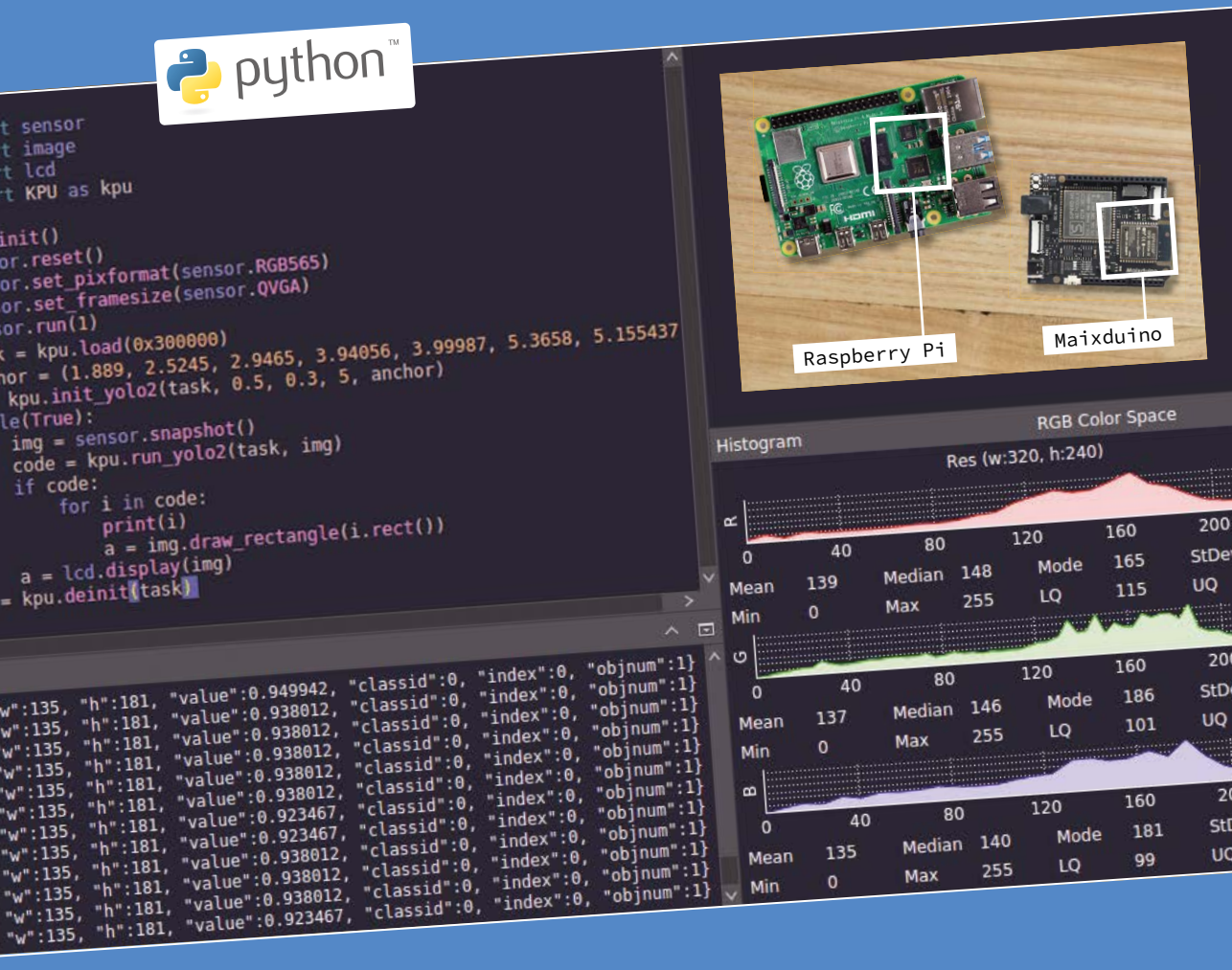
elektor books

Machine Learning mit Python

für PC, Raspberry Pi und Maixduino



Günter Spanner

elektor
design > share > sell

Machine Learning mit Python für PC, Raspberry Pi und Maixduino

Dr. Günter Spanner

elektor
design › share › sell

1. Auflage 2021

Umschlaggestaltung: Elektor, Aachen

Satz und Aufmachung: D-Vision, Julian van den Berg | Oss (NL)
Druck: Ipskamp Printing, Enschede, Niederlande

ISBN 978-3-89576-475-2 Print
ISBN 978-3-89576-476-9 eBook

Elektor-Verlag GmbH, Aachen
www.elektor.de

Warnhinweise

1. Die Schaltungen und Boards (Raspberry Pi, Maixduino) in diesem Buch dürfen nur mit geprüften, doppelt isolierten Sicherheitsnetzgeräten betrieben werden. Isolationsfehler eines einfachen Netzteils könnten zu lebensgefährlichen Spannungen an nicht isolierten Bauelementen führen.

2. Leistungsstarke LEDs können Augenschäden verursachen. Niemals direkt in eine LED blicken!

3. Autor und Verlag übernehmen keinerlei Haftung für Schäden, die durch den Aufbau der beschriebenen Projekte entstehen.

4. Elektronische Schaltungen können elektromagnetische Störstrahlungen aussenden. Da Verlag und Autor keinen Einfluss auf die technischen Ausführungen des Anwenders haben, ist dieser selbst für die Einhaltung der relevanten Emissionsgrenzwerte verantwortlich.

Programmdownload

Die Programme aus diesem Buch können unter

www.elektor.de

heruntergeladen werden. Sofern ein Programm nicht identisch mit dem im Buch beschriebenen ist, sollte die Version aus dem Download verwendet werden, da es sich dabei um die aktuellere Version handelt.

Kapitel 1 • Einführung

Nahezu alle Menschen sehen sich zunehmend mit den immer präsenter werdenden Anwendungen der "Künstlichen Intelligenz" (KI oder AI für engl. Artificial Intelligence) konfrontiert:

Musik- oder Videoempfehlungen, Navigationssysteme, Einkaufsvorschläge etc. basieren auf Verfahren, die mehr oder weniger diesem Bereich zugeordnet werden können. Häufig herrschen jedoch Unklarheiten, bezüglich der Begriffe

- Künstliche Intelligenz
- Maschinelles Lernen (Machine Learning - ML)
- Tiefes Lernen (Deep Learning - DL)
- Neuronale Netze (Neural Networks - NN)

Eine der häufigsten Anfragen bei gängigen Suchmaschinen lautet deshalb:

"Was ist der eigentliche Unterschied zwischen KI und Maschinellem Lernen?"

