



Nr. 177/Juni 2023 € 5,00

ISSN 1022-1611

# news

CLUBCOMPUTER · DIGITAL SOCIETY

**CLUB**EDU

**KI in der Schule**

**CLUB**SYSTEM

**Cloud Computing**

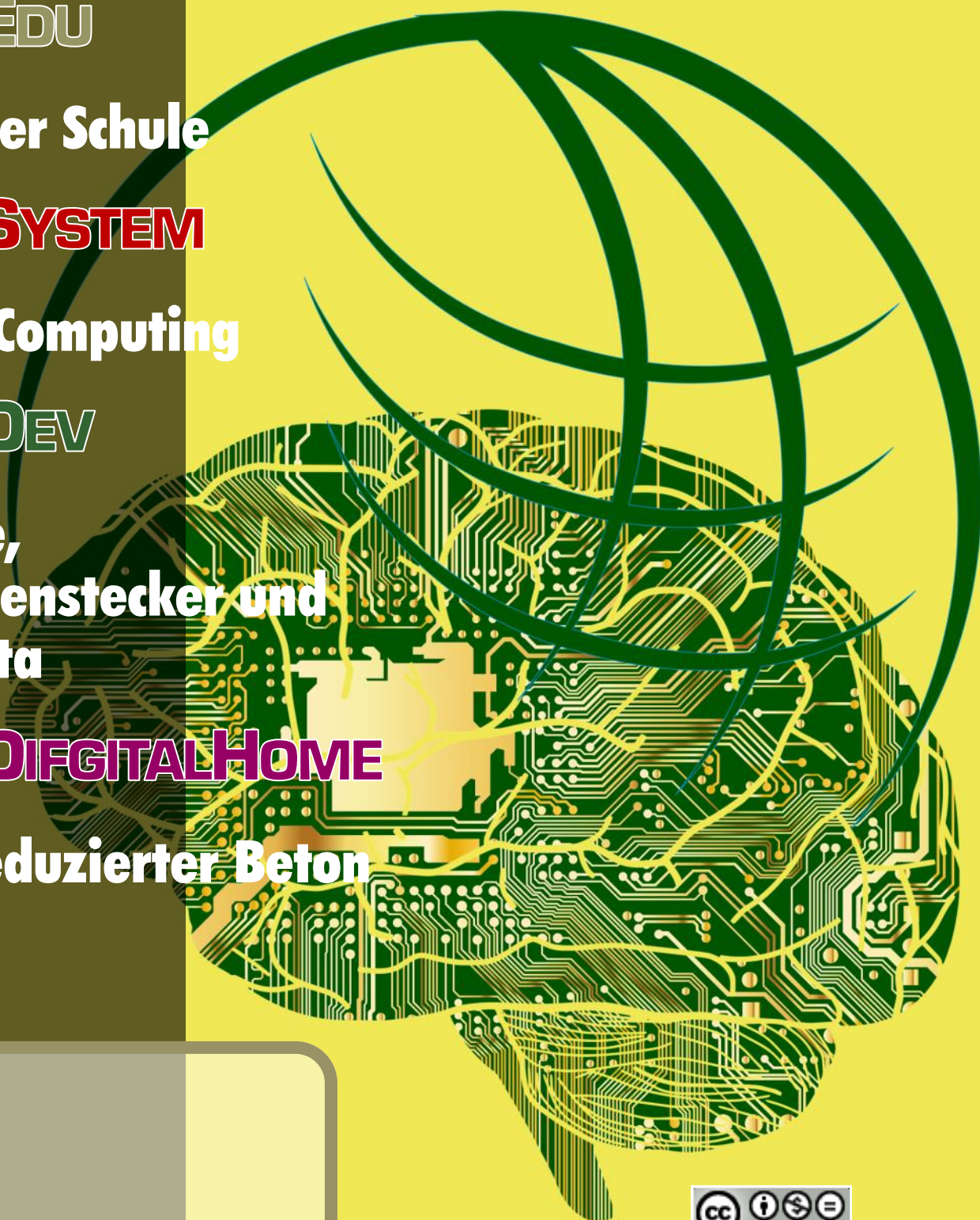
**CLUB**DEV

**ZigBee,  
Zwischenstecker und  
Tasmota**

**CLUB**DIGITALHOME

**CO2-reduzierter Beton**

Österreichische Post MZ 16Z040679 M ClubComputer, Siccardsburggasse 4/1/22 1100 Wien





# Services

<http://buero.clubcomputer.at?svc=xx|yyy>



Diese Adresse zeigt alle Aspekte einer Mitgliedschaft bei ClubComputer. Online sind alle Inhalte menügeführt. Das Kürzel ist wichtig für den Verweis auf eine konkrete Seite.

Wer lieber ein gedrucktes Dokument liest, kann ein solches über den Druck-Button rechts oben herstellen. Über den Menü-Button kann man das Menü ausblenden, über den Link-Button kann man über einen QR-Code die Seite am Handy anzeigen lassen. Über kann man im Verlauf der bereits besuchten Seiten blättern.

In der PDF-Version dieser Ausgabe führen die Links direkt zu der betreffenden Seite.

## Verein

[cc|clubcomputer](#) · [cc|finanzen](#) · [cc|history](#) · [cc|hotline](#) · [cc|konto](#) · [cc|mitglieder](#) · [cc|support](#) · [cc|vorstand](#)

## Öffentlich

[at|wissen](#) · [cc|allapps](#) · [cc|exweb](#) · [cc|inhalte](#) · [cc|newsletter](#) · [cc|wapps](#) · [pc|123](#) · [pc|pdf](#)

## Persönlich

[at|asp](#) · [cc|billing](#) · [at|domain](#) · [at|drive](#) · [at|ftp](#) · [at|mail](#) · [at|panel](#) · [at|php](#) · [at|press](#) · [at|server](#)

## Extern

[at|facebook](#) · [at|status](#) · [cc|facebook](#) · [cc|medien](#) · [cc|youtube](#) · [ds|facebook](#) · [ds|medien](#) · [ds|youtube](#)

## Druck

[cc|folder](#) · [cc|pp](#) · [cc|visit](#) · [ds|folder](#) · [pc|news](#)

## Partner

[at|cccat](#) · [at|htl3r](#) · [cc|adim](#) · [cc|jix](#) · [cc|kultur](#) · [cc|mcca](#) · [cc|metro](#) · [cc|techbold](#) · [cc|tgm](#) · [ds|digisoci](#) · [etv](#) · [pc|mtm](#) · [pc|pcnews](#) · [pc|ultraprint](#)

## Wir

[cc|calendar](#) · [cc|heuriger](#) · [cc|meating](#) · [cc|weihnac](#) · [ht](#) · [ds|digital](#)

## Du

[cc|card](#) · [cc|clubid](#) · [cc|mitmachen](#) · [cc|webfree](#) · [cc|welcome](#)

## Hilfe

[cc|statuten](#) · [xx|hilfe](#) · [xx|links](#) · [xx|pages](#) · [xx|sitema](#) · [p](#) · [xx|standorte](#)

# Inhalt

## LIESMICH

1



### Cover

Franz Fiala

Globus von [hikmat-nabiye](#)

Gehirn von [GDI](#)  
beides auf Pixabay  
soll den weltweiten  
Zugriff auf KI symbolisieren.

2

Liebe Leser, Inhalt  
Franz Fiala

2

Impressum, Autoren, Inserenten,  
Services

## CLUBSYSTEM

3

KI in der Schule  
Christian Dörninger

## CLUBSYSTEM

21

Cloud Computing  
Christian Zahler

20

Inhalt  
Christian Zahler

## CLUBDEV

15

ZigBee, Zwischenstecker und Tasmota  
Martin Weissenböck

23

## CLUBDEV

CO2-reduzierter Beton  
Günter Hartl

# Liebe Leser!

Franz Fiala

Wir bitten alle Leser, unseren wöchentlichen Newsletter zu abonnieren. In diesem Newsletter werden neu erschienene Artikel und Veranstaltungen angekündigt:  
<https://clubcomputer.at/newsletter-signup/>

## Einladung zum Stadtheurigen

am Dienstag, 4. Juli 2023 um 17:00

Gasthaus Alt Wien  
Koflergasse 26, 1120 Wien

Öffi: 62 und BadnerBahn  
Station Marx-Meidlinger-Straße

<https://www.gasthaus-alt-wien.at/>

Bitte um Anmeldungen über Homepage

*Franz Fiala*



# Autoren

Dörninger Christian MR Dipl.-Ing. Dr. 1954 3



Sonderbeauftragter für pädagogische Reformprojekte

Firma bmbwf

Absolvent TU-Wien, Technische Physik

Interessen Informatik, Didaktik, Curriculumentwicklung

[cdorninger@it4education.at](mailto:cdorninger@it4education.at)  
<https://www.it4education.at/>

Fiala Franz Dipl.-Ing. 1948 1,2



Vizepräsident von ClubComputer, Leitung der Redaktion und des Verlags der PCNEWS, Lehrer für Nachrichtentechnik und Elektronik i.R.

Werdegang Arsenal-Research, TGM Elektronik

Absolvent TU-Wien, Nachrichtentechnik

[franz.fiala@clubcomputer.at](mailto:franz.fiala@clubcomputer.at)  
<http://ffiala.cc/>

Hartl Günter Ing. 1963 23



Wirtschaftsingenieur, Systemadministrator für Windows Clients und Linux Server in Logistikcenter

Hobbies Krav Maga, Windsurfen, Lesen

[ghartl3@gmail.com](mailto:ghartl3@gmail.com)

Weissenböck Martin Dir.Dr. 1950 15



Leiter der ADIM und Autor von ADIM-Skripten, Leiter des Vereins "SCHULInfoSMS, Univ.-Lektor an der TU Wien, Direktor der HTL Wien 3 Rennweg i.R.

[martin@weissenboeck.at](mailto:martin@weissenboeck.at)  
<http://www.weissenboeck.at/>

Zahler Christian Ing. Mag. 1968 20,21



Erwachsenenbildung, MCSE, Lehrer für Elektro- und Automatisierungstechnik, Technische Mechanik und Informatik am Francisco-Josephinum Wieselburg

Firma HBLFA Francisco-Josephinum; WIFI

Absolvent TU-Wien

[office@zahler.at](mailto:office@zahler.at)  
<http://www.zahler.at/>



# Inserenten

techbold 24



Dresdner Straße 89 1200 Wien

+43 1 34 34 333

[office@techbold.at](mailto:office@techbold.at)

<http://www.techbold.at>

Produkte Reparatur, Aufrüstung, Softwareinstallation, Datenrettung. Installation und Wartung von IT-Anlagen.

# Impressum

## Impressum, Offenlegung

Richtung Auf Anwendungen im Unterricht bezogene Informationen über Personal Computer Systeme. Berichte über Veranstaltungen des Herausgebers.

Erscheint 4 mal pro Jahr: Mär, Jun, Sep, Nov

ISSN 1022-1611

Herausgeber und Verleger ClubComputer

Siccardsburggasse 4/1/22 1100 Wien

01-6009933-11 FAX: -12

[buero@clubcomputer.at](mailto:buero@clubcomputer.at)

<https://clubcomputer.at/>

ZVR: 085514499

IBAN: AT74 1400 0177 1081 2896

Mitgliedsbeitrag 2019: 46,- Euro

Konto: AT74 1400 0177 1081 2896

oder PayPal [office@clubcomputer.at](mailto:office@clubcomputer.at)

Digital Society

Graben 17/10 1010 Wien

01-314 22 33

[info@DigiSociety.at](mailto:info@DigiSociety.at)

<https://digisociety.at/>

ZVR: 547238411

IBAN: AT45 3266 7000 0001 9315

Druck Ultra Print

Pluhová 49, SK-82103 Bratislava

<http://www.ultraprint.eu/>

Versand 162040679 M

PDF-Version <http://d.pcnews.at/pdf/in177.pdf>



Namensnennung, nicht kommerziell, keine Bearbeitungen  
<http://creativecommons.org/licenses/by-nc-nd/4.0/>



# Künstliche Intelligenz für die Schule

**Christian Dorninger**

Das Thema „Künstliche Intelligenz“ (englisch: „Artificial Intelligence“) hat nun im Jahr 2023 die „pädagogischen Marktplätze“ erreicht. Mit dem Produkt Chat-GPT ist die Nutzung eines „Schreibgenerators“ für vielfältige Zwecke plötzlich aktuell geworden. Mit dem textbasierten Dialogsystem lassen sich täuschend echte Texte für den Sprachunterricht oder auch realistische Fächer (Mathematik, Naturwissenschaften, Informatik) verfassen, wobei die eigenen Fähigkeiten und die eigene sprachliche Kreativität der Studierenden und Schüler\*innen kaum mehr eingeschätzt werden kann. „Hausarbeiten“ mit originärem Ursprung werden damit fraglich und ein wesentliches Moment, einen persönlichen Schreibstil zu entwickeln, kann damit massiv gestört werden. Auf der anderen Seite erscheint es dann doch geraten, diese informatischen Entwicklungen so in ein Unterrichts- oder Studiengeschehen einzubeziehen, dass man Aufgabenstellungen findet, die jenseits von einfachen Schreibtexten die Fähigkeiten von Schüler\*innen und Studierenden mit Unterstützung von Textgeneratoren bewerten kann.

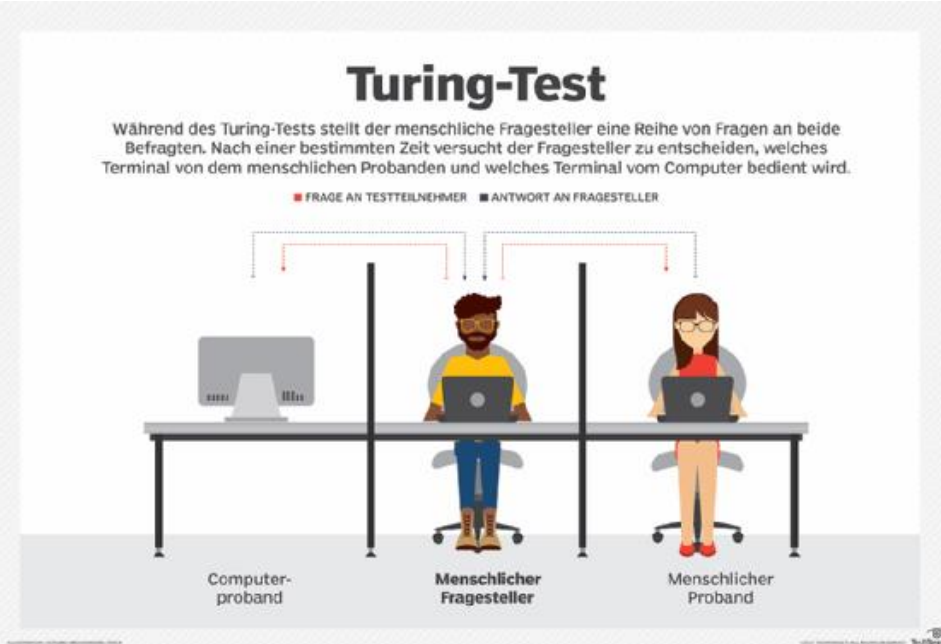
## 1. Der Turingtest

Alan Turing formulierte 1950 die Idee, wie man feststellen könne, ob ein Computer ein dem Menschen gleichwertiges Denkvermögen besitzt. Der „Test“ war eine theoretische Skizze, die im Zeitalter der Informatik und der künstlichen Intelligenz genauer und konkreter ausformuliert wurde (Dartmouth Conference 1956).

Ein menschlicher Fragesteller (Mitte) führt mittels Tastatur und Bildschirm mit zwei unbekannten Gesprächspartnern eine Unterhaltung. Der eine Gesprächspartner ist ein Mensch (Frau rechts), der andere eine Maschine (links). Wenn der Fragesteller nach einer intensiven Befragung nicht klar sagen kann, welcher von beiden die Maschine ist, hat die Maschine den Turingtest bestanden. Ihr wird ein dem Menschen ebenbürtiges Denkvermögen unterstellt.

Turing vermutete, dass es bis zum Jahr 2000 möglich sein werde, Computer so zu programmieren, dass der durchschnittliche Anbieter eine höchsten 70% Chance habe, Mensch und Maschine erfolgreich zu identifizieren, nachdem er 5 Minuten mit ihnen „Gesprochen“ hat. Die Vorhersage hat sich bisher nicht erfüllt – ein Beleg für die Unterschätzung der Komplexität künstlicher Intelligenz!

Die Überlegungen zum Turingtest waren allerdings Basis für den ersten Chat-Bot,



ELISA von J. Weizenbaum, der ein Grundrepertoire von freundlichen Fragestellungen hatte („wie geht es Dir?“) und auf Personen „eingegangen“ ist. Immerhin war etlichen Versuchspersonen nicht bewusst, dass es sich um einen nichtmenschlichen Gesprächspartner handelte. Es bestand den Turingtest allerdings nicht!

## 2. Künstliche Intelligenz wird in drei Entwicklungsstufen gesehen

Als „schwache KI“ konzentriert man sich auf die Lösung konkreter Anwendungsprobleme aus Mathematik und Informatik. Intelligente Service am Smartphone wie Routenplaner oder Sprachübersetzer, also erfolgt eine gezielte Beschäftigung mit einer Aufgabe. Auch Schachcomputer und „Alpha-Go“ gehören in diese Kategorie.

Als „starke KI“ würde ein System angesehen, das so „intelligent“ wie ein Mensch ist, also sich mit unterschiedlichen Aufgaben beschäftigt, Vorgangsweisen planen kann oder komplexe Zusammenhänge verstehen kann.

Er selbst ist der chinesischen Sprache nicht mächtig und versteht somit weder den Sinn der einzelnen Zeichen noch den Sinn der Geschichte. Danach erhält er

noch einen Zettel mit Fragen zu der Geschichte (ebenfalls in chinesischer Notation). Der Mensch findet des Weiteren einen Stapel chinesischer Skripte und ein „Handbuch“ mit Regeln in seiner Muttersprache vor. Die Skripte enthalten Zusatz- und Hintergrundinformationen zu den Geschichten (eine Art „Wissensdatenbank“). Das Handbuch ermöglicht es ihm, die Zeichen mit der Geschichte in Verbindung zu bringen, allerdings ausschließlich auf der Ebene der Zeichenerkennung (über die Form der Zeichen). Auch entnimmt er dem Handbuch Anweisungen, welche Zeichen er (abhängig von den Zeichen der Geschichte, der Skripte und der Fragen) auf den Antwortzettel zu übertragen hat. Er folgt also rein mechanischen Anweisungen und schiebt das Ergebnis (die „Antworten“ auf die Fragen) durch den Türschlitz, ohne die Geschichte oder die Fragen verstanden zu haben.

Vor der Tür wartet ein chinesischer Muttersprachler, der die Antwortzettel liest. Er kommt aufgrund des Sinngehaltes der Antworten zu dem Ergebnis, im Raum befindet sich ebenfalls ein Chinesisch sprechender Mensch, der die Geschichte verstanden hat.



Das Chinesische Zimmer ist der Name für ein Gedankenexperiment des Philosophen John Searle, der damit die „Starke KI“ angreift. Mit seiner Hilfe versucht Searle die Meinung zu widerlegen, dass digitale Computer allein dadurch Bewusstsein erlangen könnten, dass sie ein passendes Programm ausführen. Searle beschrieb einen geschlossenen Raum, in dem sich ein Mensch befindet. Ihm werden durch einen Schlitz in der Tür Zettel mit Geschichten auf Chinesisch zugestellt.

Als „Superintelligenz“ wäre eine Maschine zu verstehen, die in jedem kognitiven Feld besser als die begabtesten Menschen ist und Lösungsstrategien „lernen“ und damit weiterentwickeln kann. Da die Maschine aber keine ethischen Maßstäbe entwickeln kann, wäre es hier die Aufgabe, einen „Kern-Code“ zu schreiben.

### 3. Geschichtliche Entwicklung

Eine kurze geschichtliche Entwicklung der Künstlichen Intelligenz in Schlagworten

1936: Turingmaschine: Der britische Mathematiker Alan Turing beweist durch seine Theorien, dass eine Rechenmaschine – eine sogenannte „Turingmaschine“ – in der Lage wäre, kognitive Prozesse auszuführen, sofern diese sich in mehrere Einzelschritte zerlegen und durch einen Algorithmus darstellen lassen. Damit legt er den Grundstein für das, was wir heute unter Künstlicher Intelligenz verstehen.

1956: Der Begriff „KI“ entsteht: Im Sommer 1956 treffen sich Wissenschaftler zu einer Konferenz am Dartmouth College im US-Bundesstaat New Hampshire. Sie sind der Ansicht, dass Aspekte des Lernens sowie andere Merkmale der menschlichen Intelligenz von Maschinen simuliert werden können. Der Programmierer John McCarthy schlägt dafür den Begriff „Künstliche Intelligenz“ vor. Während der Konferenz wird mit dem „Logic Theorist“ – der es schafft, mehrere Dutzend mathematische Lehrsätze zu beweisen – zudem das erste KI-Programm der Welt geschrieben.

