt.show()

num_classes = 10 # Zahlen von 0 bis 9
image_size = 28*28 # Pixels

# Gives a new shape to an array without changing its data.
x_train = pic_train.reshape(pic_train.shape[0], image_size)
x_test = pic_test.reshape(pic_test.shape[0], image_size)

# Converts a class vector (integers) to binary class matrix.
y_train = keras.utils.np_utils.to_categorical(label_train,
    num_classes)
y_test = keras.utils.np_utils.to_categorical(label_test,
    num_classes)

model = Sequential()

model.add(Dense(units=15, activation='sigmoid', input_shape=(
    image_size,)))
model.add(Dense(units=num_classes, activation='softmax'))
model.summary()
```



```
model.compile(loss='categorical_crossentropy', optimizer='sgd',
              metrics=['acc'])
history = model.fit(x_train, y_train, batch_size=128, epochs=5,
                   verbose=False, validation_split=.1)
loss, accuracy = model.evaluate(x_test, y_test, verbose=False)

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['training', 'validation'], loc='best')
plt.show()

print(f'Test loss: {loss:.3}')
print(f'Test accuracy: {accuracy:.3}')
```

Listing 1: Vollständiger Code für ein neuronales Netzwerk das Handschriftliche Zahlen erkennt