Die Begriffe hängen zwar eng zusammen, sind jedoch keineswegs gleichbedeutend:

- Künstliche Intelligenz ist im Prinzip die übergeordnete Wissenschaft. Sie bildet die Grundlage, so wie die Mathematik die Basis für Physik oder Elektrotechnik ist. Mit Hilfe von Maschinen sollen Probleme ohne starre Algorithmen gelöst werden.

- Maschinelles Lernen ist ein Teilgebiet der künstlichen Intelligenz. Dabei steht das automatische Lernen aus Erfahrungen, also ohne, dass dafür ein spezielles Programm erstellt wurde, im Vordergrund. Dafür existieren verschiedene Methoden z. B. Neuronale Netze oder aber auch allgemeine statistische Verfahren.

- Deep Learning oder Deep Neural Learning ist wiederum eine Untermenge des maschinellen Lernens. Hier werden mithilfe von Neuronalen Netzen Methoden und Systeme entwickelt, die biologischen Gehirnen nachempfunden sind.

Der Begriff "Künstliche Intelligenz" wurde 1956 auf einer internationalen Konferenz, dem Dartmouth Summer Research Project geprägt. Eine grundlegende Idee war es, die Funktionsweise des menschlichen Gehirns zu modellieren und darauf basierend fortschrittlichere Computersysteme zu konstruieren. Man ging davon aus, dass bald klar sein würde, wie der menschliche Verstand funktioniert. Die Übertragung auf eine Maschine sollte dann nur noch ein kleiner Schritt sein.

Dies sollte sich zwar als etwas zu optimistisch erweisen. Dennoch wurden einige wichtige Erkenntnisse zusammengetragen. So wurde z. B. festgehalten, dass die Schlüsselfaktoren für eine intelligente Maschine

- eigenständiges Lernen
- die Verarbeitung natürlicher Sprache und
- Kreativität

sein sollten.

Weisheit, Kreativität oder soziale Fähigkeiten sind Begriffe, die bis heute kaum auf Maschinen zutreffen. Die Aufgaben, auf die sich Wissenschaftler derzeit konzentrieren, sind vielmehr:

- Spracherkennung und maschinelle Sprachübersetzungen
- Medizinische Diagnose wie Auswertung von Röntgenaufnahmen oder Computertomogrammen
- Internet-Suchmaschinen
- Optimierung von Verkaufsstrategien
- Wissenschaftlich-technische Anwendungen u. a. in der Astro- und Geophysik
- Prognose von Aktien- und Devisenkursen
- Handschrifterkennung
- Personenidentifizierung, etwa über Gesichts- und Fingerabdruckerkennung

Alle diese Anwendungen erfordern zwar sicher ein gewisses Maß an ursprünglich dem Menschen vorbehaltenen Fähigkeiten. Ob es sich dabei aber wirklich bereits um "Intelligenz" handelt, wird vielfach in Zweifel gezogen. Es ist deshalb sinnvoll, zunächst einen Blick darauf zu werfen, was der Begriff Intelligenz überhaupt umfassen sollte.

1.1 In Drei Stufen zur "Superintelligenz"?

Zunächst kann man die Künstliche Intelligenz in zwei Bereiche unterteilen: schwache und starke KI. Die "Starke" oder Allgemeine KI ist im Wesentlichen der Versucht, künstliche Personen zu erschaffen. Das Ziel wären Maschinen oder Roboter, die alle mentalen Kräfte eines Menschen haben, einschließlich eines Bewusstseins [12].

Eine "Schwache" KI kann dagegen lediglich bestimmte Aufgabe hervorragend ausführen, erreicht jedoch in anderen Bereichen nicht annähernd menschliche Fähigkeiten. Beispiele für schwache KI sind Computer, die Menschen im Schach schlagen können. Allerdings nicht nur durchschnittliche Spieler, sondern sogar amtierende Schachweltmeister. Schwache Künstliche Intelligenz ist inzwischen in Wissenschaft, Wirtschaft und Gesundheitswesen weit verbreitet. KI-Systeme für Unternehmen enthalten beispielsweise leistungsstarke Analysetools und können Empfehlungen und Erkenntnisse für die Geschäftsentwicklung liefern. Sie werden für Risikomanagement und Planung eingesetzt.

Starke KI beginnt, sobald Maschinen "menschlich" werden, ihre eigenen Entscheidungen treffen oder ohne menschliche Hilfe neue Fähigkeiten erlernen. Diese Systeme sind nicht nur in der Lage, klassische mathematische oder logische Aufgaben zu lösen, sondern haben auch gewisse emotionelle Fähigkeiten. Es ist vergleichsweise einfach, Maschinen so zu programmieren, dass sie als Reaktion auf Reize einige "emotionale" oder verbale Aktionen ausführen. ChatBots und virtuelle Assistenten sind z. B. bereits recht gut darin, ein

Gespräch zu führen. Auch einige Experimente, bei welchen Roboter lernen, menschliche Emotionen zu lesen, wurden bereits erfolgreich durchgeführt. Hierbei handelte es sich aber eher um die Reproduktion emotionaler Reaktionen als um echte mentale Kompetenzen.

Ein weiterer Schritt wäre dann eine sogenannte "Superintelligenz". So werden Systeme oder Maschinen bezeichnet, die dem Menschen in vielen oder sogar allen Gebieten der Intelligenz überlegen sind. Dies umfasst alle oder zumindest viele Bereiche des Intellekts. Insbesondere kreative und problemlösende Fähigkeiten bis hin zu emotionalen und sozialen Kompetenzen sind dabei mit eingeschlossen. Auch Selbstbewusstsein oder ein umfassendes Erinnerungsvermögen gelten häufig als Voraussetzungen für eine echte Superintelligenz. Ob diese Funktionen auf biologischen oder technischen Grundlagen basieren, spielt dabei keine Rolle.

Überlegungen zu einer Superintelligenz führen allerdings rasch auf ein dystopisches Szenario. Wurde erst einmal eine Künstliche Intelligenz geschaffen, die vollkommen selbstständig weiter lernen kann, ist der Schritt zum übermenschlichen Intelligenzniveau letztlich nur noch eine Frage der Zeit. Die Kontrollierbarkeit einer solchen KI könnte letztendlich zu einer existenziellen Frage für die gesamte Menschheit werden.

Dass Wissenschaftler, die in diesem Bereich arbeiten und gut davon leben, diese Gefahr kleinreden, ist sicher nur allzu menschlich. Deshalb ist es wichtig, dass sich möglichst viele Menschen mit diesen Themen befassen. Nur so wird eine realistische Einschätzung der Potentiale und Risiken möglich sein.