1966: Geburt des ersten Chatbots: Der deutsch-amerikanische Informatiker Joseph Weizenbaum vom Massachusetts Institute of Technology erfindet ein Computerprogramm, das mit Menschen kommuniziert. Über Skripte simuliert „ELIZA“ verschiedene Gesprächspartner, beispielsweise einen Psychotherapeuten. Weizenbaum ist überrascht, mit welch einfachen Mitteln „ELIZA“ die Illusion eines menschlichen Gesprächspartners erzeugen kann.

1972: KI gelangt in die Medizin: Mit „MYCIN“ findet Künstliche Intelligenz den Weg in die Praxis: Das von Ted Shortliffe an der Stanford University entwickelte Expertensystem wird zur Behandlung von Krankheiten eingesetzt. Expertensysteme sind Computerprogramme, die das Wissen eines bestimmten Fachgebietes durch Formeln, Regeln und eine Wissensdatenbank bündeln. In der Medizin dienen sie zur Unterstützung bei Diagnose und Therapie.

1986: „NETalk“ spricht: Der Computer erhält erstmals eine Stimme. Durch die Eingabe von Beispielsätzen und Phonemketten bringen Terrence J. Sejnowski und Charles Rosenberg ihrem Programm „NETalk“ das Sprechen bei. „NETalk“ kann Wörter lesen und korrekt aussprechen sowie das Gelernte auf ihm unbe-

kannte Wörter anwenden. Damit ist es eines der frühen künstlichen neuronalen Netze – also Programme die mit großen Datensätzen gefüttert werden und darauf aufbauend eigene Schlüsse ziehen können. In Aufbau und Funktion ähneln sie damit dem menschlichen Gehirn.

1997: Computer schlägt Schachweltmeister: Die KI-Schachmaschine „Deep Blue“ der Firma IBM bezwingt den amtierenden Schachweltmeister Garry Kasparov in einem Turnier. Dies gilt als historischer Erfolg der Maschinen in einem Bereich, der bislang vom Menschen dominiert wurde. Kritiker werfen jedoch ein, dass „Deep Blue“ nicht durch kognitive Intelligenz, sondern nur durch das Berechnen aller denkbaren Züge gewonnen habe.

Um 2000: Maschinelles Lernen und neuronale Netze: Expertensysteme und andere auf Wissensdatenbanken basierende Systeme hatten nur mäßigen Erfolg, da es sich als zu schwer herausstellte, das benötigte Wissen von Hand in formale Regeln zu überführen. Diese Schwäche wird durch maschinelles Lernen umgangen. Hierbei lernt das Computersystem selbstständig anhand der vorliegenden Daten und ist so auch in der Lage, verborgene Zusammenhänge zu erkennen, die ein Mensch nicht berücksichtigt hätte. Klassische Verfahren lernen dabei eine Ausgabefunktion anhand vorher extrahierter Merkmale, die durch manuelle Programmierung aus den Eingabedaten extrahiert wurden. Hierbei zeigte sich jedoch ein ähnliches Problem wie bei den Expertensystemen, dass eine manuelle Auswahl nicht immer zu einem optimalen Ergebnis führt. Eine aktuell erfolgreiche Struktur für maschinelles Lernen sind künstliche neuronale Netze (KNNs). Sie basieren auf der Fähigkeiten, die erforderlichen Merkmale selbst anhand der Rohdaten zu lernen, beispielsweise direkt aus den Kamerabil-

2011: KI erreicht den Alltag: Technologiesprünge bei der Hard- und Software bahnen Künstlicher Intelligenz den Weg in das tägliche Leben. Leistungsstarke Prozessoren und Grafikkarten in Computern, Smartphones und Tablets ermöglichen es normalen Verbrauchern auf KI-Programme zuzugreifen. Insbesondere Sprachassistenten erfreuen sich großer Beliebtheit: Apples „Siri“ kommt 2011 auf den Markt, 2014 stellt Microsoft die Software „Cortana“ vor und Amazon präsentiert 2015 Amazon Echo mit dem Sprachdienst „Alexa“.

2011: KI „Watson“ gewinnt Quizshow: Das Computerprogramm „Watson“ tritt in Form eines animierten Bildschirmsymbols in einer US-amerikanischen TV-Quizshow an und gewinnt gegen die menschlichen Mitspieler. Damit beweist „Watson“, dass es die natürliche Sprache versteht und schnell auf schwierige Fragen antworten kann.

2016: Alpha Go besiegt besten Go-Spieler: Im März 2016 besiegte das System Alpha-Go mit dem Südkoreaner Lee Sedol einen der weltbesten Go-Spieler. Das vom Unternehmen DeepMind entwickelte Programm hatte zuvor Millionen von archivierte Spielen mit Deep Learning ausgewertet und zudem mehrere Millionen Mal gegen sich selbst gespielt.

2017 – KI Programm besiegt Profi-Computerspieler: Eine künstliche Intelligenz der Firma OpenAI besiegt bei einem mit 24 Millionen Dollar dotierten Turnier des Computerspiels Dota 2 einige der weltbesten Profispieler auf diesem Gebiet. Dota 2 gilt als eines der komplexesten Videospiele überhaupt, komplexer als Go oder Schach. Dota 2 wurde allerdings hier im „Eins zu eins“-Modus gespielt und nicht im komplexeren Team-Modus. OpenAI erklärte, dass die KI nur vier Monate benötigte, um diese Spielstärke zu erreichen. Die KI wurde trainiert, indem diese immer wieder gegen sich selbst antrat. Die KI bekam das gleiche Sichtfeld wie der menschliche Spieler und durfte nur eine begrenzte Anzahl von Aktionen gleichzeitig ausführen. Ziel von OpenAI ist es nun, eine KI zu entwickeln, die die besten menschlichen Spieler auch im Team-Modus besiegen kann.

2018: KI debattiert über Raumfahrt und vereinbart einen Friseurtermin: Diese beiden Beispiele demonstrieren die Leistungsfähigkeit von Künstlichen Intelligenzen: Im Juni liefert sich „Project Debater“ von IBM mit zwei Debattiermeistern ein Rededuell über komplexe Themen – und schneidet dabei beachtlich ab. Wenige Woche zuvor demonstriert Google auf einer Konferenz, wie die KI „Duplex“ beim Friseur anruft und im Plauderton einen Termin vereinbart – ohne dass die Dame am anderen Ende der Leitung merkt, dass sie mit einer Maschine spricht.

2022: Chat-GPT macht Furore: Das textbasierte Dialogsystem Chat-GPT ist als Pilotprodukt bald in aller Munde: es kann sprachlich korrekte Dialoge führen und besitzt viele Fertigkeiten im kognitiven Bereich (Mathematik, Naturwissenschaften, etc.). es hält nach zwei Monaten Einzug an Hochschulen und Schulen.

20xx: Die nahe Zukunft: Trotz jahrzehntelanger Forschung steht die Entwicklung der Künstlichen Intelligenz noch relativ am Anfang. Damit sie in sensiblen Bereichen wie dem automatisierten Fahren oder der Medizin eingesetzt werden kann, muss sie zuverlässiger und sicherer gegen Manipulationen werden. KI-Systeme sollen zudem lernen, ihre Entscheidungen zu erklären, damit Menschen sie nachvollziehen und die Denkweise der KI besser erforschen können.

#### 4. Wo wird künstliche Intelligenz heute schon angewendet?

Eine allgemeine künstliche Intelligenz gibt es weiterhin nicht, doch für immer mehr spezielle Aufgaben werden KI-Lösungen entwickelt, die dem Menschen ebenbürtig oder überlegen sind. Dabei geht es häufig um das Erkennen und Interpretieren von Mustern in Bildern, Sprache oder Datenbanken, das Finden optimaler Lösungen in komplexen Situationen oder das Filtern relevanter Informationen.

Oft empfinden wir diese Technologien gar nicht mehr als intelligent, sobald wir uns an sie gewöhnt haben – der bekannte „KI-Effekt“.

##### Virtuelle Assistenten und Chatbots

Virtuelle Assistenten entsprechen wohl am ehesten der Vorstellung von KI, wie wir sie aus Science-Fiction-Filmen kennen: Siri, Alexa, Cortana oder Google Assistant: Die freundliche Stimme lässt sich per Sprache steuern, kann ein Kinoticket kaufen, Musik abspielen oder das Licht dimmen. Dahinter stecken beeindruckende Leistungen in der Spracherkennung (die Geräusche müssen in Wörter übersetzt werden) und der Sprachverarbeitung (in der Folge von Wörtern muss eine konkrete Anforderung erkannt werden). Chatbots, als einfachere, textbasierte Variante, sind seit Jahren vor allem im Kundenservice vieler Unternehmen im Einsatz und können die Mitarbeiter bei der großen Masse an Routine-Anfragen entlasten.

Beispiel: Google Assistant – ein virtueller persönlicher Assistent auf Basis von KI

##### Robotik

In der physischen Steuerung von Robotern stößt KI weiterhin an ihre Grenzen, vor allem in einer offenen unerwarteten Umgebung. In kontrollierten Umgebungen sieht das anders aus: Industrieroboter übernehmen zum Beispiel einen Großteil der Fertigung von Autos bei VW. Oder ein Beispiel aus der Logistik: Mit künstlicher Intelligenz kann zum Beispiel Amazon Millionen von Warenhaus-Robotern kollisionsfrei steuern.

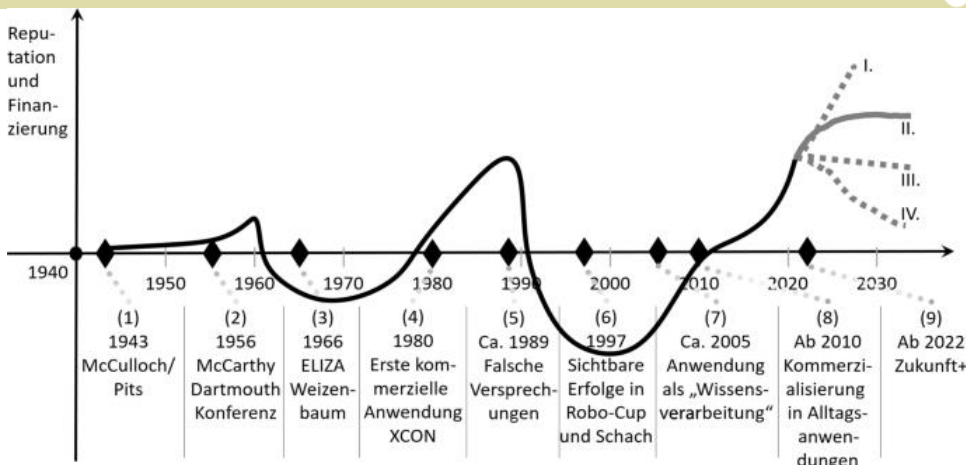
Und in virtuellen Umgebungen wie Schach, Go, oder Computerspielen hat der Computer dem Menschen bereits den Rang abgelaufen. AlphaGo im Duell gegen Lee Sedol. Quelle: [www.bbc.com/news/technology-35785875](http://www.bbc.com/news/technology-35785875)

##### Suchmaschinen

Eine weitere Spielart der KI, an die wir uns längst gewöhnt haben, sind Suchmaschinen. Nur sie ermöglichen uns, in den riesigen Datenmengen des World Wide Web zu navigieren und die für uns relevanten Inhalte zu finden.

##### Empfehlungsdienste

Den gleichen Vorteil – die relevanten Inhalte aus einem überwältigenden Angebot



zu finden, bieten uns Empfehlungssysteme, zum Beispiel in E-Shops wie Amazon oder Dienste wie Spotify oder Netflix. Sie steuern aber auch die Neuigkeiten und Gruppen, die uns in sozialen Medien vorgeschlagen werden oder werden im Recruiting von Firmen verwendet. Netflix: Ein Produkt, das ohne intelligente Empfehlungen nicht denkbar ist.

##### Content-Moderation

In sozialen Medien oder Nachrichtenforen ist die Moderation eine große Herausforderung. Trolle, Beleidigungen und verbotene Inhalte zu erkennen und zu löschen ist oft mit menschlicher Arbeitskraft nicht mehr zu bewältigen. Youtube beispielsweise setzt bei der Moderation stark auf künstliche Intelligenz: 76% der gelöschten Videos auf Youtube wurden von künstlicher Intelligenz identifiziert und markiert.

##### Gesichtserkennung

Die Gesichtserkennung hat große Fortschritte gemacht. Die meisten aktuellen Smartphones können heute mittels Gesichtserkennung entsperrt werden. Ebenfalls mittels Gesichtserkennung erkennt Facebook Photo Review, wer auf neu hochgeladenen Fotos abgebildet ist und fragt die Person nach ihrem Einverständnis. Digitalkameras lernen, automatisch auf Gesichter zu fokussieren oder bewegende Gegenstände wie Autos zu verfolgen und scharf zu stellen.

Die gleiche technologischen Grundlage wird auch in der Bekämpfung der Corona-Pandemie verwendet: In vielen britischen Städten oder in den Einrichtungen von Amazon wird so mittels der Bilder von existierenden Überwachungskameras die Einhaltung der Sicherheitsabstände registriert.

#### 5. Regelbasierte Systeme - Logiksysteme

Die Programme für künstliche Intelligenz werden hier auf der Basis mathematischer Logik ausgeführt. Die Beschäftigung mit künstlicher Intelligenz erfordert eine Befassung mit logischen Kalkülen wie der Aussage- und der Prädikatenlogik. Ein kurzer Abriss anbei:

##### 5.1. Aussagen

Klassische Logik: z.B. Es gibt unendliche viele natürliche Zahlen  $n \in \mathbb{N}$ ,  $a, b$  und  $c$ , so dass die Gleichung  $a^2 + b^2 = c^2$  (Teil des großen Fermatschen Satzes!). Ist eine Aussage. Aussagen können „wahr“ (w) oder „falsch“ (f) sein. Aussagen kann man mit „Wahrheitstabellen“ darstellen.

Die **Negation**  $\neg A$  ist beispielsweise „falsch“, wenn A „wahr“ ist. Daher zeigt die Tabelle

A	$\neg A$
w	f
f	w

Die **Konjunktion**  $A \wedge B$  („A und B“) ist dann wahr, wenn beide Aussagen „wahr“ sind. Die **Disjunktion**  $A \vee B$  („A oder B“) ist dann wahr, wenn eine der Aussagen falsch ist. in Tabellenform:

Bei der **Implikation**  $A \Rightarrow B$  ist die Aussage falsch, wenn A wahr und B falsch ist. Bei der **Äquivalenz**  $A \Leftrightarrow B$  ist die Aussage wahr, wenn A und B beide wahr oder beide falsch sind.

A	B	$A \wedge B$
w	w	w
w	f	f
f	w	f
f	f	f

A	B	$A \vee B$
w	w	w
w	f	w
f	w	w
f	f	f



Bei der **Implikation**  $A \Rightarrow B$  ist die Aussage falsch, wenn A wahr und B falsch ist. Bei der **Äquivalenz**  $A \Leftrightarrow B$  ist die Aussage wahr, wenn A und B beide wahr oder beide falsch sind.

A	B	$A \Rightarrow B$
w	w	w
w	f	f
f	w	w
f	f	w

A	B	$A \Leftrightarrow B$
w	w	w
w	f	f
f	w	f
f	f	w

Aus diesen „kleinen Wahrheitstabellen“ kann man nun eine größere logische Aussage überprüfen. Dazu wird die Wahrheitstabelle erweitert. Z.B. die Wahrheitstabellen von  $A \Leftrightarrow B$  und  $(A \Rightarrow B) \wedge (B \Rightarrow A)$  sind gleich.

A	B	$A \Leftrightarrow B$
w	w	w
w	f	f
f	w	f
f	f	w

A	B	$A \Rightarrow B$	$B \Rightarrow A$	$(A \Rightarrow B) \wedge (B \Rightarrow A)$
w	w	w	w	w
w	f	f	w	f
f	w	w	f	f
f	f	w	w	w

Die Symbole  $\neg$ ,  $\wedge$  und  $\vee$  werden als **Quantoren** bezeichnet.

## 5.2. Logik

Spezielle Sprachen, die Denkmuster der Logik unterstützen, sind die Aussagenlogik (siehe Kap. I) und die Prädikatenlogik. Die Prädikatenlogik ist eine künstliche Sprache, ähnlich einer Programmiersprache, die keine Mehrdeutigkeiten zulässt. Wort-

schatz und Grammatik sind minimal. Die Logik hat viele Bezüge zur Informatik: Die Verbindung der Aussagenlogik zur Digitaltechnik, die Logik und Komplexitätstheorie, die Verbindung von Prädikatenlogik und relationale Datenbanken und die Logik als Programmiersprache (z.B. Prolog).

## 5.3. Aussagenlogik

**Syntax** (=korrekte Schreibweise): Aussagenlogische Formeln werden induktiv in vier Schritten definiert.  $\Sigma$  wird dabei die Menge der Atome genannt.

1. Jedes Atom in  $\Sigma$  ist eine Formel.
2. Ist  $\alpha$  eine Formel, so ist  $(\neg \alpha)$  eine Formel.
3. Für Formeln  $\alpha$  und  $\beta$  sind auch  $(\alpha \wedge \beta)$ ,  $(\alpha \vee \beta)$ ,  $(\alpha \rightarrow \beta)$  und  $(\alpha \leftrightarrow \beta)$  Formeln.
4. Nur Ausdrücke, die mit den ersten drei Prämissen gebildet werden, sind Formeln.

Es gelten folgende Bindungsregeln:  $\neg$  bindet am stärksten, gefolgt von  $\wedge$ ,  $\vee$ ,  $\rightarrow$  und  $\leftrightarrow$ .

**Semantik** (= Bedeutung der aussagelogischen Formeln): Sei J eine „Bewertung“ der Atome, die mit  $\{0, 1\}$  erfolgen kann. Die Bewertung von Formeln wird erweitert, wenn folgende Regeln gelten:

1.  $J(\neg \alpha) = 1$ , wenn  $J(\alpha) = 0$ .
2.  $J(\alpha \wedge \beta) = 1$ , wenn  $J(\alpha) = 1$  und  $J(\beta) = 1$ .
3.  $J(\alpha \vee \beta) = 1$ , wenn  $J(\alpha) = 1$  oder  $J(\beta) = 1$ .
4.  $J(\alpha \rightarrow \beta) = 1$ , wenn  $J(\alpha) = 0$  oder  $J(\beta) = 1$ .
5.  $J(\alpha \leftrightarrow \beta) = 1$ , wenn  $J(\alpha) = J(\beta)$ .

Wenn alle möglichen Bewertungen J ( $\alpha$ ) der Atome Formel zusammen mit dem zugehörigen Wahrheitswert „wahr“ oder „falsch“ in eine Tabelle geschrieben werden, erhält man die Wahrheitstafel (siehe Kap. Aussagen).

**Beispiel:** Sei  $\alpha = (A \vee \neg B) \wedge (A \vee B \rightarrow C)$ , dann ist die Wahrheitstafel (ähnlich oben):

A	B	C	$A \vee \neg B$	$A \vee B$	$C \rightarrow \alpha$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Wahrheitstafel der obigen Aussage:

Es gilt:

- Eine Formel  $\alpha$  ist **erfüllbar**, wenn es wenigstens eine Bewertung J ( $\alpha$ ) = 1 gibt.
- Eine Formel heißt **tautologisch**, wenn für jedes J ( $\alpha$ ) = 1 gilt.
- Eine Formel heißt **widerspruchsvoll**, wenn für jeden Formel J( $\alpha$ ) = 0 ist.

- Eine Formel heißt **falsifizierbar**, wenn es eine Bewertung J ( $\alpha$ ) = 0 gibt.

Wenn  $\alpha$  und  $\beta$  aussagelogische Formeln sind. Dann ist  $\beta$  eine **semantische Folgerung** aus  $\alpha$ , wenn für jede Bewertung J, für die  $\alpha$  wahr ist, auch  $\beta$  wahr ist. Sei  $\alpha = (A \vee B) \wedge (\neg A \vee B)$  und  $\beta = B$ , dann ist  $\alpha = (1, 0, 1, 0)$  – bitte nachprüfen! und  $\beta = (1, 0, 1, 0)$ . Somit ist  $\beta$  eine semantische Schlussfolgerung aus  $\alpha$  und man schreibt  $\alpha \models \beta$ .

Zwei aussagelogische Formeln  $\alpha$  und  $\beta$  heißen **logisch äquivalent**, wenn  $J(\alpha) = J(\beta)$  für alle Bewertungen gilt. Die Schreibweise für die logische Äquivalenz ist  $\alpha \approx \beta$ .

