

Künstliche Intelligenz und Maschinelles Lernen zur Lösung ungelöster Probleme in der Mathematik*

Prof. Dr. V. David Sánchez A., Ph.D.
Brilliant Brains, Palo Alto, California
März 2025

Zusammenfassung

Mit Hilfe von ungelösten und gelösten Problemen der reinen und der angewandten Mathematik, z.B. jeweils der abstrakten Algebra und der numerischen Optimierung, wird der Einsatz von Methoden der Künstlichen Intelligenz und des Maschinellen Lernens (KI/ML) in der Mathematik grundlegend untersucht. Die kombinatorische Gruppentheorie wurde zu Beginn des 20. Jahrhunderts mit den grundlegenden Arbeiten von Max Dehn [4] entscheidend geprägt, einschl. die drei Dehnschen Probleme: das Wortproblem, das Konjugationsproblem und das Isomorphieproblem, die alle nur semi-entscheidbar und im Allgemeinen algorithmisch unlösbar sind. Er löste auch als erster eines, das dritte, von Hilberts 23 mathematischen Problemen. In den frühen 1880er Jahren hatte bereits Walther von Dyck [5] die Grundlagen eingeführt indem er als erster, Gruppen anhand von Erzeugern und Relationen systematisch analysierte. Eine einführende Behandlung über die Geometrie infiniter Gruppen kann z.B. in [6] nachgeschlagen werden. Die Theorie der Gruppenpräsentationen und der kombinatorischen Gruppentheorie werden jeweils in [7] und [8] behandelt. Gruppenpräsentationen anhand von Erzeugern und Relationen werden in [9] ausführlich beschrieben. Die Literaturhinweise sind nur exemplarisch und keineswegs erschöpfend. Abbildung 1(a), (b) und (c) zeigen jeweils die Definitionen von Gruppe, Erzeugern und Cayley-Graph.

In der Gruppentheorie, erlauben freie Gruppen, jede Gruppe durch Erzeuger und Relationen darzustellen. In der algebraischen Topologie treten sie als Fundamentalgruppe von Graphen. Eine Gruppe G heißt frei, wenn sie eine Teilmenge S enthält, sodass jedes Gruppenelement $g \in G$ auf genau eine Weise als reduziertes Wort von Elementen in S und deren Inversen geschrieben werden kann. Abbildung 2(a) zeigt die geometrische Definition einer freien Gruppe. Abbildung 2(b) zeigt die Worte, reduzierte Worte und die Menge der äquivalenten Worte. Schließlich, Abbildung 2(c) zeigt die Menge der äquivalenten Worte mit der Konkatenation als freie Gruppe. Der topologische Begriff des Geschlechts (g) einer kompakten orientierbaren Fläche ist eine ganze Zahl und gibt die maximale Anzahl von möglichen Schnitten entlang disjunkter, einfach geschlossener Kurven, so dass die

*Diese Zusammenfassung wurde zur Veröffentlichung freigegeben. Der Verfasser ist weltweit der jüngste Wissenschaftler, der mit dem IEEE Fellow Prize („Nobelpreis“ für Ingenieurwissenschaften) ausgezeichnet wurde [1] mit der Erwähnung „für Führung auf dem Gebiet der neuronalen und parallelen Rechenmethoden und bahnbrechende Beiträge zu autonomen Weltraumrobotern“. Er gründete und leitete als Chefwissenschaftler, EiC 15 Jahre lang, eine wissenschaftliche Zeitschrift zu KI/ML, die von Elsevier Science herausgegeben wurde [2]. Als zehnjähriger Wissenschaftsmitarbeiter der Bundesregierung beim Deutschen Zentrum für Luft- und Raumfahrt (DLR) startete er 1988 mit dem Büro des Bundesministers für Forschung und Technologie das erste Bundesprogramm für KI/ML-Forschung und Technologieentwicklung in Deutschland [3], das sich mit seinem Konsortium, zu dem DLR und Siemens Zentral-FuE in München, Deutschland, gehörten, auf Lernrobotik und Automatisierung konzentrierte. Er hat mehrfach weltweit als erster hochkomplexe Probleme der Mathematik, u.a. der abstrakten Algebra und der multivariaten Approximationstheorie mittels Künstliche Intelligenz und Maschinelles Lernen gelöst. Die Ergebnisse wurden dann z.T. im Anwendungsbereich der Halbleitertechnologie, der Robotik und der Raumfahrt erfolgreich eingesetzt, so z.B. in einem geheimen Programm des U.S. Verteidigungsministeriums mit Unterauftragnehmern Lockheed Martin und SAIC unter dem Einsatz von Neurochips. Zuletzt hat er der NASA Jet Propulsion Laboratory (JPL) und den Weltraumstreitkräften der Vereinigten Staaten (auf Englisch: United States Space Force) in Kalifornien unterstützt, keine sicherheitsvertrauten Details sind hier enthalten.

Definition of a Group

A group G is defined as a triple $(G, \times, (-1))$, where:

- o G is a set;
- o $\times : G \times G \rightarrow G$ is a binary operation (multiplication);
- o $(-1) : G \rightarrow G$ is an unary operation (inversion).