1.2 Wie Maschinen lernen können

Maschinelles Lernen ist also ein Teilgebiet des umfassenden Feldes der künstlichen Intelligenz, das sich darauf konzentriert, wie man Systemen beibringt, zu lernen, ohne für bestimmte Aufgaben programmiert zu sein. Die Schlüsselidee hinter ML ist, dass es möglich ist, Maschinen zu entwickeln, die selbstständig aus Daten lernen, Wissen generieren und nutzbringende Vorhersagen treffen.

Um diesem Ziel näher zu kommen, müssen drei Voraussetzungen erfüllt sein:

1. Verfügbarkeit von umfangreichen Datensätzen

Diese können aus Zahlen, Bildern, Texten, Audioaufzeichnungen oder anderen Informationen bestehen. Das Erstellen der Datensätze ist häufig der aufwendigste Teil bei der Erstellung eines KI-Systems.

2. Formulierung eines Lernziels

Beim sogenannten überwachten Lernen enthalten die Trainingsdaten "richtige" und "falsche" Lösungen. Diese sind entsprechend gekennzeichnet. Während des Lernprozesses lernt das System, wie man zur korrekten Einschätzungen kommt.

Beim unüberwachten Lernen muss das System dagegen z. B. selbst erkennen, in welche Kategorien ein Datensatz unterteilt werden kann. Ein typisches Beispiel ist ein Datensatz mit Bildern von Menschen. So ist es bereits verschiedenen Systemen gelungen, selbstständig zu erkennen, dass diese in Männer und Frauen eingeteilt werden können.

3. Ein lernfähiges System

Um eine optimale Leistung zu erzielen, werden hier oft verschiedene Verfahren kombiniert. Zu Beginn der Lernphase enthält das ML-System nur eine größere Anzahl von leeren Parametern. Mit einem entsprechen Algorithmus werden die Parameter so lange variiert, bis das System möglichst korrekte Ergebnisse liefert. Obwohl Neuronale Netze hier eine weit verbreitete Variante darstellen, sind sie nicht die einzige Möglichkeit für die Entwicklung lernfähiger Systeme.

Deep Learning, also "Tiefes Lernen" ist eine Methode des Maschinellen Lernens, die von der Struktur eines menschlichen Gehirns inspiriert wurde. Mit Deep-Learning-Algorithmen können komplexe, mehrschichtige Neuronale Netze effizient trainiert werden. Insbesondere der "Back-Propagation"-Algorithmus gilt als einer der vielversprechendsten Ansätze.

In Neuronalen Netzwerken werden Informationen über virtuelle Verbindungskanäle übertragen. Beim Lernen oder "Trainieren" werden die Verbindungsstärken dieser Kanäle optimiert. Mittels umfangreicher Datenmengen wird so eine große Anzahl von Parametern bestimmt, sie schließlich ein einsatzfähiges System bilden.

So können nicht nur numerische oder visuelle Daten verarbeitet werden. Auch Spracherkennungssysteme sind auf diese Weise realisierbar. Die Schallwellen können als Spektrogramme dargestellt werden, welchen ein neuronales Netzwerk bestimmte Wörter zuordnen kann.

Allerdings existiert bisher kein perfekter, universeller Algorithmus, der für alle Aufgaben gleich gut funktioniert. Aktuell werden vier Gruppen von ML-Algorithmen unterschieden. Neben überwachtem und unüberwachtem Lernen unterscheidet man noch das halb-überwachte und das "Verstärkungslernen". Jede Variante hat ihre eigenen Stärken und Schwächen.

Überwachtes Lernen wird üblicherweise zur Klassifizierung und für Regressionen verwendet. Anwendungsbeispiele sind Spam-Filterung, Spracherkennung, Computer Vision, Bilderkennung und -klassifizierung.

Beim unbeaufsichtigten Lernen wird dem Programm keine Einteilung vorgegeben. Das System nimmt von sich aus eine Gruppeneinteilung vor. Unbeaufsichtigtes Lernen wird häufig verwendet, um Daten nach gewissen Ähnlichkeiten zu unterteilen. Unbeaufsichtigtes Lernen eignet sich daher gut für komplexe Datenanalysen. So können Maschinen Muster erkennen, die dem Menschen aufgrund seiner Unfähigkeit, riesige Datenmengen zu verarbeiten, verschlossen bleiben. Diese Variante ist u. a. geeignet, um betrügerische Finanz-Transaktionen zu entdecken, Umsätze zu prognostizieren oder Kundenpräferenzen zu

analysieren. Die klassischen Anwendungen sind hier Datensegmentierung, Erkennung von Anomalien, Kundenempfehlungssysteme, Risikomanagement, Analyse gefälschter Daten oder Bilder.

Beim halb-überwachten Lernen liegt eine Mischung aus beschrifteten und unbeschrifteten Daten vor. Prinzipiell sind gewisse Vorhersageergebnisse bekannt, aber das Modell kann und darf auch eigene Muster finden, um Daten zu strukturieren und Vorhersagen zu optimieren.

Das Verstärkungs- bzw. Reinforcement-Lernen kommt dem menschlichen Pendant am nächsten. Menschen brauchen keine ständige Aufsicht, um effektiv zu lernen. Vielmehr werden aus positiven oder negativen Erfahrungen immer neue Lehren gezogen. Das Berühren eines heißen Kochtopfes oder das Überfahren eines Reißnagels mit dem Fahrrad führen zu entsprechenden Lernerfolgen. Ein Training mit vorgefertigten Datensätze wird überflüssig. Stattdessen kann das System in dynamischen, realen Umgebungen lernen.

In einem ersten Schritt werden in der verstärkten Lernforschung oftmals simulierte Spielwelten eingesetzt. Diese bieten ideale, datenreiche Umgebungen. Die erreichbaren Punktzahlen in diesen Spielen sind bestens dazu geeignet, um belohnungsmotivierte Verhaltensweisen zu trainieren. Anwendungen sind hier vor allem Autopilotensysteme, selbstfahrende Autos und Züge oder autonome Roboter [12].

In einigen Bereichen wurden mit diesen Methoden bereits Systeme entwickelt, die dem Menschen überlegen sind. So werden etwa bei der Diagnose von Röntgenbildern bestimmte Krebsarten von ML-Systemen bereits besser erkannt als von geschulten Röntgenfachärzten. Zudem können diese Maschinen im Gegensatz zu den Ärzten 24 Stunden pro Tag und 7 Tage die Woche Routineauswertungen vornehmen.

Kapitel 2 • Eine kleine Geschichte der KI

Obwohl der Begriff Künstliche Intelligenz erstmals in Dartmouth in Erscheinung trat, ist die Frage, ob Maschinen wirklich denken können, viel älter. In einer berühmten Arbeit mit dem Titel "As We May Think" wurde bereits 1945 ein System vorgeschlagen, welches die Wissensverarbeitung an die Denkweisen des Menschen anlehnen sollte. Wenig später veröffentlichte Alan Turing einen Fachartikel darüber, dass Maschinen auch Aufgaben übernehmen könnten, die ein hohes Maß an "Intelligenz" erfordern. In diesem Zusammenhang wurde auch ausdrücklich das Schachspiel genannt.