In der Logik gibt es Regeln, welche semantische Folgerungen einfacher machen:

A. Die Vererbungsregeln: Sind  $\alpha$ ,  $\beta$  und  $\gamma$  aussagelogische Formeln. Dann gelten

1. Wenn  $\alpha \approx \beta$ , so gilt auch  $\neg \alpha \approx \neg \beta$ .
2. Wenn  $\alpha \approx \beta$ , so gilt  $\gamma \wedge \alpha \approx \gamma \wedge \beta$ ,
3. Wenn  $\alpha \approx \beta$ , so gilt  $\gamma \vee \alpha \approx \gamma \vee \beta$ .

B. Die Junktoren-Minimierung: Sind  $\alpha$  und  $\beta$  aussagelogische Formeln. Dann gilt:

1.  $\alpha \rightarrow \beta \approx \neg \alpha \vee \beta$ .
2.  $\alpha \wedge \beta \approx \neg(\neg \alpha \vee \neg \beta)$
3.  $\alpha \leftrightarrow \beta \approx (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha) \approx (\neg \alpha \vee \beta) \wedge (\neg \beta \vee \alpha)$ .

Man zeige die Formeln über Wahrheitstafeln.

C. Die Äquivalenzregeln: Sind  $\alpha$ ,  $\beta$  und  $\gamma$  aussagelogische Formeln. Dann gelten

1.  $\neg \neg \alpha \approx \alpha$  (Negationsregel).
2.  $\alpha \vee \alpha \approx \alpha$  bzw.  $\alpha \wedge \alpha \approx \alpha$  (Idempotenzregeln)
3.  $\alpha \wedge \beta \approx \beta \wedge \alpha$  bzw.  $\alpha \vee \beta \approx \beta \vee \alpha$  (Kommutativgesetz)
4.  $(\alpha \vee \beta) \vee \gamma \approx \alpha \vee (\beta \vee \gamma)$  und  $(\alpha \wedge \beta) \wedge \gamma \approx \alpha \wedge (\beta \wedge \gamma)$  (Assoziativgesetz)
5.  $(\alpha \wedge \beta) \vee \gamma \approx (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$  und  $(\alpha \vee \beta) \wedge \gamma \approx (\alpha \wedge \gamma) \vee (\beta \wedge \gamma)$  (Distributivgesetz)
6.  $\neg(\alpha \wedge \beta) \approx \neg \alpha \vee \neg \beta$  und  $\neg(\alpha \vee \beta) \approx \neg \alpha \wedge \neg \beta$  (Regeln von de Morgan).

Mit all diesen Regeln lassen sich Normalformen von aussagelogischen Formeln finden, die in der Logik eine wichtige Rolle spielen.

## a) Die Negationsnormalform

Sei  $\alpha$  eine aussagelogische Formel ohne  $\rightarrow$  und  $\leftrightarrow$ . Dann ist  $\alpha$  eine Negationsnormalform (NNF), wenn jedes Negationszeichen direkt vor einem Atom steht.

**Beispiel:** Man berechne eine NNF von  $\neg((A \vee B) \wedge (\neg C \vee D))$ .

Es gilt:  $\neg((A \vee B) \wedge (\neg C \vee D))$   
 $\approx \neg(A \vee B) \vee \neg(\neg C \vee D)$  De Morgan  
 $\approx (\neg A \wedge \neg B) \vee \neg(\neg C \vee D)$  De Morgan  
 $\approx (\neg A \wedge \neg B) \vee (\neg \neg C \wedge \neg D)$  de Morgan  
 $\approx (\neg A \wedge \neg B) \vee (C \wedge \neg D)$  Negationsregel

Die letzte Form ist eine Negationsnormalform.

### b) Konjunktive Normalform

Eine Formel der Form  $\alpha = \bigvee \alpha_i$  ( $i = 1$  bis  $n$ ) wird Klausel genannt. Eine Formel  $\alpha$  heißt konjunktive Normalform (KNF), wenn sie eine Konjunktion mehrerer Klauseln ist.

**Beispiel:** Man berechne eine KNF von  $\alpha = \neg(A \vee \neg B \vee \neg(\neg C \vee D) \vee \neg A)$ .

$\neg(A \vee \neg B \vee \neg(\neg C \vee D) \vee \neg A)$   
 $\approx \neg A \wedge (B \wedge (\neg C \vee D) \wedge A)$   
 Negationsnormalform  
 $\approx (\neg A \vee B) \wedge (\neg A \vee (\neg C \vee D) \wedge A)$   
 Distributivgesetz  
 $\approx (\neg A \vee B) \wedge (\neg A \vee (\neg C \vee D) \wedge (\neg A \vee A))$   
 Distributivgesetz  
 $\approx (\neg A \vee B) \wedge (\neg A \vee \neg C \vee D) \wedge 1$   
 Klammern

Die letzte Form ist die konjunktive Normalform, eine Konjunktion von Klauseln.

### c) Disjunktive Normalform

Eine Formel der Form  $\alpha = \bigwedge \alpha_i$  ( $i = 1$  bis  $n$ ) wird Monom genannt. Eine Formel  $\alpha$  heißt disjunktive Normalform (DNF), wenn sie eine Disjunktion mehrerer Monome ist.

**Beispiel:** Man berechne die DNF von  $\alpha = \neg(A \vee \neg B \vee \neg(\neg C \vee D) \vee \neg A)$ .

$\neg(A \vee \neg B \vee \neg(\neg C \vee D) \vee \neg A)$   
 $\approx \neg A \vee (B \wedge (\neg C \vee D) \wedge A)$   
 Negationsnormalform  
 $\approx \neg A \vee ((B \wedge \neg C) \vee (B \wedge D) \wedge A)$   
 Distributivgesetz  
 $\approx \neg A \vee ((B \wedge \neg C) \wedge A) \vee ((B \wedge D) \wedge A)$   
 Distributivgesetz  
 $\approx \neg A \vee ((B \wedge \neg C) \wedge A) \vee (B \wedge D \wedge A)$   
 Klammern

A	B	C	Z	Minterme
0	0	0	1	$\bar{A} \bar{B} \bar{C}$
0	0	1	0	
0	1	0	1	$\bar{A} B \bar{C}$
0	1	1	1	$\bar{A} B C$
1	0	0	0	
1	0	1	1	$A \bar{B} C$
1	1	0	0	
1	1	1	0	

$$Z = (\bar{A} \wedge \bar{B} \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge C) \vee (A \wedge \bar{B} \wedge C)$$

es gilt:  $(\bar{A} \bar{B} \bar{C}) = (\bar{A} \bar{B} \bar{C}) \vee (\bar{A} \bar{B} \bar{C})$

Z damit rechts erweitern

$$Z = \bar{A} \bar{B} \bar{C} \vee \bar{A} \bar{B} \bar{C} \vee \bar{A} \bar{B} \bar{C} \vee \bar{A} \bar{B} \bar{C} \vee \bar{A} \bar{B} \bar{C}$$

Umformen durch geeignetes Ausklammern

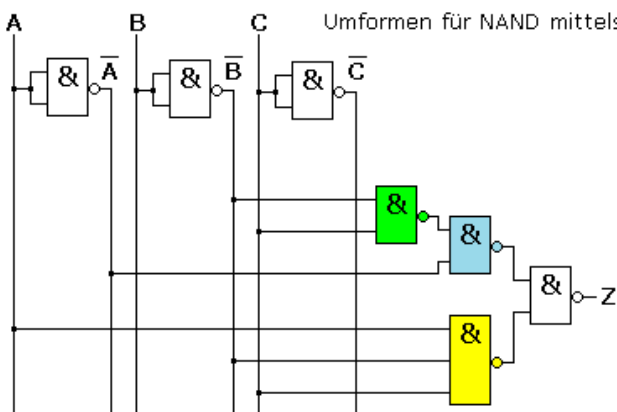
$$Z = \bar{A} \bar{C} \wedge (\bar{B} \vee B) \vee \bar{A} \bar{B} \wedge (\bar{C} \vee C) \vee \bar{A} \bar{B} \bar{C}$$

$$Z = \bar{A} \bar{C} \wedge 1 \vee \bar{A} \bar{B} \wedge 1 \vee \bar{A} \bar{B} \bar{C}$$

nochmals Ausklammern

$$Z = \bar{A} \wedge (B \vee \bar{C}) \vee \bar{A} \bar{B} \bar{C}$$

Umformen für NAND mittels doppelter Negierung



$$Z = \bar{\bar{A}} \wedge (B \vee \bar{C}) \vee \bar{A} \bar{B} \bar{C}$$

$$Z = \bar{A} \wedge (B \vee \bar{C}) \wedge \bar{A} \bar{B} \bar{C}$$

$$Z = \bar{A} \wedge (B \vee \bar{C}) \wedge \bar{A} \bar{B} \bar{C}$$

$$Z = \bar{A} \wedge (B \wedge \bar{\bar{C}}) \wedge \bar{A} \bar{B} \bar{C}$$

mit  $\bar{\bar{C}} = C$

$$Z = \bar{A} \wedge (B \wedge C) \wedge \bar{A} \bar{B} \bar{C}$$

## Ein Algorithmus zur schnellen Berechnung einer konjunktiven Normalform

**Eingabe:** Aussagelogische Formel  $\alpha$

**Verfahren:** sei  $\alpha$  von der Form  $H_1 \wedge \dots \wedge H_n$ , wobei  $H_i$  keine Konjunktionen sind

**While** ein  $H_i$  Tiefe  $\Rightarrow 4$  besitzt **do**

**for**  $i \in \{1, \dots, n\}$  **do**

**if**  $H_i$  hat Tiefe  $\Rightarrow 4$  **then**

seien  $G_1, \dots, G_m$  alle Subformeln von  $H_i$  in Tiefe 3, die Tiefe  $\Rightarrow 1$  haben

ersetze  $H_i$  wie folgt:  $H_i [G_1, \dots, G_m] \rightarrow (G_1 \Leftrightarrow X_1) \wedge \dots \wedge (G_m \Leftrightarrow X_m) \wedge H_i [X_1, \dots, X_m]$

**end if**

**end for**

Iteriere mit der entstandenen Formel als neues  $\alpha$

**end while**

Wende die KNF-Erstellung auf die entstandene Formel an

Die letzte Form ist die disjunktive Normalform, eine Disjunktion der Monome.

Logische Formeln werden direkt in digitalen Netzwerken realisiert. Dabei spielen Normalformen beim Schaltungsaufbau eine wichtige Rolle.

Die flexibelsten „Gatter“ (Logikbausteine) heißen NOR und NAND, also  $\neg \vee$  und  $\neg \wedge$ .

Das Bild zeigt die Umformung einer komplexen Schaltung Z aus drei Eingängen A, B und C und Umformung zu einer „schlanken“ Schaltung:

$$Z = \neg(\neg A \wedge \neg(\neg B \wedge C) \wedge \neg(A \neg B C))$$

Mit Kalkülen der Aussagenlogik kann man formale Beweise führen. Dazu braucht es Prämissen  $\alpha_1, \alpha_2, \alpha_3, \alpha_n$  oder Voraussetzungen und die Formel  $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \dots \wedge \alpha_n \rightarrow \beta$  heißt gültiges Argument, wenn sie eine Tautologie ist d.h. für alle  $\alpha_i$  ist  $J(\alpha_i) = 1$ .

**Beispiel:** Die Formel  $\alpha \wedge (\alpha \rightarrow \beta) \rightarrow \beta$

ist ein gültiges Argument. Sie ist von der Bauarbeit  $\alpha_1 \wedge \alpha_2 \rightarrow \gamma$  und  $\gamma = \beta$ .

Mit  $\alpha = (0,0,1,1)$  und  $\beta = (0,1,0,1)$  ergibt  $\alpha \rightarrow \beta$   $(1,1,0,1)$  und  $\alpha \wedge (\alpha \rightarrow \beta)$  ergibt  $(0,0,0,1)$  und damit ergibt  $\alpha \wedge (\alpha \rightarrow \beta) \rightarrow \beta$   $(1,1,1,1)$ .

Die Formel ist tautologisch, also ein gültiges Argument.

Auch hier gelten die Äquivalenzregeln (siehe oben) und auch entsprechende Schlussregeln:

Zu den Begriffen in den beiden folgenden Tabellen:

„Modus ponens“: Die Implikation enthält die Wahrheit.

„Modus tollens“: Verlangt, dass  $\alpha \rightarrow \beta$  und  $\neg \beta$  wahr sind; dann ist auch  $\neg \alpha$  wahr.

Konjunktion: Wenn  $\alpha$  und  $\beta$  wahr, dann ist auch  $\alpha \wedge \beta$  wahr.

Die „Ausdehnung“ sagt, dass man zu jeder wahren Prämisse  $\alpha$  jede beliebige Formel  $\beta$  hinzunehmen darf, um mit  $\alpha \vee \beta$  eine wahre Aussage zu bekommen.

**Beispiel:** Formel  $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \gamma)$  soll bewiesen werden: Zuerst wird die eine Implikation weggebracht:  $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \gamma) \wedge \alpha \rightarrow \gamma$ :

1. Prämisse:  $\alpha \rightarrow \beta$

2. Prämisse:  $\beta \rightarrow \gamma$

3. Prämisse:  $\alpha$

4. 1,3 imp:  $\beta$

5. 2,4 imp:  $\gamma$

## 5.4. Prädikatenlogik

Prädikate sind Eigenschaften von Objekten eines Universums. Man benötigt ein Vokabular, um Aussagen und Sätze zu formulieren.

**Syntax** (Schreibweise): Es gibt Terme  $t_1, \dots, t_n$ , die aus Variablen  $V$ , Konstanten  $K$  und Funktionssymbolen  $F$  bestehen und in der Form  $f(t_1, t_2, \dots, t_n)$  aufgebaut sind.  $P(t_1, t_2, \dots, t_n)$  heißt  $n$ -stelliges Prädikat. Prädikate werden durch Junktoren  $\neg, \wedge, \vee, \rightarrow$  und  $\leftrightarrow$  und Quantoren „Allquantor“  $\forall$  und „Existenzquantor“  $\exists$  zu prädikatenlogischen Formeln verknüpft.

**Beispiele:**  $\forall x (P(x) \wedge Q(x)) \wedge \exists y (S(y) \wedge \forall z R(x, z))$  oder  $\forall x (x = f(x) \vee \neg P(x, x))$  mit Termen  $x$  und  $f(x)$ .

Das Vorkommen einer Variable  $x$  heißt frei, wenn es nicht im Wirkungsbereich des Quantors liegt. Das Vorkommen von  $x$  heißt gebunden, wenn es im Wirkungsbereich des Quantors liegt.

**Beispiel:**  $\forall x (P(x) \vee Q(x)) \wedge \exists y (S(y) \wedge \forall z R(x, z))$ ;  $\forall$  wirkt auf  $x$ ,  $P$  und  $Q$ ,  $\exists$  auf  $y$ ,  $\forall$  auf  $z$ ;  $x$  in  $R(x, z)$  ist frei.

Solange ein Variablenname  $x$  sowohl frei als auch gebunden vorkommt, wähle man einen neuen Variablennamen  $z$  und ersetze alle freien  $x$  durch  $z$  (Konsistente Umbenennung!).

**Semantik** (Bedeutung): Eine Signatur besteht aus einem Tripel  $\Sigma = (K, F, R)$ , wobei  $K$  die Menge der Konstantensymbole,  $F$  die Menge der Funktionssymbole und  $R$  die Menge der Prädikatensymbole ist. Wenn  $\alpha$  eine Formel ist, ist  $\Sigma(\alpha)$  eine induzierte Signatur. Auf dieser Basis kann man für eine Formel  $\alpha$  den Wahrheitsgehalt bestimmen.

**Beispiel:**  $\alpha = \forall x \exists y P(x, y) \wedge \forall x \forall y (x = y \rightarrow P(x, f(y))) \wedge \neg P(a, a)$  Wie bestimmt sich der Wahrheitswert?

Mit  $U = \{3, 4\}$ ,  $J(P) = P = \{(3, 4), (4, 3)\}$ ,  $J(a) = a = 3$  und  $J(f) = f$  mit  $f(3) = 4$  und  $f(4) = 3$  ergibt sich

$\forall x \exists y P(3, 4) \wedge \forall x \forall y (x = y \rightarrow P(3, f(4))) \wedge \neg P(3, 3)$ ; die erste Teilformel  $J(y, y_u)P(3, y) = 1$  und  $J(y, y_u)P(4, y)$  ergeben 1, die zweite Teilformel  $J(\forall x \forall y (x = y \rightarrow P(x, f(y)))) = 1$  und  $J(\neg P(3, 3)) = 1$ .  $\alpha$  ist „wahr“.

Die Äquivalenzregeln der Prädikatenlogik sind:

- $\neg(\exists x \alpha) \approx \forall x (\neg \alpha)$  und  $\neg(\forall x \alpha) \approx \exists x (\neg \alpha)$  Quantorenwechsel
- $\forall x \forall y \alpha \approx \forall y \forall x \alpha$  und  $\exists x \exists y \alpha \approx \exists y \exists x \alpha$  Quantorentausch
- $\exists x \alpha \vee \exists x \beta \approx \exists x (\alpha \vee \beta)$  und Quantorenzusammenfassung
- Ist  $x$  keine freie Variable in  $\alpha$ , so gilt  $\exists x \alpha \approx \alpha$  und  $\forall x \alpha \approx \alpha$  Quantorelimination
- Ist  $x$  keine freie Variable in  $\beta$ , so gilt

Formel	äquivalent zu	Name der Regel
$\alpha \vee \beta, \alpha \wedge \beta$	$\beta \vee \alpha, \beta \wedge \alpha$	Kommutativgesetz /com
$(\alpha \vee \beta) \vee \gamma, \dots$	$\alpha \vee (\beta \vee \gamma), \dots$	Assoziativgesetz /ass
$\neg(\alpha \wedge \beta), \dots$	$\neg \alpha \vee \neg \beta, \dots$	De Morgan
$\alpha \rightarrow \beta$	$\neg \alpha \vee \beta$	Implikation /imp
$\alpha$	$\neg \neg \alpha$	Doppelte Negation /dn

Aussage	Ableitbar	Name der Regel
$\alpha$ und $\alpha \rightarrow \beta$	$\beta$	Modus ponens /mp
$\alpha \rightarrow \beta$ und $\neg \beta$	$\neg \alpha$	Modus Tollens /mt
$\alpha$ und $\beta$	$\alpha \wedge \beta$	Konjunktion /con
$\alpha \wedge \beta$	$\alpha, \beta$	Vereinfachung /simp
$\alpha$	$\alpha \vee \beta$	Ausdehnung /add

5.1  $\neg(P(x) \rightarrow \forall y Q(y)) \vee R(z) \wedge x \alpha \wedge \beta \approx \exists x (\alpha \wedge \beta)$  und  $\exists x \alpha \vee \beta \approx \exists x (\alpha \vee \beta)$  oder

5.2  $\forall x \alpha \wedge \beta \approx \forall x (\alpha \wedge \beta)$  und  $\forall x \alpha \vee \beta \approx \forall x (\alpha \vee \beta)$  Quantifizierung

Dazu kommt noch die konsistente Umbenennung.

Es gibt zwei Normalformen:

a) Die **Negationsnormalform** (auch hier steht jedes Negationszeichen direkt vor der Primformel)

**Beispiel:**  $\neg(P(x) \rightarrow \forall y Q(y)) \vee R(z)$  umwandeln in eine Negationsnormalform

$\neg(P(x) \rightarrow \forall y Q(y)) \vee R(z)$   
 $\approx \neg(\neg P(x) \vee \forall y Q(y)) \vee R(z)$   
 Ersetzen von  $\rightarrow$

$\approx \neg \neg P(x) \vee \neg \forall y Q(y) \vee R(z)$  de Morgan

$\approx P(x) \vee \neg \forall y Q(y) \vee R(z)$  Doppelte Negation weg

$P(x) \vee \exists y \neg Q(y) \vee R(z)$  Quantorentausch (= Negationsnormalform)

b) **Pränexe Normalform:**  $\alpha$  ist in einer pränexen Normalform, wenn sie die Form  $Q_1 x_1, \dots, Q_n x_n \beta$  hat, wobei  $\beta$  keine Quantoren enthält. Die Quantorenfolge  $Q_1 x_1, \dots, Q_n x_n$  wird als Präfix und  $\beta$  als der Kern der Formel bezeichnet.

**Beispiel:** Man transformiere die Formel  $\exists z ((\neg \exists x (P(x, z) \vee \forall y Q(x, f(y)))) \vee \forall y P(g(z, y), z))$  in eine pränexe Normalform, wobei der Kern eine konjunktive Normalform ist.