Inzwischen sind die extremen Rechenleistungen moderner Computer unbestritten. Trotzdem wird bezweifelt, ob eine Maschine wirklich "intelligent denken" kann. Schon die fehlende exakte Definition des Begriffs "Intelligenz" ist problematisch. Zudem sind wichtigste Fortschritte im Bereich der KI für viele Menschen gar nicht erkennbar. Die neuen Methoden werden häufig auf sehr subtile Weise eingesetzt. So besteht bei Untersuchung des Kaufverhaltens und der anschließenden Beeinflussung von Einkaufsentscheidungen kein Interesse an einer diesbezüglichen öffentlichen Aufklärung.

Darüber hinaus zeigt sich eine gewisse Tendenz, immer wieder neu zu definieren, was "intelligent" bedeutet. Wenn Maschinen ein Problem gelöst haben, wird dieses hinterher schnell als trivial oder als einfache Rechenleistung betrachtet. So galt Schach als "Spiel der Könige" jahrhundertelang als massiv Intelligenz erforderndes Strategiespiel. Nachdem der Schachweltmeister geschlagen war, galt das Spiel nur noch als "Rechenleistung".

Die Zukunft der KI wird besser verständlich, wenn man einige wichtige Punkte ihrer Historie betrachtet. Nach der Dartmouth Konferenz war einer der nächsten Meilensteine ein Computerprogramm, das über Tastatur und Bildschirm mit Menschen kommunizieren konnte. "ELIZA" überraschte viele Leute mit der Illusion eines menschlichen Gesprächspartners.

Für erste medizinische Anwendungen wurden schließlich Expertensysteme eingesetzt. Entsprechende Computerprogramme unterstützten Ärzte bei Diagnosen und Routineanalysen. Seit 1986 lernen Computer langsam das Sprechen. Über vorgegebene Lautfolgen waren Rechner erstmals in der Lage, ganze Wörter verständlich auszusprechen.

Schließlich ebneten weitere technologische Entwicklungen der künstlichen Intelligenz den Weg in den Alltag. Leistungsstarke Prozessoren in Smartphones und Tablets bieten Verbrauchern umfangreiche KI-Anwendungen. Apples "Siri", Microsofts "Cortana"-Software, Amazons "Alexa" erobern die Märkte. Seit 2018 diskutieren KI-Systeme über Weltraumfahrt und vereinbaren Friseurtermine.

Schließlich gelang es einem System namens "AlphaGo", den amtierenden Weltmeister im Go-Spiel zu schlagen. Dieses asiatische Brettspiel weist deutlich mehr Varianten auf als Schach. Mit der reinen Vorausberechnung von Spielzüge sind die Erfolgsaussichten daher äußerst gering. Mit Hilfe von KI-Methoden gelang dann jedoch das scheinbar Unmögliche. Dabei erregte ein sehr spezieller Spielzug besondere Aufmerksamkeit. Hier wurde erst viel

später klar, wie brillant dieser war. Viele Experten sprachen bereits von Kreativität und Intuition seitens der Maschine.

Wenig später erlernte ein Nachfolgesystem das Spiel sogar ohne jegliche menschliche Unterstützung. Damit erreichten selbstlernende Maschinen eine neue Dimension. Das aufwendige Training mit menschlichen Partnern war überflüssig geworden. Das neue System erschloss sich die Grundlagen und Feinheiten des Spiels, und entwickelte sogar eigenständige Strategien.

Bei Schach und Go liegen sämtliche Informationen für alle Beteiligten offen. Poker dagegen ist ein Spiel mit unvollständigen bzw. verdeckten Informationen. Damit bekommen "Intuition" und die Einschätzung anderer Spieler eine besondere Bedeutung. Dennoch gelangen den Maschinen auch hier entscheidende Erfolge, indem sie wichtige Poker-Turniere für sich entscheiden konnten. So sind KI-Systeme inzwischen in der Lage, das Poker-Spiel selbstständig zu erlernen. Einige entdeckten sogar die Fähigkeit zu bluffen. Die beim Poker gewonnenen Erkenntnisse sind auch in anderen Bereichen wie der Medizin oder in professionellen Verhandlungstechniken von Interesse. Auch hier spielt die Intuition häufig eine wichtige Rolle. Deshalb sind Medizinstudenten erfahrenen Ärzten bei der Diagnose von Krankheiten meist unterlegen. Auch in der Finanzwirtschaft und in der Wertpapieranlage sind Erfahrung und Intuition gefragt. Deshalb finden sich in diesen Bereichen zunehmend Anwendungen für die Erkenntnisse aus den Pokersiegen der KI.

Dass die Entwicklung der KI keineswegs frei von Rückschlägen und Krisen war, zeigt u.a. das sogenannte XOR-Problem. Das XOR- oder "Exklusiv-ODER"-Gatter ist ein klassisches Problem in der Technik. Die Aufgabe besteht darin, ein Netzwerkelement zu konstruieren, das eine XOR-Logik emuliert. Diese liefert eine Eins am Ausgang, wenn die beiden Eingänge nicht gleich sind. Wenn die Eingänge dagegen gleich sind, wird eine Null ausgegeben (s. Tabelle):

Input 1	Input 1	Input 1
0	0	0
0	1	1
1	0	1
1	1	0

Oberflächlich betrachtet scheint das XOR-Gatter ein sehr einfaches Problem zu sein. Es zeigte sich jedoch, dass diese Funktion ein großes Problem für neuronale Netzwerkarchitekturen darstellt.

Sogenannte "Perzeptrons" (von engl. perception, "Wahrnehmung") sind elementare Einheiten, die biologischen Neuronen entsprechen. Sie bilden daher die Basis künstlicher neuronaler Netze. Jede Einheit kann eine Eingabe von anderen Einheiten empfangen. Im Perzeptron wird die Summe aller empfangenen Werte gebildet und anschließend entschieden, ob ein Signal an andere Einheiten weitergeleitet wird. Das XOR-Problem ist jedoch nicht linear trennbar. Dies lässt sich am besten zeigen, wenn die XOR-Eingabewerte in einem

Diagramm dargestellt werden. Wie aus Abbildung 2.1 ersichtlich ist, gibt es keine Möglichkeit, die Vorhersagen durch eine einzige gerade Klassifizierungslinie zu trennen. Ein Perzeptron kann jedoch nur Klassen trennen, die durch gerade Linien separierbar sind.