Konsistente Umbenennung:  $\exists z ((\neg \exists x (P(x, z) \vee \forall y Q(x, f(y)))) \vee \forall w P(g(z, w), z))$

Es gilt:  $\approx \exists z ((\forall x \neg P(x, z) \vee \forall y Q(x, f(y))) \vee \forall w P(g(z, w), z))$  Quantorenwechsel

$\approx \exists z ((\forall x \neg P(x, z) \vee \forall y Q(x, f(y))) \vee \forall w P(g(z, w), z))$  de Morgan

$\approx \exists z (\forall x ((\neg P(x, z) \wedge \exists y \neg Q(x, f(y))) \vee \forall w P(g(z, w), z)))$  2 mal Quantorenwechsel

$\approx \exists z (\forall x (\exists y (\forall w (\neg P(x, z) \wedge \neg Q(x, f(y))) \vee P(g(z, w), z))))$  3 mal Quantifizierung

$\approx \exists z \forall x \exists y \forall w (\neg P(x, z) \wedge \neg Q(x, f(y))) \vee P(g(z, w), z)$  Klammern (Kern fett!).

Typische Anwendungsbereiche der logikbasierten KI sind Mustererkennung (Fingerabdruck, Iriserkennung, Gesichtserkennung), Sprachverarbeitung (Language Parsing) oder eine semantische Analyse. Mit mathematischer Logik werden Expertensysteme betrieben (MYCIN war eines der ersten unterstützenden Diagnosewerkzeuge), es ist auch Grundlage von Verfahren zum maschinellen Lernen (siehe unten), es erlaubt regelbasiertes Schlussfolgern oder Modellieren und ist die Basis für logik- und funktionale Programmierung (z.B. mit PROLOG) und ist die Basis für Programme, die sich selbst abwandeln können.

Auch in der Wissensrepräsentation zur Darstellung von Fakten und Zusammenhängen werden logische Kalküle eingesetzt. (Hierarchische) Taxonomien werden über logische Darstellungen repräsentiert.

**Beispiel Kontoverzinsung:** Daten (2, %) wird in der Syntax als (2%) geschrieben, bedeutet im Kontext eine Verzinsung von 2% auf dem Konto und die Wissensrepräsentation ergibt bei € 100 nach einem Jahr 102 € am Konto.

## 6. Suchalgorithmen und Entscheidungsstrategien

Eine zweite Klasse von Verfahren der KI sind Suchtechniken, die dann auch zu Entscheidungsstrategien führen. Es gibt eine größere Anzahl von Möglichkeiten, etwa Netzwerke mit vielen Knoten und Überhängen. Nun geht es darum, den



„optimalen“ Pfad durch das Netzwerk zu finden.

Einfache Suchalgorithmen vernachlässigen die spezielle Natur des jeweiligen Problems. Deshalb können sie allgemeiner und abstrakter implementiert werden, wodurch dieselbe Implementierung für eine große Auswahl von Problemen verwendet werden kann. Der Nachteil einfacher Suchalgorithmen sind die entstehenden Kosten: Der Suchraum von Suchproblemen ist im Allgemeinen sehr groß, einfaches Suchen läuft jedoch nur in kleinen Suchräumen in annehmbarer Zeit ab.

### Suche in Listen

Algorithmen zur Suche in Listen sind die einfachsten Suchalgorithmen überhaupt. Das Ziel der Suche in Listen ist es, ein bestimmtes Element einer Liste zu finden, von dem der zugehörige Suchschlüssel bekannt ist. Da dieses Problem in der Informatik oft anzutreffen ist, sind die verwendeten Algorithmen – sowie deren Komplexität – sehr gut untersucht.



Abbildung: Suche in einer Skip-Liste

Der einfachste Suchalgorithmus für Listen ist die lineare Suche. Bei ihr wird solange ein Element nach dem anderen durchlaufen, bis ein Element mit dem gesuchten Schlüssel angetroffen wird. Die lineare Suche hat eine Laufzeit von  $O(n)$  ( $n$  ist die Anzahl der Elemente der Liste) und kann sowohl auf sortierte als auch unsortierte Listen angewendet werden. Ein fortgeschrittenes Verfahren ist die binäre Suche mit einer Laufzeit von  $O(\log n)$ . Für große Listen ist sie viel effizienter als die lineare Suche, setzt jedoch voraus, dass die Liste vorher sortiert wurde und ein wahlfreier Zugriff auf die Elemente möglich ist. Die Interpolationssuche, auch Intervallsuche genannt, ist eine Verbesserung der binären Suche, die eine Gleichverteilung der Daten voraussetzt. Die Laufzeit ( $\log(\log n)$ ) ist nur für sehr große Datenmengen besser als die der binären Suche. Ein weiterer Suchalgorithmus für Listen ist der Grover-Algorithmus, der auf Quantencomputern zum Einsatz kommt und quadratisch schneller als die klassische lineare Suche für unsortierte Listen abläuft. Für die Suche in Listen kann auch Hashing (ist eine Abbildung, die eine große Eingabemenge auf eine kleine Zielmenge abbildet) verwendet werden, das im Durchschnitt eine konstante Zeit  $O(1)$ , im schlechtesten Fall jedoch lineare Zeit benötigt.

### Suche in Bäumen

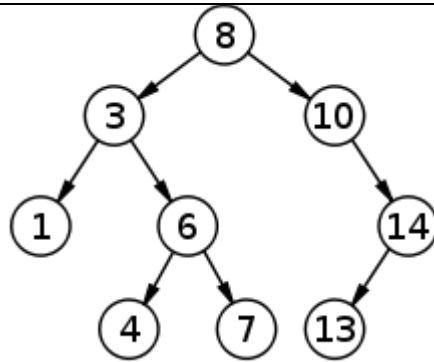


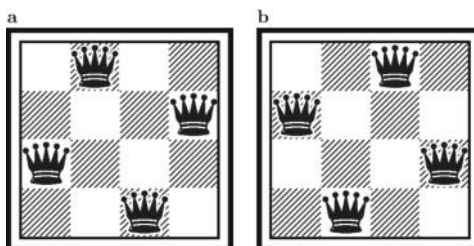
Abbildung: Ein binärer Suchbaum der Größe neun und der Tiefe drei

Die Suche in Bäumen ist die Königsdisziplin unter den Suchalgorithmen. Sie durchsucht Knoten von Bäumen, unabhängig davon, ob der Baum explizit oder implizit (während der Suche generiert) ist. Dabei wird folgendes Prinzip angewendet: Ein Knoten wird aus einer Datenstruktur entnommen. Seine Kindknoten werden untersucht und gegebenenfalls der Datenstruktur hinzugefügt. Je nach Auswahl der Datenstruktur kann der Baum in verschiedenen Reihenfolgen durchsucht werden. Die Verwendung einer Warteschlange führt so zu einer Breitensuche, bei der der Baum Ebene für Ebene durchlaufen wird. Bei Verwendung eines Stacks hingegen wird jeweils bis zu einem Blatt gesucht und erst anschließend mit dem nächsten Kindknoten fortgefahren. Dies wird als Tiefensuche bezeichnet.

### Suche in Graphen

Viele Probleme der Graphentheorie können mit Hilfe von Suchalgorithmen gelöst werden. Beispiele für diese Probleme sind das Problem des Handlungsreisenden, die Berechnung kürzester Pfade und die Konstruktion eines minimalen Spannbaums. Die entsprechenden Algorithmen sind zum Beispiel Kruskals Algorithmus, Dijkstras Algorithmus oder Prim's Algorithmus, die als Erweiterungen der Algorithmen für die Suche in Bäumen gesehen werden können.

Suchalgorithmen können auf Spielsituationen angewandt werden: 4 Damen-Problem: Die Schachdamen bedrohen sich wenn sie auf horizontaler, vertikaler oder diagonaler Linie stehen. Nun soll keine Dame die andere bedrohen. Zwei mögliche Lösungen:



Ein möglicher Ablauf der Suche wäre:

- Dame A1
- Dame A2; Bedrohung
- Dame A2 nach B2; Bedrohung
- Dame B2 nach C2
- Dame A3; Bedrohung
- Dame A3 nach B3; Bedrohung
- Dame A3 auf Reihe 3, Dame C2 auf D2... usw.

Nun können etliche Algorithmen helfen, einen Suchprozess in einer Liste oder auf einem Graphen (siehe oben) voranzutreiben:

### Algorithmus zur Tiefensuche

**Datenstrukturen:** L ist eine Liste von Knoten

**Eingabe:** Füge die initialen Knoten in die Liste L ein

#### Algorithmus:

1. Wenn L leer ist, dann breche ab
2. Wähle ersten Knoten K aus Liste L, R ist die Restliste
3. Wenn K ein Zielknoten ist, dann gebe aus: Zielknoten und Weg dorthin (Weg im Graphen)
4. Wenn K kein Zielknoten ist, dann sei N (K) die (geordnete) liste der direkten Nachfolger von K, mit dem Weg dorthin markiert

$L = N(K) ++ R$  (wobei ++ Listen zusammenhängt!)

Mache weiter mit 1.

### Algorithmus zur Breitensuche

**Datenstrukturen:** L ist eine Menge von Knoten, markiert mit dem dorthin führenden Weg

**Eingabe:** Füge den initialen Knoten in die Menge L ein

#### Algorithmus:

1. Wenn L leer, dann breche ab
2. Wenn L einen Zielknoten K enthält, dann gebe aus: K und Weg dorthin
3. Sonst sei N (L) die Menge aller direkten Nachfolger der Knoten von L, mit dem Weg dorthin

Mache weiter mit Schritt 1 und  $L = N(L)$ .

### Rückwärtssuche

Man geht von einem bekannten Zielknoten aus und versucht, den Anfangsknoten zu erreichen

Voraussetzung ist die explizite Angabe des Zielknotens und man kann von jedem Knoten die direkten Vorgänger berechnen

Eine Rückwärtssuche ist besser als eine Vorwärtssuche, wenn die Verzweigungsra-

te in Rückwärtsrichtung kleiner ist als die Verzweigungsrate in Vorwärtsrichtung.

Man kann auch die normale Suche in die Rückwärtssuche kombinieren.

### Evolutionäre (genetische) Algorithmen

Genetische bzw. evolutionäre Algorithmen orientieren sich an der Evolution von Lebewesen. Daher gibt es eine

- Anfangspopulation d.h. eine Menge von Individuen (Zustände, Objekte), die üblicherweise als Bitstring dargestellt werden. Ein Bit oder einen Teilfolge von Bits entspricht dabei einem Gen.
- eine Bewertungsfunktion, die Fitnessfunktion genannt wird.

Gesucht ist ein optimaler Zustand.

Daher werden Zustände als Bitfolgen kodiert

Die Bewertung entspricht der Fitness der Bitfolge

Höhere Fitness bedeutet mehr Nachkommen

Man beobachtet die Entwicklung einer Menge von Bitfolgen (Population)

Nachkommen werden durch zufällige Prozesse unter Verwendung von Operationen analog zur Mutation und Crossover (Rekombination) erzeugt

An Operatoren wird verwendet: Selektion, Rekombination (Austausch), Mutation ( $1 > 0$ ,  $0 \rightarrow 1$ ), Aussterben, Ende der Evolution

### Algorithmus genetische Suche

**Eingabe:** Anfangspopulation, Fitnessfunktion  $\Omega$ , Populationsgröße  $K$

**Datenstrukturen:**  $S, S'$ : Mengen von Individuen

### Algorithmus

$S$  = Anfangspopulation;

while do:  
( $K$  enthält ein Individuum mit maximalem  $\Omega$ )

$S' = \emptyset$

for  $i = 1$  to  $K$  do

wähle zufällig (mit  $\Omega$  gewichtet)  
zwei Individuen  $A$  und  $B$  aus  $S$ ;

Erzeuge Nachkommen  $C$  aus  $A$  und  $B$   
durch Rekombination;

$C' = \text{Mutation von } C$ ;

$S' = S' \cup \{C'\}$ ;

end for

$S = S'$

end while

gebe Individuum mit maximaler Fitness aus



### Beispiel: n-Damen mit evolutionärem Algorithmus

**Individuen:** Folge von  $n$  Zahlen, wobei die  $i$ -te Zahl die Position (Spalte) der Dame in der Zeile  $i$  angibt.

**Fitness:** Anzahl der Damenpaare, die sich nicht gegenseitig bedrohen, also ( $n$  über  $2$ )  $= n \cdot (n-1) / 2$

Operationen: Mutation ändert den Eintrag in der Folge auf beliebigen Wert zwischen 1 und  $n$ .

Da man aber schon weiß, dass die Zustände mit doppelten Zahlen in der Folge nicht gültig sind, lässt man nur Permutationen von 1 bis  $n$  als Individuen zu.

### 4 Damen am Spielfeld

Population  $\{3, 2, 1, 4\}$

Bewertung:  $\Omega(3, 2, 1, 4) = 2$

Mutation:  $[3, 2, 1, 4] \rightarrow [4, 2, 1, 3]$   $[1, 2, 3, 4]$

Bewertung:  $\Omega(4, 2, 1, 3) = 3$ ;  $\Omega(1, 2, 3, 4) = 0$

Mutation:  $[4, 2, 1, 3] \rightarrow [2, 4, 1, 3]$  .....

Bewertung:  $\Omega(2, 4, 1, 3) = 4$  ....

Kodierung  $[2, 4, 1, 3]$  ist eine Lösung!

Typische Anwendungsbereiche für Suchstrategien sind eben „Strategiespiele“ wie Schach oder Go, oder Verfahren in Computernetzwerken, um einen Pfad, der minimale Ressourcen verbraucht, zu finden.

Nun erfolgt der Schritt von den Suchverfahren zu **heuristischen Bewertungsfunktionen** für eine KI-Aufgabe: z.B. Schach: Ein Schachmeister wird befragt und es werden Bewertungsfunktionen aufgesetzt:

$B(s) = a_1 \cdot \text{Material} + a_2 \cdot \text{Bauernstruktur} + a_3 \cdot \text{Königssicherheit} +$

$a_4 \cdot \text{Springer\_im\_Zentrum} + a_5 \cdot \text{Läufer\_Diagonalabdeckung} + \dots$   
(Material meint die Expertise von „gelungenen Zügen“).

Die Gewichtungen  $a_i$  werden mit Experten abgesprochen! Die optimale Bewertungsfunktion muss gefunden werden – durch regelmäßiges Üben.

Die Experten werden nur nach relevanten Eigenschaften  $f_1(s), \dots, f_n(s)$  gefragt und die Bewertungsfunktion wird zu  $B(f_1(s), \dots, f_n(s))$ . Mit einer zufälligen Belegung wird gestartet und im Lauf der Spiele wird die Bewertungsfunktion laufend verändert.

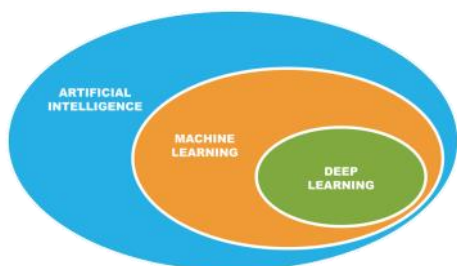
Das zentrale Problem dabei ist nicht die Bewertung einzelner Züge, sondern vom gesamten Spiel - Lernen durch Verstärkung (siehe unten).

### 7. Maschinelles Lernen

Maschinelles 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. Dazu bauen Algorithmen beim maschinellen Lernen ein statistisches Modell auf, das auf Trainingsdaten beruht und welches gegen die Testdaten getestet wird. Das heißt, es werden nicht einfach die Beispiele auswendig gelernt, sondern Muster und Gesetzmäßigkeiten in den Lerndaten erkannt. So kann das System auch unbekannte Daten beurteilen (Lerntransfer) oder aber am Lernen unbekannter Daten scheitern (Überanpassung; englisch overfitting). Aus dem weiten Spektrum möglicher Anwendungen seien hier genannt: automatisierte Diagnoseverfahren, Erkennung von Kreditkartenbetrug, Aktienmarktanalysen, Klassifikation von Nukleotidsequenzen, Sprach- und Texterkennung sowie autonome Systeme.

Dabei wird auf den Begriff „Agent“ zurückgegriffen, der Information verarbeitet und aus einer Eingabe eine Ausgabe produziert. Ein Softwareagent berechnet aus einer Benutzereingabe ein Ergebnis. Agenten brauchen immer eine genaue Anleitung, ein Programm, um diese Ausgaben zu erzeugen. Ein Agent wäre dann „intelligent“, wenn er sich das Programm durch „Metaanweisungen“ (auf höhere Ebene) selber schaffen könnte. Da aber die direkte Programmierung eines intelligenten Agenten nicht möglich ist, muss man das Fernziel versuchen, anders zu erreichen: Durch einen Lernprozess, also maschinelles Lernen.

Beim maschinellen Lernen spielen Art und Mächtigkeit der Wissensrepräsentation eine wichtige Rolle. Man unterscheidet zwischen symbolischen Ansätzen, in denen das Wissen – sowohl die Beispiele als auch die induzierten Regeln – explizit repräsentiert ist, und nicht-symbolischen Ansätzen, wie neuronalen Netzen, denen zwar ein berechenbares Verhalten „antrainiert“ wird, die jedoch keinen Einblick in die erlernten Lösungswege erlauben; hier ist Wissen implizit repräsentiert.



<https://dataflog.com/read/deep-learning-methods-machine-learning/>

Ein Untergebiet von maschinellem Lernen sind neuronale Netze (siehe nächstes Kapitel). Diese Lernalgorithmen sind von Nervenzellenverbindungen im menschlichen Gehirn inspiriert. Das Gehirn verarbeitet Informationen über Neuronen und Synapsen. Analog dazu bestehen künstliche neuronale Netze aus mehreren Reihen von Datenknoten, die mit gewichteten Verbindungen untereinander vernetzt sind.

### Typische Anwendungen

- *Erweiterung und Anpassung des Lexikons eines linguistischen Systems durch automatische Verarbeitung von geschriebenen Sätzen, wobei der Inhalt der Sätze gleichzeitig automatisch erfasst wird*
- *Adaptierung von gewichteten Bewertungungsverfahren in einem Zweipersonenspiel, wobei man abhängig von Gewinn/Verlust gewichte verändert.*
- *Lernen einer Klassifikation durch Vorgabe von Trainingsbeispielen, die als positiv/negativ klassifiziert werden*

Maschinelles Lernen kann automatisiert Wissen generieren, Algorithmen trainie-

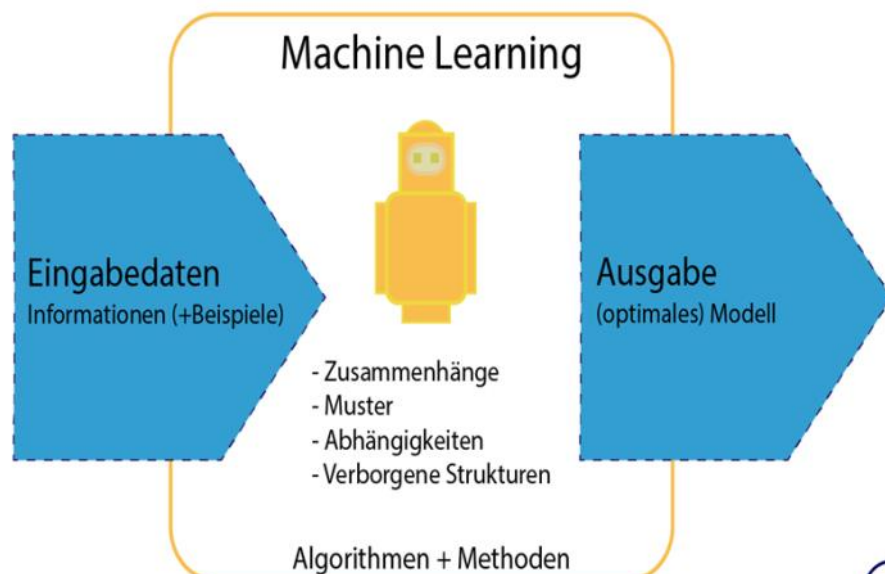
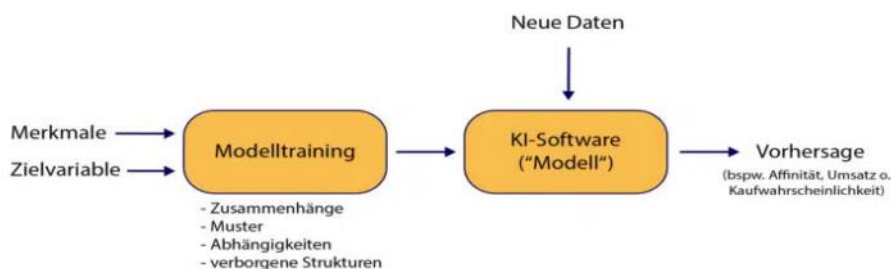
ren, Zusammenhänge identifizieren und unbekannte Muster erkennen. Diese identifizierten Muster und Zusammenhänge lassen sich auf einem neuen, unbekannten Datensatz anwenden, um so Vorhersagen zu treffen und Prozesse zu optimieren.

Machine Learning nutzt Daten, um Muster und Zusammenhänge in Daten zu identifizieren (Modelltraining). Anhand der erlernten Muster lässt sich eine Vorhersage für die Zukunft erstellen. Anders als bei der traditionellen Softwareentwicklung, liegt bei Machine Learning der Schwerpunkt auf dem selbstständigen Lernen aus Daten. Das bedeutet, dass der Algorithmus aus den Daten lernt und seinen Programmcode alleine erstellt.

### Maschinelles Lernen kann folgende Aufgaben erledigen

- **Vorhersage von Werten** auf Basis der analysierten Daten treffen (bspw. Stromverbrauch oder Umsatzforecast),
- **Berechnung von Wahrscheinlichkeiten** für bestimmte Ereignisse (bspw. Kaufwahrscheinlichkeit oder Kündigungswahrscheinlichkeit),
- **Erkennen von Gruppen** und Clustern in einem Datensatz,

## Wie funktioniert Machine Learning?



- **Erkennen von Zusammenhängen** in Sequenzen,
- **Reduktion von Dimensionen** ohne großen Informationsverlust
- **Optimierung** von Geschäftsprozessen.

Damit maschinelles Lernen funktioniert und die Software die Entscheidung treffen kann, muss ein Mensch den Algorithmus trainieren. Durch das Bereitstellen von Trainings- und Beispieldaten, kann der Algorithmus Muster und Zusammenhänge erkennen und somit aus den Daten lernen. Diesen Prozess nennt man auch Modelltraining. Maschinelles Lernen wird häufig mit den Begriffen Data Mining und predictive Analytics in Verbindung gebracht. Letztlich nutzen Data Mining und predictive Analytics die Verfahren des maschinellen Lernens. Damit Machine Learning funktioniert und ein Muster lernen kann, muss es von einem Menschen trainiert werden. Dieser Lernprozess beginnt mit einem vorbereiteten Datensatz (Trainingsdatensatz).

### Einordnung der Lernverfahren

Die Struktur eines lernenden Systems lässt sich wie folgt beschreiben:



**Agent:** Diese ausführende Einheit soll mit Erfahrungen verbessert werden; d.h. et- was lernen

**Lerneinheit:** Hier wird der Lernvorgang gesteuert und bewertet. Insbesondere wird vorgegeben, was gut und schlecht ist.

**Umwelt:** In der Umwelt soll agiert werden. Die Rückmeldung über den Ausgang bzw. den Effekt von Aktionen kommt aus dieser Umwelt. Das kann eine reale oder künstliche Umwelt sein.

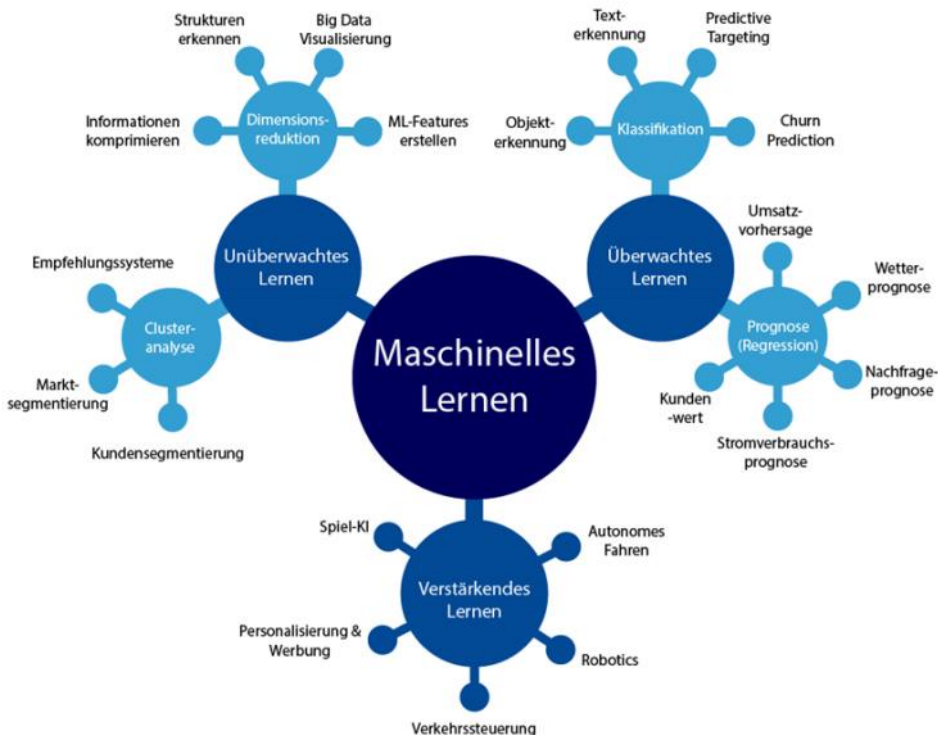
## Lernmethoden

Maschinelles Lernen besteht aus verschiedenen Arten von maschinellen Lernmodellen, die verschiedene algorithmische Techniken verwenden. Abhängig von der Art der Daten und dem gewünschten Ergebnis kann eines von vier Lernmodellen genutzt werden: überwacht, unüberwacht, teil- überwacht oder bestärkend. Innerhalb jedes dieser Modelle können eine oder mehrere algorithmische Techniken angewendet werden – je nach den genutzten Datensätzen und den beabsichtigten Ergebnissen. Algorithmen des maschinellen Lernens sind grundsätzlich dafür ausgelegt, Dinge zu klassifizieren, Muster zu finden, Ergebnisse vorherzusagen und fundierte Entscheidungen zu treffen. Die Algorithmen können einzeln oder kombiniert eingesetzt werden, um bei komplexen und unvorhersehbaren Daten die bestmögliche Genauigkeit zu erzielen.

**Überwacht:** Der Algorithmus lernt eine Funktion aus gegebenen Paaren von Ein- und Ausgaben. Dabei stellt während des Lernens ein „Lehrer“ den korrekten Funktionswert zu einer Eingabe bereit. Ziel beim überwachten Lernen ist, dass dem Netz nach mehreren Rechengängen mit unterschiedlichen Ein- und Ausgaben die Fähigkeit antrainiert wird, Assoziationen herzustellen. Ein Teilgebiet des überwachten Lernens ist die automatische Klassifizierung. Ein Anwendungsbeispiel wäre die Handschrifterkennung. Lernen kann auch ständig (Inkrementell) oder in einer Trainingsphase, die irgendwann beendet ist, erfolgen.

**Teilüberwachtes Lernen:** Nur für einen Teil der Eingaben sind die dazugehörigen Ausgaben bekannt

**Unüberwacht:** Der Algorithmus erzeugt für eine gegebene Menge von Eingaben ein statistisches Modell, das die Eingaben beschreibt und erkannte Kategorien und Zusammenhänge enthält und somit Vorhersagen ermöglicht. Dabei gibt es Clustering-Verfahren, die die Daten in mehrere Kategorien einteilen, die sich durch charakteristische Muster voneinander unterscheiden. Das Netz erstellt somit selbständig Klassifikatoren, nach denen es die Eingabemuster einteilt. Ein wichtiger Algorithmus in diesem Zusammenhang ist der EM-Algorithmus, der iterativ die Parameter eines Modells so festlegt, dass es



die gesehenen Daten optimal erklärt. Er legt dabei das Vorhandensein nicht beobachtbarer Kategorien zugrunde und schätzt abwechselnd die Zugehörigkeit der Daten zu einer der Kategorien und die Parameter, die die Kategorien ausmachen.

Des Weiteren unterscheidet man zwischen Batch-Lernen, bei dem alle Eingabe/Ausgabe-Paare gleichzeitig vorhanden sind, und kontinuierlichem (sequen- tiellem) Lernen, bei dem sich die Struktur des Netzes zeitlich versetzt entwickelt.

**Bestärkendes Lernen:** Das bestärkende Lernen ist ein Bereich des maschinellen Lernens, der sich mit der Frage beschäftigt, wie Agenten in einer Umgebung agieren sollten, um einen bestimmten Wert der kumulierten Belohnung zu maximieren. Aufgrund seiner Allgemeingültigkeit wird dieses Gebiet auch in vielen anderen Disziplinen untersucht, z. B. in der Spieltheorie, der Kontrolltheorie, der „Operations Research“, der Informationstheorie, der simulationsbasierten Optimierung, den Multiagentensystemen, der Schwarmintelligenz, der Statistik und den genetischen Algorithmen. Viele Algorithmen des Verstärkungslernens verwenden Techniken der dynamischen Programmierung. Verstärkungslernalgorithmen werden in autonomen Fahrzeugen oder beim Lernen eines Spiels gegen einen menschlichen Gegner eingesetzt.

Die folgenden Machine Learning Algorithmen können für viele Problemstellungen genutzt werden:

- Lineare Regression
- Entscheidungsbaum
- Clustering Algorithmen

- Self Organizing Maps
- Hierarchisches Clustering
- Random Forest
- künstliche neuronale Netze
- Feed forward neural networks
- Recurrent Neural Networks (RNN)
- viele weitere.

Für die **Bewertung von Lernverfahren** kann man Maßzahlen entwickeln:

- Klassifikation von „Vogel“ anhand *bekannter Attribute wie „kann-fliegen“, „hat-Federn“ usw.*
- *Vorhersage, dass ein Auto noch ein Jahr keinen defekt hat aufgrund der Parameter wie Alter, gefahrene Kilometer, Marke, Kosten der letzten Reparatur usw.*
- *Medizinischer Test auf HIV: Antikörper*
- *Vorhersage der Interessen bzw. Kaufentscheidung eines Kunden aufgrund der bisherigen Käufe und anderer Informationen (online-Buchhandel).*
- *Kreditwürdigkeit eines Kunden einer Bank, aufgrund des Einkommens, Alters, der Eigentumsverhältnisse usw.*

Ein Klassifikator mit ja/nein – Antworten klassifiziert Objekte mit Attributen, um ein zukünftiges Ereignis vorherzusagen. Es gibt

- Eine Menge von Objekten (mit innerer Struktur)
- Ein Programm  $P : M \rightarrow \{0,1\}$ , das Objekte klassifiziert
- Die wahre Klassifikation  $K : M \rightarrow \{0,1\}$

Es gilt: Im Fall  $P(x) = K(x)$  liegt das Programm richtig mit

- richtig – positiv, wenn  $P(x) = 1$  und  $K(x) = 1$
- richtig – negativ, wenn  $P(x) = 0$  und  $K(x) = 0$
- Im Fall  $P(x) <> K(x)$  liegt das Programm falsch mit
- falsch – positiv, wenn  $P(x) = 1$  und  $K(x) = 0$
- falsch - negativ, wenn  $P(x) = 0$  und  $K(x) = 1$

Nun kann man Wahrscheinlichkeiten ermitteln, mit denen P eine richtig-positive u.a. Klassifikation macht.

**Recall** (Richtig-positiv-Rate, hit rate):  $|\{x \in M \mid P(x) = 1 \wedge K(x) = 1\}| / |\{x \in M \mid K(x) = 1\}|$  ist der Anteil der richtig klassifizierten Objekte bezogen auf alle tatsächlich richtigen!

**Richtig-Negativ-Rate** (correct rejection rate):  $|\{x \in M \mid P(x) = 0 \wedge K(x) = 0\}| / |\{x \in M \mid K(x) = 0\}|$  ist der Anteil der als falsch erkannten Objekte bezogen auf alle tatsächlich falschen!

**Precision** (positive predictive value):  $|\{x \in M \mid P(x) = 1 \wedge K(x) = 1\}| / |\{x \in M \mid P(x) = 1\}|$  ist der Anteil der richtigen Objekte unter den als scheinbar richtigen erkannten.

**Negative Vorhersage Rate** (negative predictive value):  $|\{x \in M \mid P(x) = 0 \wedge K(x) = 0\}| / |\{x \in M \mid P(x) = 0\}|$  ist der Anteil der richtigen Objekte als falsch klassifizierten unter allen falsch klassifizierten!

Beispiel Gelbfieber: Körpertemperatur  $> 38,5^\circ$ , dann Gelbfieber  $\rightarrow$  Recall  $= 1$ , Precision  $= 0,0001$ ; ist keine gute Diagnose. Man muss beide Größen (Recall und Precision) ermitteln, um den Test genauer zu machen!

Nun führen wir den Begriff der Entropie ein:  $p_i$  ist eine diskrete Wahrscheinlichkeitsverteilung ( $i = 1, \dots, n$ ).

Dann gilt  $I(a_k) = \log_2(1/p_k) = -\log_2(p_k) \Rightarrow 0$ , wobei  $I$  der Informationsgehalt des Zeichens  $a_k$  ist (Symbol  $a_k$  ist wie Ziehen einer Zahl aus einer Urne).

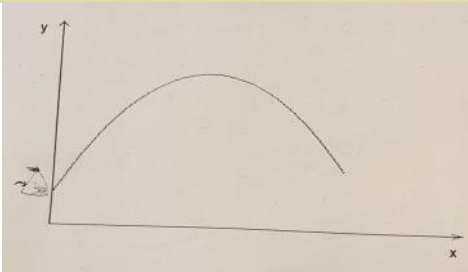
Die Entropie ist dabei der mittlere Informationsgehalt der Symbole in der Wahrscheinlichkeitsverteilung:

$H(X) = \sum p_i \cdot \log_2(1/p_i) = -\sum p_i \cdot \log_2(p_i) \Rightarrow 0$ ; dabei ist  $X$  die Zufallsvariable und  $p_i$  die Wahrscheinlichkeitsverteilung.

**Beispiel:** Nimmt man ein Bernoulli-Experiment (Kopf, Zahl), wobei Kopf  $p$  und Zahl  $1-p$  hat, dann ergibt sich eine Parabel:

Die auf der y-Achse aufgetragene Entropie hat Werte zwischen 0 und 1, die x-Werte sind  $p$  mit 0 bis 1.

d.h. die Entropie als mittlerer Informationsgehalt eines Münzwurfes ist maximal, wenn man das Zeichen nicht vorhersagen kann ( $p = 0.5$ )!



Bei  $p = 0.9$  kann man vorhersagen, dass Kopf häufig auftritt. Bei  $p = 0.1$  tritt Zahl häufig auf. Die Entropie wäre in diesen Fällen 0,46!

### Lernen mit Entscheidungsbäumen

Auch Bäume können ein Klassifikationsprogramm repräsentieren. Es sollen Algorithmen gefunden werden, die einen guten Entscheidungsbaum erstellen. Güte meint hier, dass der Baum möglichst flach ist.

Beginn. Es gibt eine Menge von Attributen  $a \in A$  mit möglichen Werten  $W_a$ . Ein Objekt ist dann  $A \rightarrow W_a$ . Ein Konzept  $K$  ist repräsentiert durch ein Prädikat  $P_k$  z.B.  $P_k(0) = \text{True}$ .

Beispiel: Bücher mit den Attributen (Autor, Titel, Seitenzahl, Preis, Erscheinungsjahr) mit beispielweise Preis  $\geq 10$  € oder Seitenzahl  $\Rightarrow 500$  Seiten

Dann ist ein Entscheidungsbaum ein Konzept  $K$  als endlicher Baum, der an inneren Knoten zum Wert eines Attributs folgende Abfragen machen kann: Für einen Wert  $v$  mit  $a \leq v$  oder  $a > v$  gibt es einen Teilbaum für „Ja“ und einen für „Nein“. Bei diskreten Attributen wird der exakte Wert abgefragt. Die Blätter des Baumes sind mit „Ja“ oder „Nein“ markiert  $\rightarrow$  ein Entscheidungsbaum  $B_k$  ist die Darstellung eines Algorithmus zum Erkennen, ob ein vorgelegtes Objekt zum Konzept  $K$  gehört.

Wenn es nur diskrete Attribute gibt, dann entsprechen die Konzepte genau den Entscheidungsbäumen: Zu jedem Konzept kann man offenbar eine (aussagelogische) Formel in der disjunktiven Normalform angeben:

$a_1 = v_1 \wedge \dots \wedge a_n = v_n$  als Konjunktion, wenn die Tupel  $(v_1, \dots, v_n)$  im Konzept enthalten sind. Diese kann man dann in einen Entscheidungsbaum überführen.

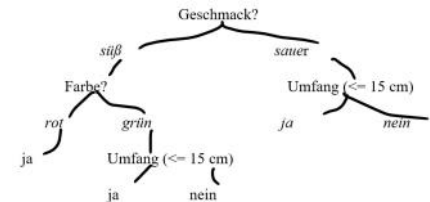
Bei reellwertigen Attributen ist dies nicht ganz einfach!

Beispiel: Als praktische Anwendung kann man reale Konzepte mit einer Menge von Attributen beschrieben werden. Für Tiere beispielsweise gilt:

Größe	Reell
Gewicht	Reell
Kann fliegen	boolsch
Nahrung	pflanzlich/tierisch
Körpertemperatur	Reell

Die Menge der Insekten könnte in einem Ereignisbaum mit den obigen Attributen klassifiziert werden.

Beispiel: Nun klassifizieren wir Äpfel



Es gibt nun verschiedene Algorithmen, die die Aufgabe lösen, einen Entscheidungsbaum für ein Konzept zu lernen. Die Anzahl der Fragen soll gering sein!

Bäume können sehr umfangreich werden. Dann kann man einen Baumschnitt durchführen: Wenn eine gewisse Baumtiefe erreicht ist, wird der Teilbaum durch ein Blatt ersetzt und das Blatt hat das Ergebnis des Teilbaums.

Entscheidungsbäume können trotz Fehlern funktionieren: Der Algorithmus kann Ersatzattribute vorsehen. Mit neuen Daten können dann die Ersatzattribute einbezogen werden.

Lernverfahren können nun mit unterschiedlichen Attributen und auch mit der Nutzung von Entropieverfahren (Informationsgehalt) weiter entwickelt werden. Dabei können durchaus komplexe Testverfahren eingesetzt werden: Modelle für maschinelles Lernen können bezüglich des Algorithmus und seines Outputs so komplex sein, dass sie vom Menschen weder erklärt noch verstanden werden können. Dies wird als „Blackbox“-Modell bezeichnet. Es stellt für Unternehmen ein Risiko dar, wenn sie nicht feststellen können, wie und warum ein Algorithmus zu einem bestimmten Ergebnis oder einer bestimmten Schlussfolgerung oder Entscheidung gelangt ist.

Beispiel z.B. Aus „Learning Analytics“: Analyse von saisonalen Effekten bei Lernplattformen. Wie ist die Nutzung vor, in und nach der Prüfungswoche (Log-File Analyse). Die Teilnehmerstatistik der Nutzung wird ausgewertet. Dabei ist die Teilnahmefrequenz vor und in der Prüfungswoche vier bis zehn-mal so hoch wie vorher (und nachher). Was sagt das über das (nicht kontinuierliche) Lernverhalten?

### 8. Neuronale Netze und Deep Learning

Deep Learning ist ein Teilbereich von Machine Learning. Er beschreibt einen vom menschlichen Gehirn inspirierten Aufbau

des Programms aus künstlichen neuronalen Verbindungen, das neuronale Netzwerk. Während klassisches Maschine Learning auf feste Modellgruppen zur Erkennung und Klassifizierung zurückgreift, entwickelt Deep Learning Algorithmen eigenständig weiter oder erstellen eigenständig neue Modell-ebenen innerhalb neuronaler Netzwerke. Grundlagen zu den neuronalen Netzen und Stärken der Kombination von KI-Ansätzen (Unterschied siehe Graphik).

**Ein Ausgangspunkt:** Man finde einen Satz von möglich wenigen Merkmalen, der als Eingabe für einen Lernalgorithmus verwendet werden kann.

Fluch der Dimension: 10 Mio. Pixel mit 3 Byte pro Bildpunkt für ein Bild sind notwendig!

Möglicher Ausweg: Reduktion der Dimension des Vektorraums in Richtung der höchsten Varianz, um dann die Daten mittels linearer Abbildung auf die Hauptkomponenten zu projizieren. Und da entstand eben seit 1995 „Deep Learning“.

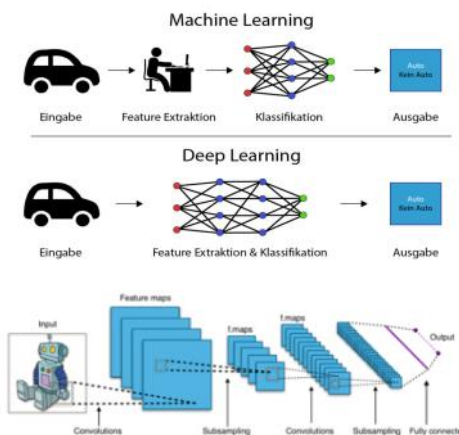
Dabei wird das neuronale Netz trainiert, indem ihm immer wieder Daten vorgelegt werden. Durch diese Wiederholung lernt das neuronale Netz die Daten jedes Mal exakter einzuordnen. Dabei wird die Natur als Vorbild genommen: zweiteiliges mehrschichtiges Netzwerk, wobei die erste Schicht „vortrainiert“ wird und die zweite Schicht mit überwachtem Lernen arbeitet.