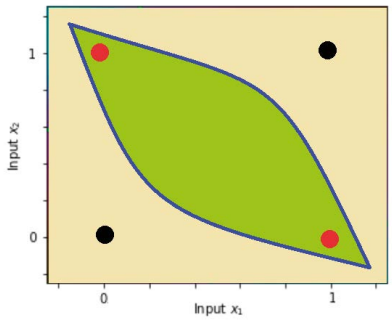


Abbildung 2.1: XOR-Problem

Deshalb kann das XOR-Problem nicht mit einem Perzeptron gelöst werden. Die anderen logischen Operatoren AND, OR und NOT konnten problemlos emuliert werden, nicht jedoch die XOR-Funktion. Diese Erkenntnis stürzte die KI-Entwicklung in eine tiefe Krise. Es dauerte mehrere Jahre, bis sich einzelne Forschergruppen wieder mit dem Thema Neuronale Netze und Machine Learning ernsthaft auseinander setzten.

Die Lösung des Problems besteht darin, über die einschichtige Architektur hinaus zu gehen. Zusätzliche Schichten von "Neuronen", insbesondere sogenannte verborgene Schichten, erlaubten es schließlich das XOR-Problem zu lösen. Wie diese Lösung im Detail aussieht, wird in Kapitel 10.9 ausführlich erläutert.

Trotz ihrer inzwischen recht langen, wechselvollen Geschichte und ihrer beachtlichen Erfolge ist die Künstliche Intelligenz in vielen Bereichen noch nicht annähernd ausgereift. Viele Anwendungen müssen noch wesentlich zuverlässiger bzw. fehlertoleranter werden. Sensible Bereiche wie autonomes Fahren oder medizinische Diagnosen erfordern ein Höchstmaß an Zuverlässigkeit. Zudem müssen KI-Systeme zunehmend transparent werden, Entscheidungen müssen für Menschen nachvollziehbar bleiben. Andernfalls wird die KI-basierte Vergabe eines Kredits oder eine medizinische Therapieentscheidung im Alltagsleben niemals wirklich akzeptiert werden.

Kapitel 3 • Lernen aus großen Datenmengen

Der Forschungsbereich "Machine Learning" hat unter anderem die Aufgabe, Daten zu analysieren, zu erkennen oder zu interpretieren. Eine große Ansammlung von Daten ist zunächst meist mehr oder weniger nutzlos. Erst die Kategorisierung und Einteilung in sinnvolle Gruppen, statistische Auswertungen oder Trendanalysen etc. geben den Datenmengen einen gewissen praktischen Nutzen. Ein Ziel der Entwicklungsarbeit im Rahmen des maschinellen Lernens ist es, Daten in Wissen umzuwandeln und daraus nützliche Schlussfolgerungen zu ziehen.

Lange Zeit war die Anwendung von ML-Verfahren einem kleinen Kreis von Experten vorbehalten. Die verwendeten Algorithmen waren komplex und für Laien kaum verständlich. Erst durch die Entwicklung von Open-Source-Bibliotheken bekamen auch Nicht-Fachleute die Chance, sich intensiver mit den Themen KI und Machine Learning zu befassen. Damit wurden einem weiten Anwenderkreis Verfahren zugänglich, die es erlauben, aus komplexen Datenstrukturen nützliches Wissen zu extrahieren. Insbesondere die Programmiersprache "Python" erlaubt es, umfangreiche Funktionen in einfache Verpackungen (engl. "Wrapper") einzubetten. Mit Hilfe von simplen Python-Anweisungen kann nun praktisch jeder die leistungsfähigen Algorithmen einsetzen. Dafür sind keine Supercomputer erforderlich. Vielmehr können auch kleine Systeme Muster in großen Datenmengen erkennen oder Vorhersagen über zukünftige Ereignisse ableiten [2].

Im nächsten Abschnitt werden Aufbau und Inhalt der Forschungs- und Entwicklungsbereiche Künstliche Intelligenz und Machine Learning vorgestellt. Zudem wird die zugehörige Terminologie genauer erläutert werden. So wird die Basis für die Lösung von praxisrelevanten Anwendungen geschaffen. Dabei werden insbesondere die folgenden Themen eine zentrale Rolle spielen:

- Verschiedene Varianten des Machine Learning und der KI
- Lernfähige Systeme, Neuronale Netze und Trainingsmethoden
- Installation einer Entwicklungsumgebung für die Programmiersprache Python bzw. deren interaktiver Variante IPython
- Erstellen einer Softwarebasis für ML-Methoden und -verfahren

3.1 Maschinelles Lernen und Künstliche Intelligenz

Die Forschungsgebiete der Künstlichen Intelligenz sind eng miteinander verbunden. Oft fällt es schwer, einzelne Teilbereiche voneinander zu trennen. Neben den Methoden des maschinellen Lernens existieren zudem noch weitere Gebiete, wie etwa die klassischen Expertensysteme oder sogenannte Evolutionäre Algorithmen. Viele dieser Teilbereiche haben allerdings in den letzten Jahren etwas an Bedeutung verloren, da sich die aktuellen Entwicklungen zunehmend auf Neuronale Netze und Deep Learning konzentrieren.

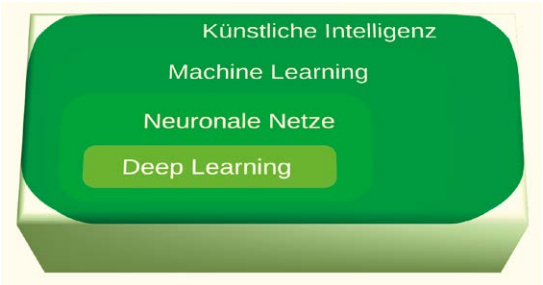


Abbildung 3.1: Inhalte der Künstlichen Intelligenz

Das Maschinelle Lernen als Teilbereich der künstlichen Intelligenz wird immer mehr zum zentralen Kern der aktuellen Forschung. Das selbstständige "Lernen" aus umfangreichen Datenmengen ersetzt die personalaufwendige und problemspezifische explizite Programmierung. Die Verarbeitung von Daten mit bekannten Zusammenhängen und das "Lernen" von Strukturen durch sogenanntes "Trainieren" spielen dabei eine zentrale Rolle. Später können die erlernten Regeln auch auf neue Daten und ursprünglich unbekannte Zusammenhänge angewendet werden. Die neuen Verfahren analysieren bekannte Daten, extrahieren daraus bestimmte Regeln und treffen schließlich darauf basierende Entscheidungen. Mit Hilfe verschiedener mathematischer Methoden werden Datensätzen in hierarchische Ebenen aufgeteilt. So können schließlich praktische Anwendungen wie Bilderkennung bzw. -kategorisierung entstehen. In Python kann die komplexe Mathematik der Neuronalen Netze auf einfache Funktionen reduziert werden. So wird es möglich, relativ schnell praktische Projekte umzusetzen und relevante Ergebnisse zu erhalten.

Das sogenannte "Deep Learning" kann wiederum als Spezialgebiet der Machine-Learning-Verfahren betrachtet werden. Hier geht es darum, das menschliche Lernverhalten möglichst effektiv nachzuahmen. Die Basis hierfür ist die Verwendung großer Datenmengen als "Lernmaterial". Tiefe Lernalgorithmen für Neuronale Netzwerke liefern aktuell die besten Ergebnisse im Bereich der Bild- oder Mustererkennung etc. Die dazu verwendeten Neuronalen Netze bestehen meist aus mehreren Zwischenschichten, welche als Verbindung zwischen den Eingangs- und Ausgangsknoten dienen.

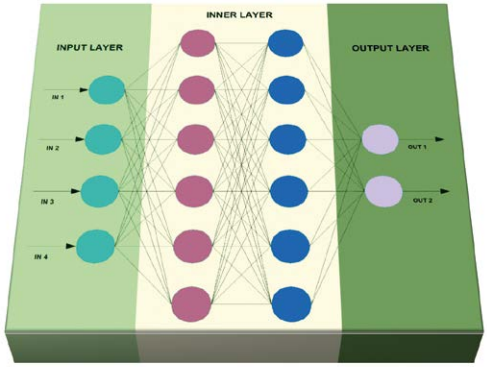


Abbildung 3.2: Neuronales Netzwerk

Die Schichten oder "Layer" enthalten sogenannte künstliche Neuronen. Die Funktion dieser Neuronen wird durch mathematische Funktionen emuliert (s. Kapitel 11). Das Ziel dieser Entwicklung besteht darin, die Struktur des menschlichen Gehirns mit seinem Netzwerk aus Neuronen, Axonen, Dendriten und Synapsen möglichst effizient nachzubilden.

Tiefes Lernen ist gekennzeichnet durch die Verwendung mehrerer Schichten in den Neuronalen Netzwerken. Die Eingangsschicht (Input Layer) verarbeitet entweder die direkten Rohdaten, beispielsweise die einzelnen Pixel eines Bildes, oder aber bereits vorverarbeitete bzw. gefilterte Datenströme. Die inneren bzw. verborgenen Schichten (engl. Inner oder Hidden Layer) und Ebenen übernehmen die Weiterverarbeitung und Datenreduktion. Die Ausgabeschicht (Output Layer) liefert schließlich die Ergebnisse. Der typische Aufbau eines solchen Neuronalen Netzes ist in Abbildung 3.2 dargestellt.

Durch die Deep-Learning-Methoden können auch komplexe Zusammenhänge modelliert werden. Die Vielzahl von Netzebenen erlaubt es auch, hochgradig nichtlineare Funktionen abzubilden. Ohne diese Struktur wären bereits einfache logische Gatter-Funktionen unlösbar, wie etwa das Beispiel des XOR-Gates (s. Kapitel 2) zeigt. Tiefe Lernalgorithmen sind dagegen in der Lage, auch vergleichsweise aufwendige Aufgaben zu lösen. Durch mehrfache "Trainingsläufe" kann das vertiefte Lernen mit jedem Berechnungsschritt verbessert werden. Damit hat es sich zu einem der zentralen Entwicklungstreiber im Bereich der Künstlichen Intelligenz entwickelt [11].

Die wesentlichen Unterschiede zwischen Machine Learning und Deep Learning sind nochmals in der folgenden Tabelle zusammengefasst:

	Lerndaten	**Größe der zum Lernen erforderlichen Datensätze**	**Notwendige Hardware-Struktur**	**Erforderliche Trainings-laufzeiten**	**Interpretation der Entschei-dungen**
Machine Learning	Strukturiert	Alle Größen	Einfach	Kurz	Einfach
Deep Learning	Keine speziellen Erfordernisse	Umfangreich	Anspruchsvoll	Tage bis Wochen	Nahezu unmöglich

Ein wichtiger Vorteil des Tiefen Lernens im Vergleich zum allgemeinen Machine Learning ist die Fähigkeit, auch unstrukturierte Daten zu verarbeiten. So lassen sich auch reale Informationen wie Bilder, Töne und Videos etc. mehr oder weniger direkt als Eingabedaten verwenden. Andere Machine Learning Algorithmen, wie z. B. Entscheidungsbaumverfahren, besitzen diese Fähigkeit nicht. Sollen hier etwa Bilder als Eingabedaten genutzt werden, muss immer eine aufwendige und spezielle Programmanpassung durch Menschen erfolgen.

Beispiele für KI-Verfahren, welche ohne Neuronale Netze auskommen, sind die sogenannten Evolutionären Algorithmen. Diese Programme basieren ebenfalls auf einem Lernverfahren der Natur. Jedoch wird hier nicht das Gehirn nachgebildet, sondern die Evolution. Entsprechend der biologischen Variante folgen hier die Algorithmen in verschiedenen Generationen aufeinander. Nach dem Darwin-Prinzip des "Survival of the Fittest" wird jede

mögliche Lösung bewertet und die jeweils beste ausgewählt. Die "mutierten" Varianten führen so zu einem optimierten, d. h. immer besser an das vorgegebene Problem angepassten Algorithmus.

Die letzte Spalte der Tabelle deutet ein unter Umständen gravierendes Problem von KI-Algorithmen an. Bei komplexen Neuronalen Netzen kann meist nicht mehr nachvollzogen werden, wie Entscheidungen genau getroffen wurden. Bei massiven Fehlentscheidungen von KI-Systemen kann dies zu erheblichen juristischen oder sozialen Problemen führen. Letztendlich wird sogar die Akzeptanz von entsprechenden Entscheidungen wesentlich von deren Nachvollziehbarkeit abhängen. KI-Anwendungen, welche hier nicht über ein Mindestmaß an Transparenz verfügen, werden sich daher kaum in (lebens-)wichtigen Bereichen durchsetzen können.

Kapitel 4 • Hardwarebasis

Bis vor wenigen Jahren herrschte die Meinung vor, Machine Learning oder KI könne nur auf Server-Farmen oder "Super-Computern" laufen. Die Hochleistungsmaschinen, die im Schach gewannen, Jeopardy-Meister wurden oder menschliche GO-Spezialisten besiegten, prägten das Bild der KI in der Öffentlichkeit. Mit den Fortschritten in Hard- und Software hat sich die Lage jedoch dramatisch geändert. Selbst ein Mittelklasse-PC verfügt heute über mehrere Gigabyte an Arbeitsspeicher und auch Kleinrechner wie ein Raspberry Pi 4 sind mit bis zu 8 GB Speicher verfügbar. Sogar auf Mikrocontrollern wie dem ESP32 erbringen inzwischen ML-Applikationen erstaunliche Leistungen. Daneben kommen immer häufiger spezielle Chips wie der Kendrite K210 auf den Markt, die bereits hardwareseitig auf neuronale Strukturen hin ausgelegt sind.