### Wie funktioniert das neuronale Netzwerk?

Das funktioniert, indem die Gewichtung für die einzelnen Verbindungen zwischen den Neuronen-Schichten immer wieder angepasst werden. Das in den Lerndurchläufen erzeugte Modell kann dann auch auf Daten angewandt werden, die die Künstliche Intelligenz im Training noch nicht kennengelernt hat. Dabei hat die Architektur des Netzwerkes „versteckte“ Schichten, die nicht direkt an die Eingabe- oder Ausgabe-Schicht gekoppelt sind, werden sie »Deep Neural Networks« genannt. Deep Neural Networks können Hunderttausend oder Millionen Neuronen-Schichten aufweisen. Es entstehen sehr komplexe Architekturen, womit beim sogenannten »Deep Learning« immer komplexere Probleme gelöst werden können.

Die Verfahren zum Einsatz der vielen Schichten nennen sich „Convolutional Neural Networks“ (CNN), „Deep Belief Networks“ (CBN) und „Recurrent Neural Networks“ (RNN).

CNN nutzt beispielweise Prinzipien der linearen Algebra (Matrizenmultiplikation), um Muster in einem Bild zu erkennen. Dies wird mit graphischen Verarbeitungseinheiten zum Trainieren der Modelle bewerkstelligt. Die Eingabedaten sind Bilder, die nach verschiedenen Aspekten



(Verzerrungen, Gewichtungen) ausgewertet werden. Auf diese Weise kann CNN zwischen Bildern unterscheiden und sie klassifizieren. Eine erste Version von CNN -LeNet (benannt nach dem Erfinder Yann LeCun) konnte handgeschriebene Ziffern erkennen (Lesen von Schecks und Postleitzahlen). Diese Konstruktion aus 1980 wurde 2012 wieder aufgegriffen und mit verbesserter Technologie konnten große Datensätze abgearbeitet werden.

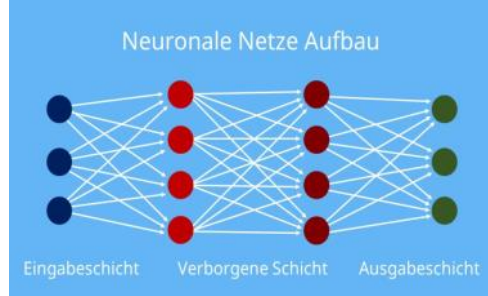
Wie oben schon angeführt, wurden die Anzahl der Schichten („Layer“) erhöht. Dabei wurde eine „Faltungsschicht“ als erste Schicht eingeführt, die Informationen über die weiteren Schichten enthält. Dabei werden „Bilder“ in Pixel dargestellt. Die Faltungsschicht wandelt nun das gegebene Bild in numerische Werte um und ermöglicht es dem Netzwerk „räumliche Bilder“ zu „verstehen“. „Faltung“ ist mathematisch eine Operation, um zwei Datensätze zusammenzuführen.

Ein „Pooling“ von Schichten wird nun verwendet, um die Dimension des Bildes zu reduzieren. Das Ergebnisfeld enthält dann die Durchschnittswerte bei der Berechnung des Musters des Empfangsfeldes. Als zweite Methode wird das Pixel mit dem maximalen Wert ausgewählt und sendet es zum Ausgangsfeld. Dabei kommen „Filter“ zum Einsatz, die z.B. die Größe 3\*3 haben und das Eingangsbild „überstreichen“. Für die einzelnen Ausschnitte werden Matrizenmultiplikationen durchgeführt.

Struktur eines typischen CNNs zur Bildklassifikation. Subsampling entspricht Pooling. Dieses Netz besitzt pro Convolutional Layer mehrere Filterkernel, sodass Schichten an Feature Maps entstehen, die jeweils die gleiche Eingabe bekommen, jedoch aufgrund unterschiedlicher Gewichtsmatrizen unterschiedliche Features extrahieren.

### Praxisbeispiele

- Datensatz mit handgeschriebenen Zahlen (Ziffern von 0 bis 9), um alle Schreibweisen dieser Ziffern gut trennen zu können.
- RNNs können mit historischen Daten



trainiert werden und schaffen dann selbst „Kunstwerke“.

- Gedichte aus der Literatur können gelernt und abgewandelt werden.

### Literatur

**Angerer** Christoph (2018), Neuronale Netze – Revolution für die Wissenschaft? In: Spektrum der Wissenschaft 1.18, S 12 - 21

**Dilger** Werner (2004-06), Künstliche Intelligenz in der Schule, Vorlesungen an der Uni Chemnitz, Chemnitz

**Forum Neue Medien** in der Lehre (2019), Learning Analytics: Einsatz an österreichischen Hochschulen, Konzeptpapier

**Gopnik** Alison (2018), Computer lernen wie die Kinder in: Spektrum der Wissenschaft 4.18, S 74 - 79

**Kwon** Diana (2018), Robotik: Selbständig lernende Roboter In. Spektrum der Wissenschaft 7.18, S 66 - 72

**Lenzen** Manuela (2018), Künstliche Intelligenz – was sie kann & was uns erwartet, Beck-Verlag, München

**Mödrtscher** Felix (2019), Neuronale Netze und Deep Learning, Seminarmanuskript, FH Technikum Wien

**Nielsen** Michael (2018), Künstliche Intelligenz – Alpha Go – Computer lernen Intuition in: Spektrum der Wissenschaft 1.18, S 22 - 27

**Schmidt-Schauß** Manfred und Sabel David (2012), Einführung in die Methoden der künstlichen Intelligenz, Vorlesung am Institut für Informatik, Goethe-Universität Frankfurt am Main

**Stettes** Oliver (2019), Keine Angst vor Robotern – eine Aktualisierung, IW-Report 17/2019, Köln

**Unger** Luise, Mathematische Grundlagen, Kurseinheit 7: Logik; Fernuni Hagen 2020 Mathematik-Fakultät

**Universität München**, Panel (2019), wie schlau kann KI werden? LM-Uni München

**Wikipedia** (2016), Turingtest, <https://de.wikipedia.org/wiki/Turing-Test>



# ZigBee, Zwischenstecker und Tasmota

Martin Weissenböck

Themen in diesem Beitrag:

- ZigBee: besser als proprietäre Apps beim Aufbau der Home-Automation.
- Tipps für den Kauf von Zwischensteckern
- Tasmota kennen lernen: Konfigurieren der Tasmota-Firmware für einen Zwischenstecker
- Node-Red und die Tasmota-Firmware

Was wird hier *nicht* behandelt:

- Ersatz der Firmware von proprietären Wifi-Komponenten durch die Tasmota-Firmware
- Tasmota für die Microcontroller ESP-8266 und ESP-32

Diese Themen werden in weitere Beiträgen behandelt

## 1 ZigBee

Über ZigBee habe ich schon in der Ausgabe 172 geschrieben:

<https://clubcomputer.at/2022/02/02/das-automatische-heim-mit-zigbee/>

Kurz zur Erinnerung: ZigBee ist eine Spezifikation für drahtlose Netzwerke, die in der Home-Automation weit verbreitet ist. Viele Komponenten, wie Schalter, Lampen oder Zwischenstecker, können mit Apps wie SmartLife oder eWeLink zwar sehr einfach und durchaus komfortabel gesteuert werden. Der Nachteile:

- Daten des Heimnetzes (SSID [Accesspointname] und Passwort) müssen in die App eingetragen werden. Diese Daten werden in einem zentralen Server (vermutlich in China) gespeichert. Wollen wir das? Und überträgt die App beim Nach-Hause-Telefonieren (© E.T.) vielleicht noch andere Daten?
- Der Umweg über China verzögert die Steuerung.
- Die Steuerung läuft über das WLAN – besser wäre ein eigenes, sicheres Netz
- Was passiert, wenn dieser zentrale Dienst nicht funktioniert? Dann „steht“ die Steuerung.

Auch wenn die Apps viele Einstellmöglichkeiten bieten, können Sonderwünsche kaum erfüllt werden. Somit spricht viel für eine lokale Lösung. Eigene Problemstellungen können umgesetzt werden. Für technikaffine Menschen ein weites Betätigungsfeld! Die Tasmota-Firmware spielt dabei eine wichtige Rolle – mehr darüber in diesem Beitrag.



Und was kommt als nächstes? Viele verschiedene Standards machen den Konsumenten das Leben schwer. Der neue Standard „Matter“ ermöglicht die Kommunikation zwischen Komponenten unterschiedlicher Hersteller. Mit Tasmota geflashte Bauteile können über ein Matter-Gateway verbunden werden.

## 2 Zwischenstecker

Eine beliebte Anwendung der Home-Automation ist das Schalten von beliebigen Verbrauchern über einen Zwischenstecker, der (wie der Name sagt) zwischen Steckdose und Verbraucher geschaltet wird. Worauf ist beim Kauf zu achten?

### Die schlechte Nachricht

Bei Lidl werden die ZigBee-Silvercrest-Komponenten nur mehr in Reststücken verkauft. Schade.

### Die gute Nachricht

Auf der Amazon-Webseite gibt es Ersatz. Wer Amazon nicht mag, kann ja selbst im Internet andere günstige Angebote suchen, leider oft mit unangenehmen Versandkosten.

## ZigBee

Ich behandle hier die Home-Automation mit ZigBee, daher muss auch „ZigBee“ auf der Verpackung stehen. „WiFi“-Zwischenstecker sind meist etwas billiger, aber binden den Zwischenstecker fast immer an proprietäre Software. *Wichtig bei der Auswahl!*

## Größe

Das Bild zeigt drei verschiedene Zwischenstecker.

Der erste ist von IKEA und belegt zwei Plätze auf der Steckerleiste, der zweite sogar drei! Nur der dritte Typ („Gosund“, auch unter anderen Namen erhältlich) ist so konstruiert, dass die Nachbarplätze nicht belegt werden und somit auch zwei

Zwischenstecker nebeneinander Platz finden. *Meine Empfehlung!*

## Lokaler Schalter

Der Zwischenstecker soll einen Taster haben, damit der Verbraucher auch lokal ein- und ausgeschaltet werden kann. Leider nicht selbstverständlich – so fehlt beispielsweise beim IKEA TRÅDFRI-Zwischenstecker dieser Taster. *Unbedingt darauf achten!*

## Messfunktion

Zwischenstecker, die Strom, Spannung, Leistung, Energie und in manchen Fällen auch den Leistungsfaktor messen, kosten auch nicht mehr. *Meine Empfehlung!*

## 3 Was ist nun Tasmota?

Meine ersten Versuche mit der Home-Automation habe ich mit WiFi-Komponenten gemacht. Die Nachteile waren bald klar: WLAN-Daten in eine chinesische App eintragen, überlastetes 2,4-GHz-Netz in der näheren Umgebung, wenig Konfigurationsmöglichkeiten usw. ZigBee hat dagegen viele Vorteile.

Viele Komponenten der Home-Automation verwenden das Heim-WLAN. Manche können für ZigBee umprogrammiert werden – das 2,4-GHz-Band wird ja ebenso verwendet. So hat die Firma Sonoff viele interessante Komponenten zum Schalten und Steuern herausgebracht, meistens mit dem ESP-8266 Chip. Die Komponenten werden für die WLAN-Anwendung mit zentralem Server ausgeliefert. Erfreulicherweise bereitet Sonoff seine Geräte auf ein Umprogrammieren („Flashen“) vor, um diese Abhängigkeit zu entfernen. Theo Arends hat das Programm Tasmota entwickelt, um dieses Umprogrammieren zu vereinfachen.

Tasmota steht für Theo-Arends-Sonoff-MQTT-OTA. MQTT (MQTT – Wikipedia) ist das Protokoll von ZigBee und OTA steht für „Over the air“, also für das drahtlose Umprogrammieren. Inzwischen kann Tasmota auch für Nicht-Sonoff-Geräte verwendet werden. Siehe <https://www.tasmota.info/>



Sobald die Tasmota-Software installiert ist, kann die Funktion mancher Bauteile zusätzlich verändert oder erweitert werden. Diese Bauteile können dann für ZigBee konfiguriert und in einer Home-Automation-Steuerung eingesetzt werden.

Und was mache ich mit meinen alten WiFi-Komponenten? Soweit möglich mit Tasmota flashen!

#### 4 Zwischenstecker-Angebote im Internet

Ich habe Angebote mit 45 € pro Stück (!) gefunden: groß, ohne Leistungsmessung, aber dafür von einem namhaften Hersteller. Ja, immer noch herrscht bei einigen die Goldgräberstimmung! Da zahlt es sich schon aus, noch ein wenig weiter zu suchen.

**Mein Vorschlag:** Zwischenstecker wie im Bild ganz rechts. Sie sind unter verschiedenen Bezeichnungen wie Gosund oder NOUS erhältlich. Ich vermute, dass alle von demselben Produzenten in China kommen. Hier geht es um die Versionen NOUS A1Z, A1T und A1. Alle Preise habe ich von der Amazon-Webseite.

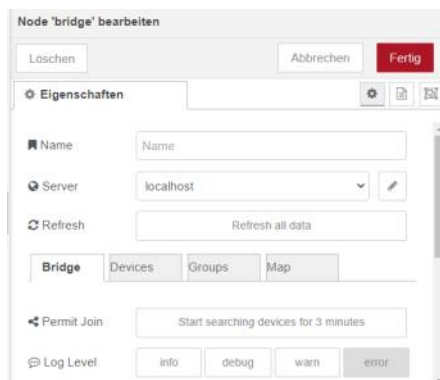
- Der Zwischenstecker NOUS A1Z ist schon für ZigBee vorbereitet. 4 Stück davon kosten 55,45 €. Das ist zwar der teuerste Stecker in dieser Reihe, aber er kann sofort in einer ZigBee-Homeautomation verwendet werden. Klein, schaltet 16 A und misst elektrische Größen. Siehe Abschnitt 5.
- Den Typ NOUS A1T gibt es um 50,41 € für 4 Stück, also etwas mehr als 12 € pro Stück. Elektrischen Daten wie beim A1Z. Tasmota ist installiert, aber noch nicht für ZigBee konfiguriert. Mehr dazu im Abschnitt 6.
- Die „GHome Smart WLAN Steckdose“ entspricht dem NOUS A1 und ist etwas billiger: rund 39 € für 4 Stück. Mehr im Abschnitt 7.

#### 5 Der schnellste Weg: NOUS A1Z

Der Zwischenstecker A1Z ist schon für ZigBee vorprogrammiert. In der Packung liegt eine Beschreibung bei, die die Installation einer App empfiehlt. Genau das wollen wir nicht. Die App ist aber nicht notwendig, den Hinweis einfach ignorieren!

Mit Node-Red kann der Zwischenstecker sofort verwendet werden. ZigBee und Node-Red sollten schon eingerichtet und aktiv sein, um den adaptierten Stecker gleich ausprobieren zu können.

Der Vorgang ist immer derselbe. Zuerst den Zwischenstecker ans Netz anschließen, eine rote LED beginnt zu blinken. Dann in der ZigBee-Bridge das Anlernen aktivieren: „Start searching devices for 3 minutes“ anklicken.



Nach ein paar Sekunden hört die LED auf zu blinken – fertig. Das Anlernen kann wieder deaktiviert werden. Nach 3 Minuten endet die Suche nach neuen Komponenten auch automatisch. „Refresh all data“ anklicken, dann „Devices“. Und da ist schon der neue Stecker:



Vorläufig ist der Name gleich der IEEE-Adresse (0xa4c...). Benennen wir den Zwischenstecker auf „A1Z-2“ um - es ist mein zweiter A1Z-Zwischenstecker. Abspeichern mit „set“ nicht vergessen!



Hier ein einfacher Node-Red Flow zum



Testen des Steckers:

Der erste inject-node schickt zum Umschalten {"state":"toggle"}:



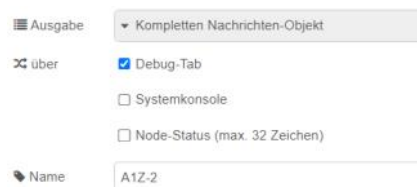
Im Zigbee2mqtt Node „out“ wird eingetragen:



Mehr ist nicht notwendig. „Übernahme (deploy)“ rechts oben nicht vergessen. Mit einem Klick auf „toggle“ wird der Stecker ein- und ausgeschaltet.

Der zweite Teil des Flows fordert den Status an. Dazu einen leeren String senden: {"state":""}.

Das ist die Konfiguration des Debug-Nodes:



Nach einem Klick auf den zweiten inject-Knopf wird im Debug-Bereich rechts angezeigt:

28.4.2023, 11:07:01 node: A1Z-2

zigbee2mqtt/A1Z-2 : msg : Object

```

▼ object
  _msgid: "40e3221b516cb862"
  ▼ payload: object
    current: 0.06
    energy: 0.01
    indicator_mode: "off/on"
    linkquality: 160
    power: 7
    power_outage_memory: "restore"
    state: "ON"
    voltage: 230
    topic: "zigbee2mqtt/A1Z-2"
  ► payload_in: object
  ► changed: object
  ► payload_raw: object
  ► homekit: object
  ► format: object
    selector: "zigbee2mqtt_A1Z-2"
  ► item: object

```

#### Die Messwerte

Beschreibung	Elektrische Größe	Einheit
current	Strom	A
energy	Arbeit = Energie	kWh
power	Wirkleistung	W
voltage	Spannung	V

Die restlichen Einträge:

**indicator\_mode:** "off/on" Die LED leuchtet im eingeschalteten Zustand und ist dunkel im ausgeschalteten Zustand. Andere Werte: "on" = ständig leuchten, "off" = ständig dunkel, "on/off" = umgekehrt wie "off/on".

**link\_quality:** 160 Ein Zahl zwischen 0 und 255, die die Qualität der drahtlosen Verbindung angibt.

**power\_outage\_memory:** restore Nach dem Abschalten und dem Wiederkehren der Spannung wird der letzte Zustand wieder hergestellt. Mögliche andere Werte: "on" = eingeschaltet, "off" = ausgeschaltet.

**state:** "ON" Der aktuelle Zustand: "ON" = eingeschaltet, "OFF" = ausgeschaltet.

Das **JavaScript msg Object** enthält noch viele weitere Daten, auf die ich hier nicht eingehe.



## 6 NOUS A1T

### 6.1 Den Zwischenstecker konfigurieren

Der Zwischenstecker ist bereits mit der Tasmota-Software „geflasht“, also programmiert, muss aber noch für ZigBee adaptiert werden. Der A1Z hat doch schon gut funktioniert. Warum sollen wir uns mit dem A1T beschäftigen?

- Der A1T liefert weitere Messwerte (Blindleistung, Scheinleistung und Leistungsfaktor).
- Das Konfigurieren der A1T zeigt, wie die Tasmota-Firmware angewendet wird. Diese Erfahrungen sind nützlich, wenn andere Hardware mit Tasmota geflasht werden.
- Der A1T ist etwas billiger als der A1Z.

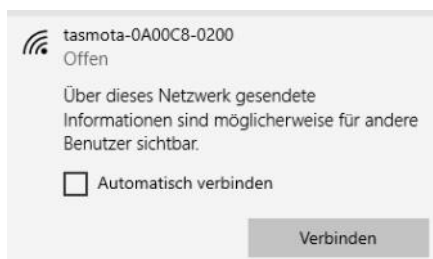
Los geht's – es schaut auf den ersten Blick kompliziert aus, ist aber in ein paar Minuten erledigt.

ZigBee und Node-Red sollten schon eingerichtet und aktiv sein, um den adaptierten Stecker gleich ausprobieren zu können. Bei mir läuft die ZigBee-Steuerung – der Coordinator oder das Gateway – auf einem Linux-Rechner mit der Adresse 192.168.12.71. Bitte ans eigene Netz anpassen. Natürlich kann auch ein Raspberry verwendet werden – ist leider nicht mehr so preiswert.

**Schritt 1:** Steckdose ans Netz anstecken. Der Zwischenstecker erzeugt dabei ein eigenes WLAN, dessen SSID (Name) mit „tasmota“ beginnt. Auf einem Mobiltelefon dieses WLAN suchen und dort anmelden. Oft meldet das Mobiltelefon, dass kein Internetzugang möglich ist. Das ist in Ordnung, bestätigen!

Der Vorgang funktioniert auch mit dem PC, wenn dieser einen WLAN-Zugang hat. Die Screenshots hier stammen von meinem PC.

Falls kein passendes WLAN erscheint, den Zwischenstecker durch Drücken der Taste für 40 Sekunden zurück setzen.



#### Verbinden!

### WLAN

WLAN 2

Ein

tasmota-0A00C8-0200  
Kein Internet, offen

Die Angabe nach „tasmota“ (hier: „0A00C8“) notieren. Übrigens: das sind die letzten 6 Ziffern der hexadecimalen IEEE-Adresse.

**Schritt 2:** Die Steckdose muss im lokalen Netz angemeldet werden. Ich habe dafür ein eigenes WLAN mit der SSID „seminarhaus2“ angelegt, das nur für diese Flash-Arbeiten verwendet wird und nach dem Umprogrammieren wieder abgedreht wird. Nach dem Anmelden die Webseite <http://192.168.4.1> aufrufen.