In diesem Buch sollen daher mehrere verschiedene Systeme zum Einsatz kommen. Neben dem klassischen Universal-PC werden sowohl der Raspberry Pi als auch der sogenannte "Maixduino" in den einzelnen Projekten ihre Fähigkeiten beweisen.

Die folgende Tabelle liefert eine Übersicht, welche Projekte mit welcher Hardware durchgeführt werden können. Dies bedeutet nicht, dass es nicht möglich wäre, die ein oder andere Lösung auch auf eine andere Hardwarebasis zu portieren. Allerdings ist es beispielsweise relativ aufwendig, mit einem PC direkt weitere Geräte zu steuern, da ein PC normalerweise nicht über entsprechende Schnittstellen verfügt. Mit dem Raspberry Pi oder dem Maixduino dagegen können über die vorhandenen Pins problemlos LEDs oder Relais angesteuert werden. Spracherkennung und -synthese wären dagegen auch auf einem PC machbar. Die Implementierung in Python ist jedoch auf einem Raspberry wesentlich einfacher.

	PC, Laptop	Raspberry Pi	Sipeed Maixduino
Grundlagen	X	X	
Iris-Klassifizierung	X	X	
Handschriften-Erkennung	X	X	X
Kleidungssortierung	X	X	
Spracherkennung /-synthese		X	
Objekterkennung		X	X
Gesichtserkennung		X	X
Heimautomatisierung		X	X
Größe (ca. cm)	30 x 40 x 15	9 x 6 x 1,5	7 x 6 x 1
Preis (ca. €)	500	50	30
Stromaufnahme (W)	50	10	5

Kapitel 5 • Der PC als universelle KI-Maschine

Ein moderner PC oder Laptop kann auch im Bereich des Machine Learnings gute Dienste leisten. Allerdings sollten dafür einige Voraussetzungen erfüllt sein. Die folgende Tabelle liefert einen Überblick zu den Minimalanforderungen:

CPU	Quadcore mit mindestens 3 GHz
RAM	mindestens 16 GB
Festplattenkapazität	1 TB

Sicher kann man auch mit einer etwas geringeren Ausstattung bereits einige Einstiegsprojekte umsetzen. Allerdings werden dann beispielsweise die Trainingszeiten für Neuronale Netzwerke bereits für einfachere Anwendungen sehr lang.

5.1 Der Computer als Programmierzentrale

Der PC dient nicht nur als Grundlage für den Einstieg in ML-Projekte. Vielmehr ist er auch für andere Zwecke erforderlich. So kann etwa der Maixduino nicht direkt programmiert werden. Der PC ist hier als "Hostrechner" notwendig, welcher über die USB-Schnittstelle das Maix-Board programmiert. Hier kann zudem ein aktiver USB-Hub gute Dienste leisten. Er hat den besonderen Vorteil, dass er einen gewissen Schutz für den USB-Port des verwendeten PCs oder Laptops bietet. Wichtig ist, dass es sich tatsächlich um einen **aktiven** Hub handelt. D. h. das Gerät muss über eine eigene 5-V-Stromversorgung über ein separates Netzteil verfügen. Nur so kann gewährleistet werden, dass im Falle eines Kurzschlusses hinter dem Hub der Rechner-Port geschützt ist.

Abbildung 5.1: Aktiver USB-Hub

Natürlich bietet auch ein aktiver Hub keinen absoluten Schutz. Das "Durchschlagen" eines Kurzschlusses über einen aktiven Hub hinweg bis zum USB-Port des Rechners ist jedoch sehr unwahrscheinlich.

Will man das Maix-Board ohne PC, d. h. im Stand-Alone-Betrieb verwenden, dann ist ebenfalls ein externes 5 V-Netzteil erforderlich. Im Idealfall kann hierfür auch das Netzteil des Hubs verwendet werden. So ist das Board nach seiner Programmierung unabhängig von einem USB-Port einsetzbar. Wichtig ist, dass das Netzteil ausreichend Strom liefern kann. 2000 mA Nennstrom sollten das Minimum sein. Falls das Netzteil noch über einen USB-Micro-Connector verfügt, ist zusätzlich ein Adapter auf USB-C (s. Abb. 7.4) erforderlich. Weitere Informationen zum Stand-Alone-Betrieb finden sich in Abschnitt 2.

Abbildung 5.2: USB-C-Adapter

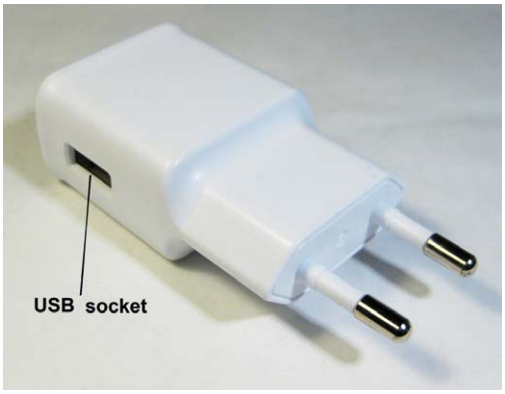


Abbildung 5.3: USB-Netzteil

An dieser Stelle sei noch erwähnt dass der Maixduino, ähnlich wie der Arduino, auch über einen integrierten Spannungsregler verfügt. Damit kann die Versorgung der Boards auch über ein beliebiges Steckernetzteil mit einer Ausgangsspannung von 6 bis 12 V erfolgen. Das Netzteil muss einen Strom von mindestens 2 A liefern können. Für die Verbindung zum jeweiligen Board muss das Netzteil über einen Standard-Hohlbuchsenstecker verfügen.

Bei verschiedenen Anwendungen hat sich gezeigt, dass das Maix-Board etwas stabiler läuft, wenn es zusätzlich über die Hohlbuchse mit Spannung versorgt wird. Insbesondere bei Verwendung einer SD-Karte und wenn zusätzliche Hardware an den Pins des Maixduinos angeschlossen ist, scheint die USB-Versorgung an ihre Grenzen zu kommen. Auch hierzu liefert der Abschnitt 7.4 weitere Hinweise.