Das lokale Netzwerk (hier: „seminarhaus2“) auswählen und mit dem Passwort anmelden. Mit „Save“ speichern. Diese Information bleibt meines Wissens im lokalen Netz und wird nicht nach China geschickt.

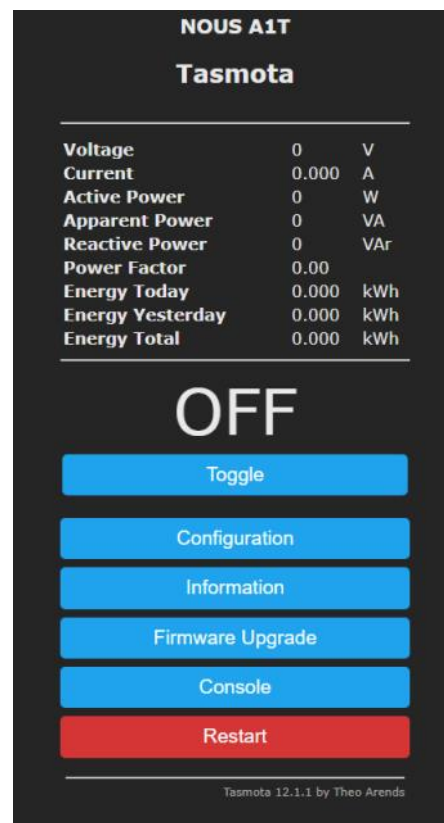


**Schritt 3:** Die folgende Seite erscheint automatisch. Wenn das nicht der Fall ist, über einen Browser die Adresse 192.168.4.1 aufrufen. Dann wird eine neue Adresse kurz angezeigt (in dem Beispiel: 192.168.10.13) – rasch notieren! Die Weiterleitung erfolgt automatisch, aber wenn es nicht klappt, müssen wir die Adresse von Hand aus eintragen.



**Schritt 4:** Das WLAN „tasmota...“ ist nun abgeschaltet. Wer beim Schritt 1 das WLAN am PC auf „tasmota...“ eingestellt hat, stellt es jetzt wieder auf das lokale Netzwerk zurück, also auf jenes Netzwerk, das wir im Schritt 2 angegeben haben. Der PC und der Zwischenstecker müssen jetzt im selben Netzwerk sein.

**Schritt 5:** Falls die Adresse aus Schritt 3 nicht schon automatisch aufgerufen wird, jetzt über den Browser öffnen (in meinem Beispiel mit <http://192.168.10.13>).



Beim Anklicken von „Toggle“ muss der Schalter jetzt hörbar klicken. Wenn ein Verbraucher angeschlossen ist, zum Beispiel eine Lampe, ist der jetzt eingeschaltet.

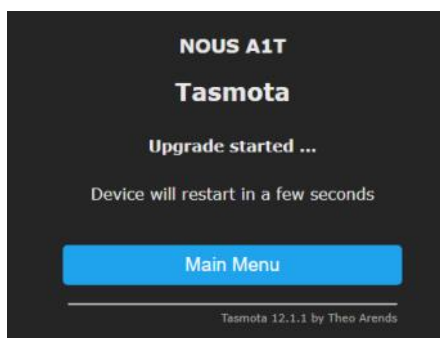
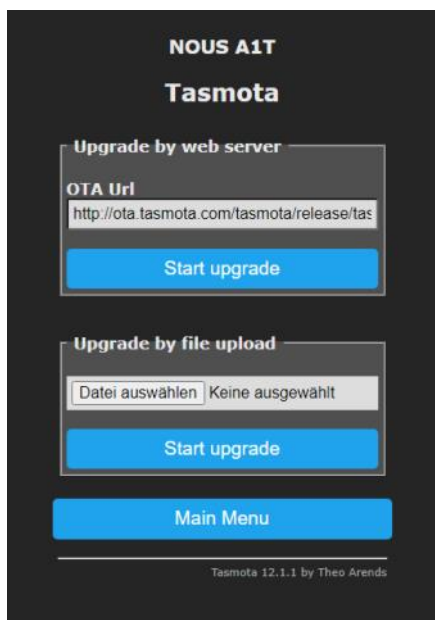
#### Schritt 6: Update

Die aktuellste Firmware einspielen. Ende April 2023 war das die Version 12.5.0. „Firmware update“ jetzt anklicken.

Die Url ist schon voreingestellt. An dieser Stelle könnte auch eine andere Softwareversion eingespielt werden, z.B. die deutschsprachige Version dieser Zwischensteckersteuerung. Allgemein ist das der Punkt, an dem beim Flashen einer beliebigen OTA-Komponente die passende Firmware ausgewählt wird.

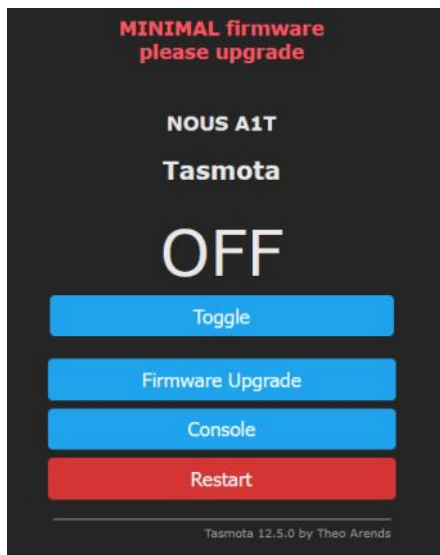
Unter „Upgrade by web server“ den Button „Start upgrade“ anklicken.



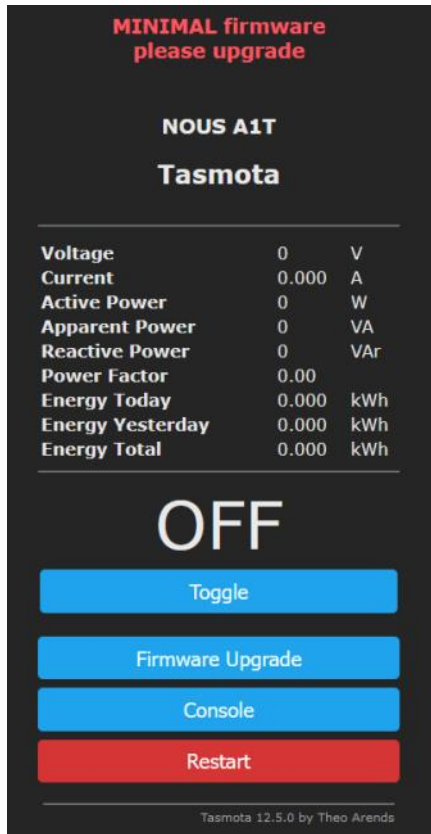


Dauert etwas – in dieser Zeit die Verbindung nicht unterbrechen!

Dann erscheint die Startseite wieder, rechts unten mit der neuen Versionsnummer.



Dann wird angezeigt:



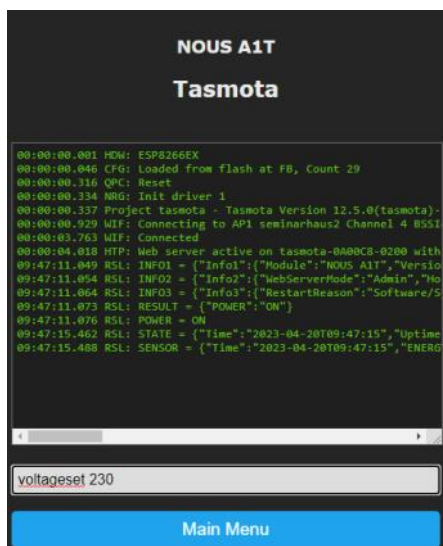
Trotz des Hinweises „please upgrade“ sind für das Upgrade keine weiteren Schritte notwendig. Nach einem Restart verschwindet der Hinweis.

Dann zum „Main Menu“.

#### Schritt 7: Kalibrieren

Die angezeigte Spannung ist vermutlich nicht korrekt. Die Messeinrichtung des Zwischensteckers muss nun kalibriert werden. Ein Voltmeter sollte jeder besitzen. Wenn nicht, nehmen wir an, dass die Spannung 230 Volt beträgt.

„Console“ anklicken und in die Kommandozeile „voltage set 230“ (bzw. die gemessene Spannung) eintragen. Eingabetaste drücken.

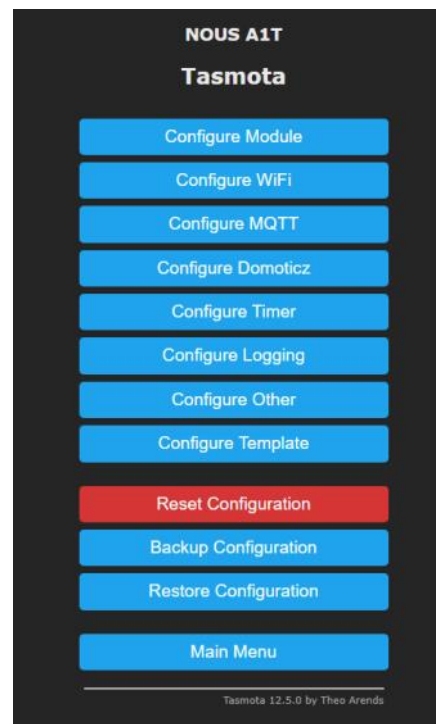


Zurück ins „Main Menu“

Jetzt müssen die Energie-Anzeigen stimmen. Gegebenenfalls mit „Toggle“ auf „ON“ stellen.

#### Schritt 8: Konfigurieren

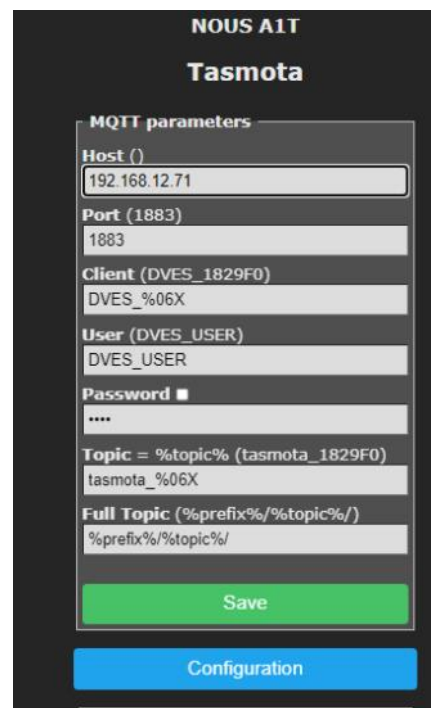
„Configuration“ anklicken.



Dann „Configure MQTT“ wählen.

Host eintragen: bei mir 192.168.12.71.

Der Rest bleibt ungeändert.



Mit „Save“ speichern. Das Gerät startet neu.

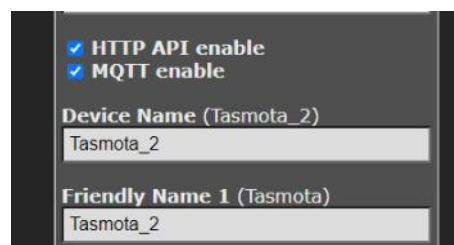


### Schritt 9: Umbenennen

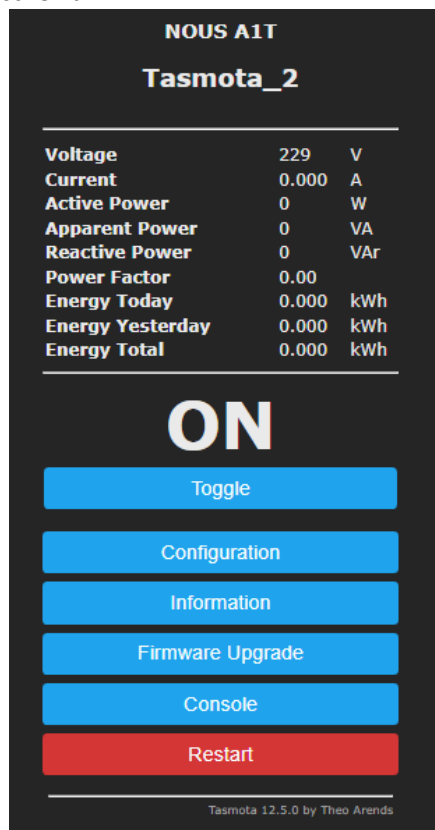
Kann auch ausgelassen werden. Umbenennen, zum Beispiel auf „Tasmota\_2“.

Na ja, nicht sehr kreativ. besser wäre vielleicht „Stehlampe“ oder ähnliches.

„Configuration“, dann „Configuration Other“ wählen:



Device Name eintragen und wieder mit „Save“ speichern. Nach dem Neustart erscheint:



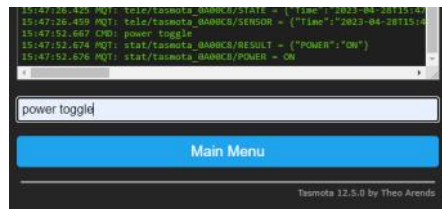
Das Fenster kann offen bleiben.

Der Zwischenstecker kann immer noch über „Toggle“ geschaltet werden. Ausprobieren!

Übrigens können alle Funktionen auch über die Konsole ausgelöst werden. Zum Toggeln in das Eingabefeld

power toggle

eingeben und Eingabe drücken.



### Schritt 10: Information

Ein Klick auf „Information“ liefert alle Einstellungen des Zwischensteckers:

Zurück ins „Main Menu“.

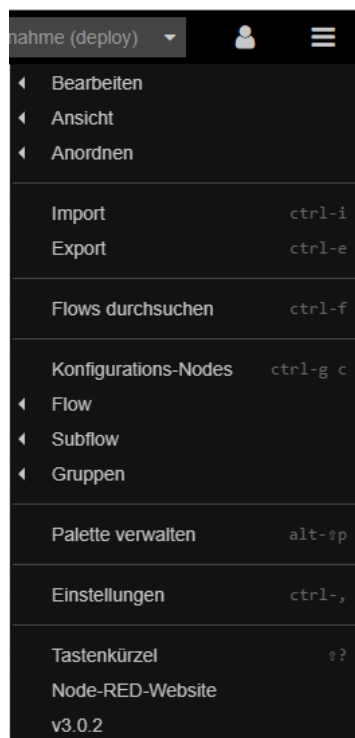
### 6.2 Erproben mit Node-Red

Node-Red ist ein Programmierwerkzeug zum Verbinden von Hardwarekomponenten (Node-RED (nodered.org)).

Der Schalter kann immer noch direkt über seine Webadresse (hier: 192.168.10.13) gesteuert werden. Wir steuern ihn jetzt *zusätzlich* über das ZigBee-Netzwerk und verwenden Node-Red.

In einem neuen Fenster Node-Red aufrufen, normalerweise über Port 1880 am ZigBee-Server. Bei mir ist daher <http://192.168.12.71:1880> zu öffnen.

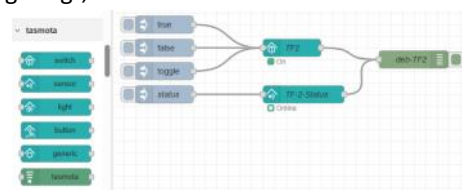
Für Tasmota steht ein eigener Satz an Nodes bereit. Zum Installieren den Menü-Button rechts oben anklicken, dann „Palette verwalten“



Weiter: „node-red-contrib-tasmota“ suchen und installieren.

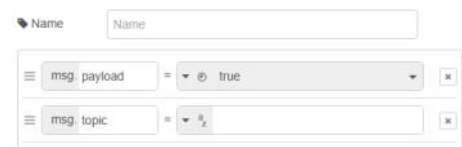


Unter der Überschrift „tasmota“ sind im Werkzeugkasten (Dashboard) auf der linken Seite mehrere neue Nodes zu sehen; damit ein paar Verbindungen, wie rechts gezeigt, aufbauen.

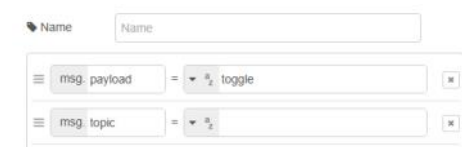


Dazu ein paar Details:

Die „inject“s für „true“ und „false“ (gezeigt nur für „true“) sehen so aus:



Für „toggle“ wird der String „toggle“ übergeben:



Alle drei gehen an einen „switch“-Knoten. Als „Device“ wird die Bezeichnung aus den Eigenschaften eingetragen, *aber mit einem „\_“ statt eines „.“*. In unserem Beispiel also „tasmota\_0A00C8“:



Bei „Broker“ auf den Bleistift rechts klicken und „Name“ und „Address“ ausfüllen.

Der Name kann frei gewählt werden, „Tasmota“ passt ganz gut. Normalerweise läuft ZigBee auf dem „localhost“.



Name: Tasmota

Address: localhost Port: 1883

☐ Enable secure (SSL/TLS) connection

Client ID: Leave blank for auto generated

Username:

Speichern und „Übernahme (deploy)“ rechts oben nicht vergessen! Mit Klicks auf „true“, „false“ und „toggle“ schaltet das Relais. Die Debuggerspalte zeigt:

```
22.4.2023, 17:08:57 node: deb-TF2
switch1: msg: Object
*{ topic: "switch1", payload: true, _msgid: "e3dd12f2f557e23c" }
22.4.2023, 17:08:59 node: deb-TF2
switch1: msg: Object
*{ topic: "switch1", payload: false, _msgid: "89267255a62f5588" }
22.4.2023, 17:09:02 node: deb-TF2
switch1: msg: Object
*{ topic: "switch1", payload: true, _msgid: "207ea5122682941f" }
```

Interessant ist die Status-Abfrage.

An den „sensor“-Knoten wird ein leerer String gesendet:

Name: status

msg: payload =

msg: topic =

Im Tasmota-Knoten wird, ähnlich wie oben, eingetragen. Ich habe den Namen „TS-2-Status“ vergeben, weil damit der Status meines zweiten Tasmota-Zwischensteckers aufgerufen wird.

Settings Advanced

Broker: Tasmota

Device: tasmota\_0A00C8

Name: TF-2-Status

☒ Dashboard ☐ Send enable/disable message

Bezeichnung	Bedeutung	Einheit, Norm
Time	aktuelle Zeit	ISO 8601
TotalStartTime	Startzeitpunkt	ISO 8601
Total	gesamte Arbeit = Energie seit dem Startzeitpunkt	kWh
Yesterday	Arbeit, Energie gestern	kWh
Today	Arbeit, Energie heute	kWh
Power	Wirkleistung	W
ApparentPower	Scheinleistung	VA
ReactivePower	Blindleistung	Var
Factor	Leistungsfaktor	cos(φ)
Voltage	Spannung	V
Current	Strom	A

Der Klick auf „status“ liefert in der Debug-Spalte (die Einträge sind aufgeklappt):

```
msg: Object
  object
    topic: undefined
    payload: object
      Time: "2023-04-22T16:31:52"
      ENERGY: object
        TotalStartTime: "2023-04-22T14:23:24"
        Total: 0.002
        Yesterday: 0
        Today: 0.002
        Power: 10
        ApparentPower: 14
        ReactivePower: 9
        Factor: 0.75
        Voltage: 228
        Current: 0.06
      _msgid: "637caef60161653c"
```

Interessant sind alle Energie- und Leistungsdaten, sie sind wesentlich informativer als beim A1Z.

### 7 NOUS A1 und andere

Viele WiFi-Komponenten können durch ein Update „Over-the-air“ (drahtlos) mit neuer Firmware versorgt werden. Um WiFi-Geräte für Tasmota umzuprogrammieren, benötigen wir einen Accesspoint, der den chinesischen Server vortäuscht. Ein entsprechend programmierter Raspberry mit WLAN wird dafür gern verwendet. Sobald nun die Tasmota-Software installiert ist, geht es wie hier im Abschnitt 6.1 beschrieben weiter. Prinzipiell werden auch andere WiFi Home-Automation Komponenten so für Tasmota und ZigBee vorbereitet. Ja sogar ein billiges ESP-8266- oder ESP-32-Board mit einem Preis unter 2 € kann auf dieselbe Weise für die ZigBee Home-Automation vorbereitet werden. Die vielen Ein-Ausgabeanschlüsse dieser Microcontroller laden zum Basteln ein. Über die Erfahrungen damit sind wie erwähnt weitere Beiträge geplant.

# Grundlagen der Informatik—Inhalt

Christian Zahler

1 172 Trends in der IT	12 173 Plotter
2 173 Historische Entwicklung	13 174 Datenstrukturen
3 171 PC-Architektur	14 174 Software
4 171 Netzteil/Akku	15 176 Netzwerk-Grundlagen
5 171 Zentralprozessor (CPU)	16 Internet-Grundlagen
6 172 Interner Speicher	17 177 Cloud Computing
7 173 Schnittstellen	18 Betriebsinformatik
8 171 Externe Speicher	19 173 Multimedia
9 173 Eingabegeräte	20 176 Informatik-Recht
10 174 Bildschirm	21 172 Herstellung von ICs
11 174 Drucker	22 172 Quellen



# Cloud Computing

Christian Zahler

Allgemein versteht man unter der Cloud (engl. *cloud* = Wolke) Dienste, die auf Web-Basis (Protokolle HTTP, HTTPS) angeboten und genutzt werden. Oft unterscheidet man zwischen **Public Cloud**-Dienste (öffentliche Dienste), die über das Internet genutzt werden, und **Private Cloud**-Dienste, die innerhalb von Firmen (Intranet) angeboten werden.