Kapitel 6 • Raspberry Pi

Der Raspberry Pi sich in den letzten Jahren zu einem der beliebtesten Controllerboards überhaupt entwickelt. Obwohl er häufig in hardwarenahen Projekten eingesetzt wird, kann er auch für Machine-Learning-Anwendungen erfolgreich verwendet werden. Insbesondere die neuere Generation des Raspberry Pi 4 bringt dafür bereits ausreichend Rechenleistung mit. Allerdings ist die verfügbare RAM-Größe von entscheidender Bedeutung. Mit einem Pi 4 mit 8 GB RAM ist man bestens gerüstet. Mit der schnelleren CPU, der neuen GPU, 4K-Unterstützung, USB 3.0, USB-C, Bluetooth 5.0 und Gigabit Ethernet lassen sich auch anspruchsvollere Projekte umsetzen. Der Raspberry Pi 4 setzt damit einen weiteren Meilenstein hinsichtlich Performance und Ausstattung. Der Pi 4 ist 3x schneller als sein Vorgänger und erlaubt deutlich schnellere Multimedia-Leistungen, die insbesondere bei der Bildverarbeitung von Vorteil sind. Insgesamt reicht der Pi 4 damit in vielerlei Hinsicht an die Leistung eines x86-basierten PCs heran.

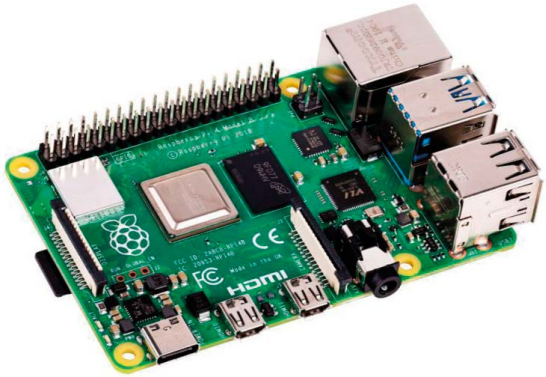


Abbildung 6.1: Raspberry Pi 4

Die folgende Tabelle fasst die wichtigsten Parameter des Pi 4 zusammen:

Prozessor:	64-bit-Quad-Core-ARM-Cortex-A72 (4x 1,5 GHz)
Video	Dual-Display-Unterstützung mit Auflösungen von bis zu 4K über zwei Micro-HDMI-Ports Hardware-Videodekodierung mit bis zu 4Kp60 2x Micro-HDMI (bis zu 4Kp60 Support) 2-Kanal MIPI DSI-Port (Display) 2-Kanal MIPI CSI-Port (Kamera)
Audio:	4-poliger Stereo-Audio-Anschluss
RAM	bis zu 8 GB RAM - LPDDR4
WLAN:	Dual-Band mit 2,4/5 GHz 2,4 GHz und 5 GHz IEEE 802.11b/g/n/ac wireless LAN
Bluetooth:	5.0 BLE
LAN:	Gigabit-Ethernet
Schnittstellen:	2x USB 3.0 2x USB 2.0

GPIO:	Standard 40-Pin GPIO-Header (kompatibel zu früheren Boards)
SD-Karte:	microSD (für Betriebssystem und Datenspeicherung)
Stromversorgung:	5 V / 3 A (via USB-C)

6.1 Remote Desktop

Der Raspberry Pi verfügt über 4 USB-Ports und eine bzw. im Falle des Pi 4 sogar zwei HDMI-Schnittstellen. Damit kann der Minirechner mit Tastatur, Maus und Bildschirm ausgestattet werden. Alternativ ist das Board aber auch über den Windows Remote Desktop steuerbar. Diese Variante bietet einige Vorteile. So kann man Tastatur, Maus und Bildschirm gemeinsam mit dem PC nutzen. Die eigene Ausrüstung für den "RasPi" entfällt. Dies ist insbesondere bei beengten Platzverhältnissen vorteilhaft. Man erspart sich also z. B. einen zusätzlichen Monitor oder das Hin- und Herwechseln zwischen den beiden Systemen.

Eine Remote-Desktop-Verbindung zum Raspberry Pi erlaubt die vollständige Steuerung des Raspberry über LAN oder WLAN. Prinzipiell ist der Raspberry Pi zwar auch fast ausschließlich per Konsole steuerbar, insbesondere bei ML-Anwendungen ist jedoch eine grafische Benutzeroberfläche meist vorteilhaft.

Die Remote-Desktop-Verbindung ist auf allen modernen Windows Systemen bereits vorinstalliert und bietet sich daher geradezu für die Steuerung des Raspberry an. Falls man den Pi also fernsteuern will, ist die Remote-Desktop-Verbindung auch im Hinblick auf das Datenvolumen eine sehr effiziente Variante.

Auf Seiten des Pi ist lediglich ein einziges Paket erforderlich. Dieses kann über

```
sudo apt-get install xrdp
```

problemlos installiert werden. Alle wichtigen Einstellungen sind bereits vordefiniert. Man kann sich also unmittelbar nach Fertigstellung der Installation auf dem Pi einloggen. Auf einem Windows PC befindet sich die "Remote-Desktop-Verbindung" im Startmenü.

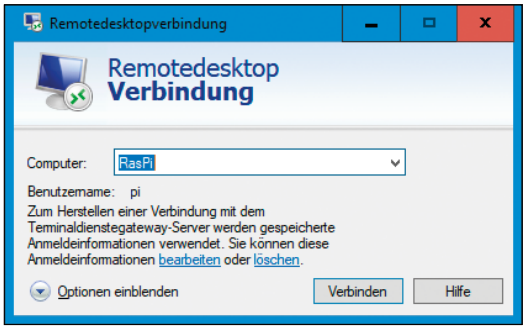


Abbildung 6.2: Startfenster des Remote-Desktops

Im Startfenster wird als Hostname entweder die IP des Pi angegeben oder der Name des Raspberry (Standard: raspberrypi). Danach wird bereits der Login-Bildschirm des Pi angezeigt.

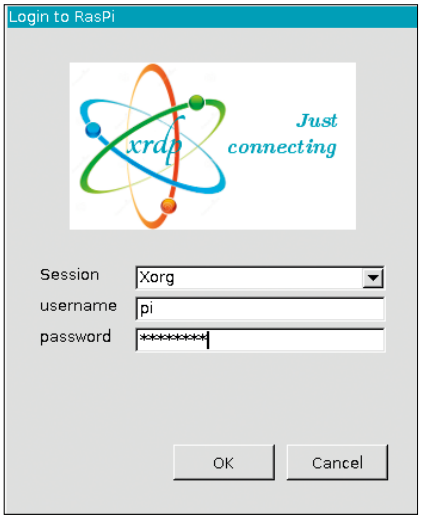


Abbildung 6.3: Login-Fenster zum Raspberry Pi

Hier gibt man die Login-Informationen ein (dieselben wie per SSH, d. h. Standardname pi- Standard-Passwort raspberry). Anschließend erscheint der Desktop des Raspberry in einem eigenen Fenster.