## Cloud-Begriffe

Man unterscheidet heute drei Ebenen der Cloud Services: **SaaS**, **PaaS** und **IaaS** (siehe nebenstehende Tabelle).

**Software as a Service (SaaS):** Dies ist der höchste Abstraktions-Level. Ein SaaS-Anbieter stellt eine fertige Anwendung über das Internet zur Verfügung. SaaS-Anwendungen können meist unmittelbar genutzt werden, viele Anbieter ermöglichen aber auch mehr oder weniger umfangreiche Anpassungen ihrer Systeme.

**Platform as a Service (PaaS):** PaaS erfüllt die Anforderungen derjenigen, die benutzerdefinierte Anwendungen als Dienste erstellen und ausführen möchten. Die zugrundeliegende Infrastruktur (Rechner, VMs, Storage und Netzwerk) wird dabei hinter Services und Schnittstellen für Entwickler versteckt, damit sich diese voll und ganz auf die Anwendungsentwicklung konzentrieren können.

**Infrastructure as a Service (IaaS):** IaaS-Anbieter stellen Rechen-, Speicher- und Netzwerkinfrastruktur (Firewalls, Load-Balancer etc.) zur Verfügung. Typischerweise virtuelle Maschinen, für deren Inhalte (Software) man selbst verantwortlich ist. IaaS ähnelt dem traditionellen Hosting, bei dem Unternehmen die Hostumgebung als logische Erweiterung des firmeninternen Datacenters nutzen, ihre Server dort jedoch genauso warten müssen, als handelte es sich um firmeninterne Server.

## Microsoft Cloud-Produkte

Microsoft betreibt bereits seit 1995 Cloud Services (die damals noch nicht so hießen). Begonnen wurde mit dem Hotmail-Dienst, 1998 folgte die Einführung der Windows Update Services, die heute den größten Clouddienst darstellen.

Das wichtigste Cloud-Produkt ist Office 365, ein SaaS-Dienst für Unternehmen und Bildungseinrichtungen jeder Größenordnung.

Zusätzlich gibt es eine ganze Reihe weiterer Cloud-Produkte, die in Zukunft noch weiter ergänzt werden:

Dienste	Angebote	Zielgruppe
<b>SaaS</b> <b>Software as a Service</b>	<ul style="list-style-type: none"><li>• Microsoft Office 365</li><li>• Microsoft OneDrive, Outlook.com (ehemals Hotmail)</li><li>• Google Apps, Google Docs</li><li>• Dropbox</li><li>• CRM Online (SalesForce, WebEx)</li></ul>	Institutionen jeder Größe, Privatpersonen, Selbstständige
<b>PaaS</b> <b>Platform as a Service</b>	<ul style="list-style-type: none"><li>• Microsoft SQL Azure</li><li>• Google App Engine</li><li>• Force.com</li></ul>	IT-Planer, Softwareentwickler
<b>IaaS</b> <b>Infrastructure as a Service</b>	<ul style="list-style-type: none"><li>• Microsoft Windows Azure Platform</li><li>• Sun Cloud</li><li>• HP Cloud Enabling Computing</li><li>• AppNexus</li></ul>	IT-Abteilungen, IT-Dienstleister

**Microsoft Windows Intune** ist ein System Management Tool, das es IT-Verantwortlichen erlaubt, PCs einer Einrichtung über eine einfache, webbasierte Konsole von jedem Ort aus zu verwalten und zu sichern, neue Software zu installieren oder bestehende Anwendungen zu ergänzen.



Mit Windows Intune erhalten IT-Administratoren einen umfassenden Überblick über den Zustand jedes einzelnen Rechners einer Institution – egal wo sich dieser PC oder der Administrator befindet. Microsoft Intune geht dabei über die bisherigen Möglichkeiten des Microsoft System Center hinaus und erlaubt die Festlegung institutionsweiter Update-Regeln für installierte Software. Service Packs können zentral gesteuert und installiert werden.

**Windows Azure:** „Cloud-Betriebssystem“ von Microsoft. Windows Azure stellt eine Plattform für hochverfügbare, skalierbare Webanwendungen dar, ist daher in erster Linie für Programmierer interessant. Die Plattform bietet auch Storage (der auf einen lokalen Server gemountet wird und



wie ein NTFS-Laufwerk agiert), Online Backup, Benutzer-authentifizierung, Virtualisierung und weitere Möglichkeiten.

**SQL Azure:** Hochverfügbarer, skalierbarer Datenbankdienst auf der Basis von SQL Server-Technologien. SQL Azure erlaubt das Erstellen von Datenbanken in den Rechenzentren von Microsoft. Die



Verwaltung erfolgt webbasiert bzw. mit den bekannten Tools von SQL Server; das Abrufen der Daten kann mit TSQL-Befehlen erfolgen.

## Wo liegen die Daten, wie sicher sind sie?

Die Daten liegen in Rechenzentren, die vom jeweiligen Cloud-Dienstanbieter betrieben und gewartet werden.

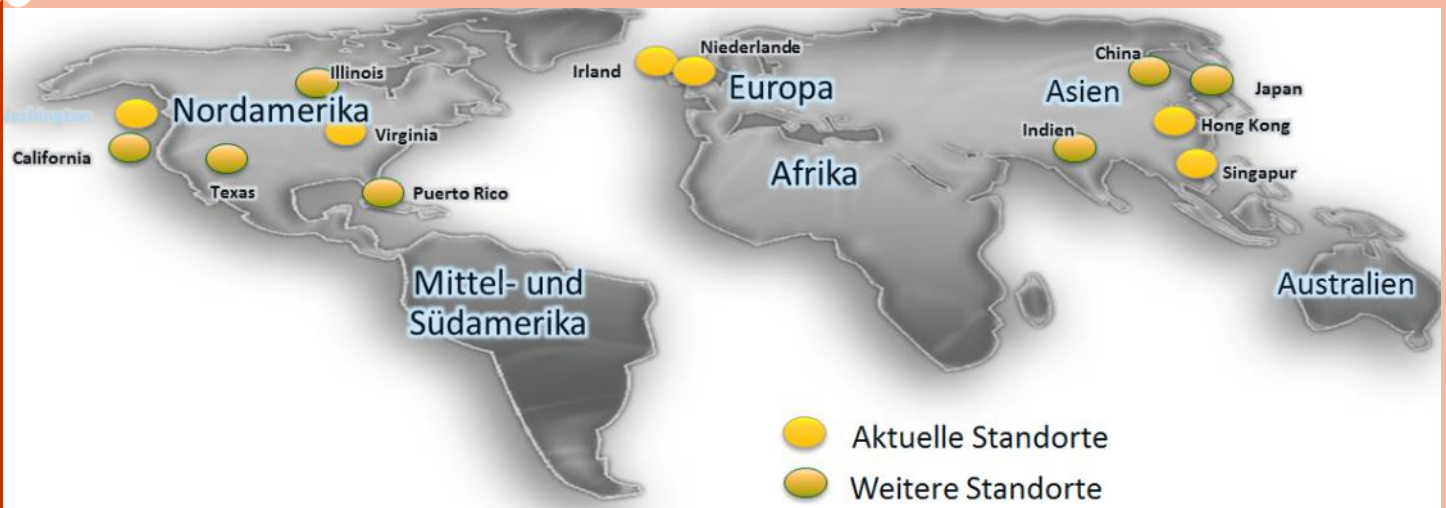
Am Beispiel Microsoft soll dieses Konzept etwas näher vorgestellt werden.

Microsoft betreibt pro Kontinent bzw. Teilkontinent mindestens zwei Rechenzentren, die sich gegenseitig absichern, für den reibungslosen Betrieb sorgen.

Anhand des Standortes wird den Office 365-Benutzern ein bestimmtes Rechenzentrum zugewiesen. Benutzer in Europa verwenden beispielsweise als primäres Rechenzentrum Dublin (Irland), als Backup-Rechenzentrum wird Amsterdam (Niederlande) verwendet.

Microsoft hat bisher mehr als 2,4 Milliarden USD in den Aufbau seiner Rechenzentren investiert.

Alle Rechenzentren sind durch die Firma CyberTrust sowie nach ISO 27001 zertifiziert.



#### Grafik: Microsoft

Die folgenden Fotos zeigen das Microsoft-Rechenzentrum in Dublin, Irland. Alle Fotos auf dieser Seite sind © Microsoft.

Innerhalb der Rechenzentren sorgen redundante Systeme für Stabilität und Zuverlässigkeit.

In den Rechenzentren wurden neun Datensicherheits-Ebenen umgesetzt:

- Router mit Filterung
- Firewalls
- Intrusion Detection Systeme (IDS, zur Erkennung von softwareseitigen Einbruchs- bzw. Eindringversuchen)
- Sicherheit auf Betriebssystemebene
- Authentifizierung der Anwendungen
- Gegenmaßnahmen auf Anwendungsebene
- Virens Scanner
- Abgetrennte Datennetzwerke
- Authentifizierung gegenüber den Daten selbst

Die Datenkommunikation erfolgt grundsätzlich verschlüsselt über SSL.

Die Betriebsprozesse sind an etablierten Konzepten (ITIL/MOF) ausgerichtet und werden laufend überprüft.

Bei der Verwendung von Cloud-Diensten, bei denen Daten zwischen lokalen Netzwerken bzw. einzelnen Rechnern und den Rechenzentren eines Anbieters ausgetauscht werden und auch in diesen Rechenzentren gespeichert werden, stellt sich als zentrale Frage die Umsetzung von Datensicherheit und Datenschutz.

Viele Anbieter werben daher mit internationalen Zertifizierungen wie **ISO 27001**. Diese internationale Norm legt strenge physische, Prozess- und Verwaltungskontrollen fest.

Die Sicherheit der Daten wird auch gewährleistet durch:

- geographisch verteilte und sich gegen-



seitig absichernde Rechenzentren

- die verschlüsselte Datenkommunikation über SSL und
- die garantierte Verfügbarkeit von 99,9%.

Der Schutz gegen Datenmissbrauch wird durch Erfüllung verschiedener Richtlinien gewährleistet:

- EU-Datenschutzrichtlinie 95/46/EG
- EU Safe-Harbor-Bestimmungen



- **Standardvertragsklauseln der europäischen Union („EU Model Clauses“)**: Die EU-Standardvertragsklauseln in Verbindung mit einer erweiterten Auftragsdatenverarbeitungserklärung (ADV) regeln den Datenzugriff auf EU-Datenschutz-niveau, auch wenn von außerhalb der EU, z.B. im Supportfall, zugegriffen wird.



# CO2-reduzierter Beton

Günter Hartl

Es musste ja so kommen. Jetzt werden die klimaneutral hergestellten Baumaterialien nur noch zwischen den Vogelbrutzeiten verbaut. Auf solch eine Lobby können baulärmgeplagte Nachbarn jedoch selten zurückgreifen. Oder gibt es auch Bauphasenstopps, wenn Schwangere in der Nachbarschaft ansässig sind? Keine Ahnung! (Bilder rechts)

Jedoch spiegelt diese Anekdote nur den heutigen Zeitgeist wider. Der nächste Tischreibeißer betrifft die galoppierenden Preisentwicklungen in Österreich. Speziell bei Drogerieartikeln sind die Unterschiede zu Deutschland oftmals nur mit einem begleitenden Tourette-Syndrom ertragbar.

Gut, wenn man in Salzburg wohnt, kann dieses Problem mit einer kurzen Autofahrt über die Grenze gelöst werden. Die anderen kaufen meist über Amazon oder eBay ihre Hygieneartikel oder lassen sich gleich das Zeug per Post aus Deutschland schicken. Vieles wird ja nicht nach Österreich geschickt, weshalb eine Lieferadresse in Deutschland von Vorteil ist. Verschiedene Firmen bieten diese schon kostenlos an, wobei nur die Lieferung nach Österreich und eine überschaubare Gebühr fällig werden.

Auf GitHub hat jemand eine Chrome-Extension zusammengeschustert, um die Preisunterschiede gleich abchecken zu können. Im Webstore von Chrome würde das nicht auf so viel Gegenliebe treffen. Muss man im Entwicklermodus installieren, funktioniert aber. Egal, ob Drogeriemarkt oder Möbelhaus – die Preisunterschiede zwischen den Piefkes und Schluchtenscheißern sprechen für sich. Vergleichen lohnt sich hier definitiv. Klonk. GitHub - gomadoge/price-checker-extension (Bilder rechts unten)

Tja, und die Wiener Zeitung hat nun auch nach jahrhundertelanger Tradition das Zeitliche gesegnet. Genauso wie das Gasthaus »Altes Jägerhaus« neben dem Lusthaus.

Dafür können zukünftige Journalisten gleich im Bundeskanzleramt auf Linie gebracht werden. Mittlerweile dürfte es auch bis zum Schottengymnasium durch-



gedungen sein, dass große Konzerne und Bundeskanzlerämter schon mehr PR-Journalisten beschäftigen als mittelgroße Verlage.

Unsere PCNEWS ist somit die letzte Bastion gegen diese despektierlichen Entwicklungen – weder KI noch schleimig angebotene parteinahe Wochenendtrips für Autoren werden unserer analogen Brand-schutzmauer etwas anhaben können. Das St. Pöltner Gewerbegebiet hat auch seinen Reiz und vor allem keine Vogelbrutgebiete.

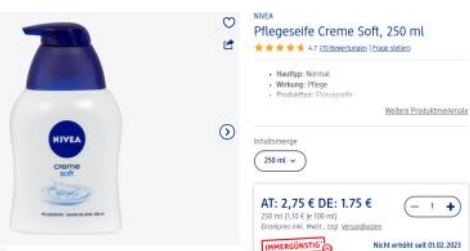
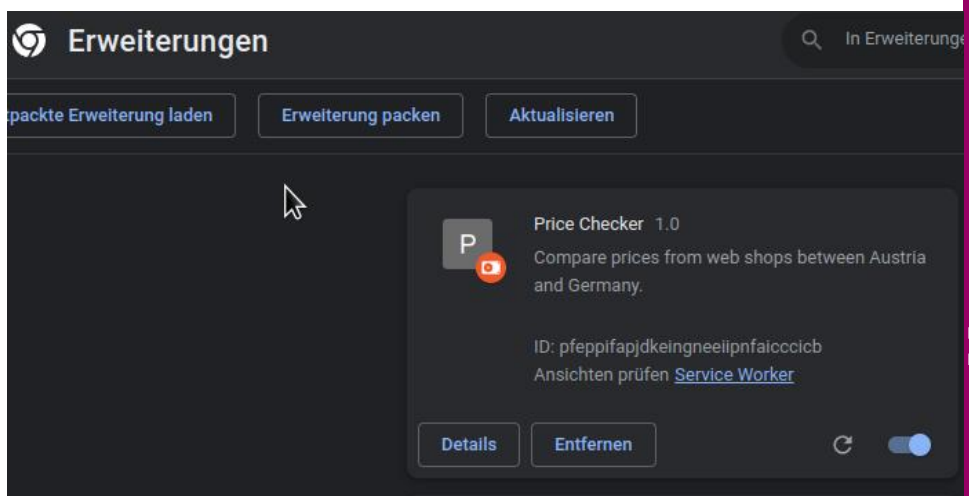
Gleichwohl treffen uns die Preissteigerungen ebenso, weswegen wir bei unseren Sparmaßnahmen noch kreativer werden mussten. Mit einem selbstgebackenen Blauensteiner-Kuchen für meinen Chefredakteur hätten wir die Personalkosten zumindest deutlich eindämmen können. Aber mein moralischer Kompass korrelierte nicht mit diesem unreinen Gedanken – und so beschränkten wir stattdessen die PCNEWS auf 24 Seiten.

Diese CO2 reduzierte PCNEWS ist somit nicht nur klimaneutral, sondern ein unabhängiges Bollwerk des Printwesens. Den Verlust der Wiener Zeitung und des Playboy haben wir zwar überlebt, die härtere Nuss wird aber der umgreifende Fachkräftemangel in Regierungskreisen werden.

Apropos: Sind die von der Kärntner Landesregierung inzwischen gebrieft worden, dass monatliche Windows-Updates doch keine Backups sind?

Wie auch immer, Gott gib uns Kraft und der Druckerpresse genug Strom. Auf dass unsere Zeitung weiterhin resilient bleibt.

Man liest sich, Gruß Günter



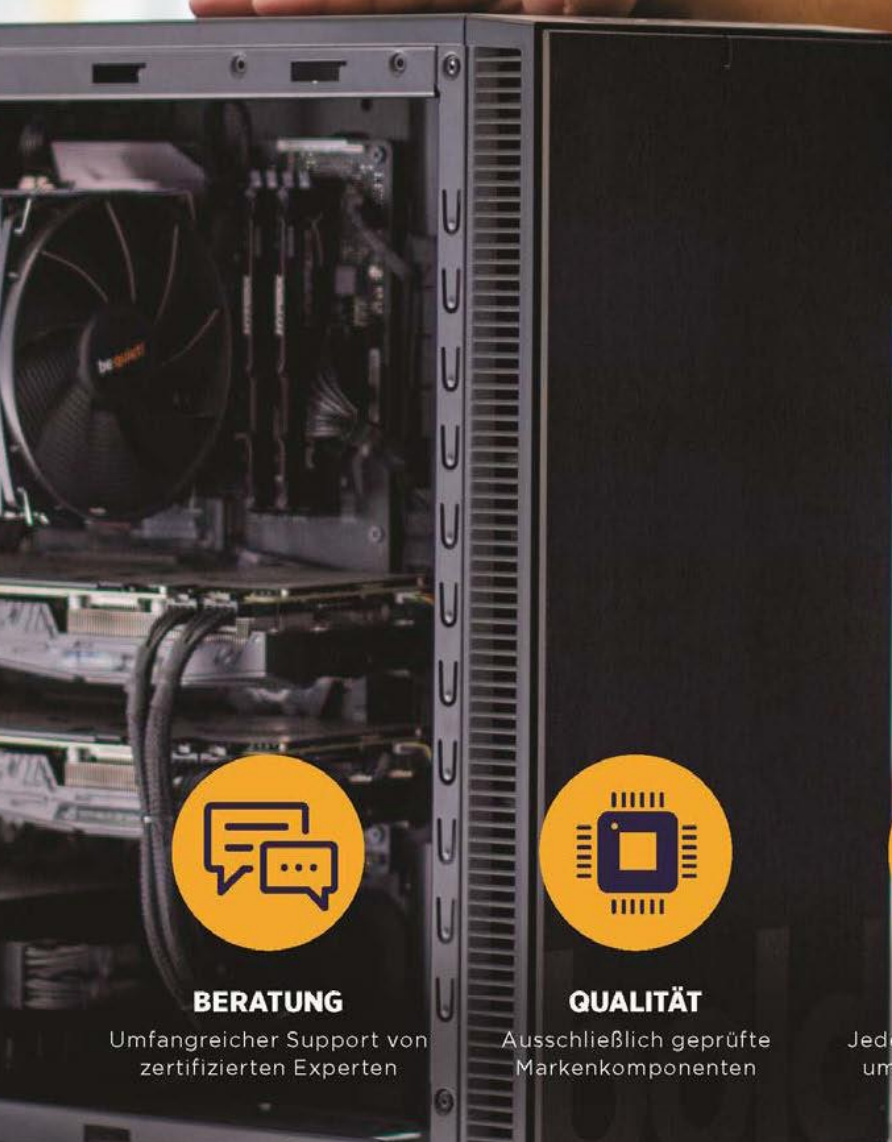


# techbold

# WIR BAUEN DEINEN PC

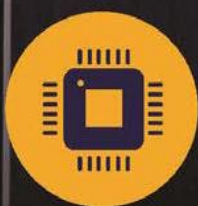
Nutze die langjährige Erfahrung der techbold Computer Experten für die perfekte Konfiguration deines PC-Systems. Egal ob Gaming Maschine, Office-PC oder Workstations für professionelle Anwendungen wie CAD, 3D Grafik und Videoschnitt – wir erstellen dir ein Angebot mit dem perfekten Preis-Leistungs-Verhältnis.

[www.techbold.at/pc-zusammenstellen](http://www.techbold.at/pc-zusammenstellen)



## BERATUNG

Umfangreicher Support von  
zertifizierten Experten



## QUALITÄT

Ausschließlich geprüfte  
Markenkomponenten



## TESTS

Jede Konfiguration wird  
umfangreich getestet



## GARANTIE

3 Jahre Garantie auf alle  
individuellen PC-Systeme