Multiplication must satisfy the following axioms:

- o Associativity: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$;
- o Existence of an identity element $e \in G$:

$$e \cdot g = g \cdot e = g, \quad \forall g \in G;$$

- o Existence of an inverse element for each $g \in G$:

$$\exists g^{-1} \in G : g \cdot g^{-1} = g^{-1} \cdot g = e.$$

(a)

Generators

Let G be a group and $\mathcal{X} \subseteq G$. The set \mathcal{X} is said to be a *generating set* of the group G

$$G = \langle \mathcal{X} \rangle,$$

if any element $g \in G$ can be represented as a finite product of elements from \mathcal{X} and their inverses:

$$\forall g \in G, \exists \{x_i\}_{i=1}^N, x_i \in \mathcal{X} \cup \mathcal{X}^{-1}, \text{ such that } g = x_1 x_2 \cdots x_N.$$

A finite sequence of elements from the set $\mathcal{X} \cup \mathcal{X}^{-1}$ is called a *word* and denoted by ω :

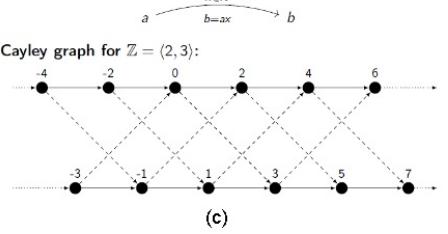
$$\omega(\mathcal{X}) = \omega(x_1, x_2, \dots, x_N).$$

(b)

Cayley graph

Let G be a group and \mathcal{X} be a generating set. Then the *Cayley graph* $Cayley(G; \mathcal{X})$ is defined as follows:

- Vertices $V = G$.
- An edge $(a, b) \in E$ exists iff $a \cdot x = b$ for some $x \in \mathcal{X}$. We will say that this edge is labelled by x .



(c)

Abbildung 1: Gruppentheorie [10] (a) Gruppe (b) Erzeuger (c) Cayley-Graph

Geometric Definition of a Free Group

Definition

A group is called *free* if its Cayley graph (for some generating set) is a tree.

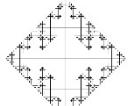


Figure: Free group with 2 generators

(a)

Word Space

Let's consider a finite symmetrized alphabet \mathcal{X} , i.e.,

$$x \in \mathcal{X} \Rightarrow x^{-1} \in \mathcal{X}.$$

We will identify words of the following form (cancellation of mutually inverse letters):

$$x_1 x_2 \cdots x_k x_k^{-1} x_{k+1} \cdots x_n \sim x_1 x_2 \cdots x_k^{-1} x_k x_{k+1} \cdots x_n \sim \\ \sim x_1 \cdots x_{n-1} x_n.$$

If no cancellation can be performed in a word, we call such a form of the word *reduced*. The number of letters in the reduced form is called the *length* of the word.

The reduced form is unique.

The set of equivalent words is denoted by $W(\mathcal{X})$.

(b)

Free Group

Define the operation $\#$, concatenation, on words ω_1, ω_2 as appending of words. That is, for

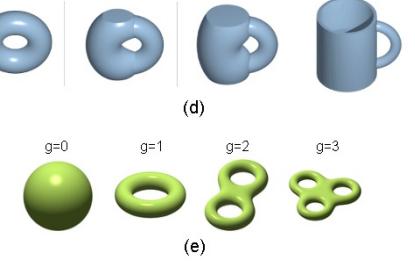
$$\omega_i = x_1^i \cdots x_{n_i}^i, \quad i = 1, 2;$$

Let

$$\omega_1 \cdot \omega_2 := x_1^1 \cdots x_{n_1}^1 \cdot x_1^2 \cdots x_{n_2}^2.$$

Definition

The set $W(\mathcal{X})$ with the concatenation operation is a group, called the *free group* $F(\mathcal{X}) := (W(\mathcal{X}), \cdot)$.



(c)

Abbildung 2: Freie Gruppen [10] und Henkelkörper (a) Freie Gruppe – geometrische Definition (b) Wortraum (c) Die Menge der äquivalenten Worte mit der Konkatenation als freie Gruppe (d) Flächengeschlecht $g=1$: von Donut zu Kaffetasse (e) Henkelkörper vom Geschlecht g : Kugel ($g=0$), Torus ($g=1$), Doppeltorus ($g=2$)), Brezel ($g=3$)

Fläche nach dem Schnittvorgang immer noch zusammenhängend ist. Umgangssprachlich, gibt das Geschlecht einer Fläche die Anzahl der Löcher oder der Henkel der Fläche an. Deshalb haben beide, eine Kaffeetasse und ein Donut, dasselbe Geschlecht $g=1$, vgl. Abbildung 2(d). Weitere Beispiele folgen in Abbildung 2(e). Die Henkelzerlegung bildet in der Differentialtopologie die Grundlage für die Klassifikation und Beschreibung von Mannigfaltigkeiten.

Im Bereich der Booleschen Algebra, bei der Minimierung Boolescher Funktionen, wurde das Quine–McCluskey Verfahren, das von Willard V. Quine in [11] entwickelt und von Edward J. McCluskey Jr. in [12] erweitert wurde, durch die Einführung von Heuristiken erfolgreich übertroffen, so z.B. durch die Algorithmen in [13], die wesentlich effizienter sind und den Speicherbedarf sowie die Rechenzeit um mehrere Größenordnungen reduzieren, womit ein wichtiger Fortschritt in der automatischen Logiksynthese erzielt wurde. Das globale Minimum wird dabei lediglich approximiert, lässt sich nicht garantieren, hat sich aber in der Praxis als eine gute Approximation bewährt. Die hardwaretechnische Realisierung der dazugehörigen digitalen Schaltung wird nach der Minimierung der Booleschen Funktion $f : \{0, 1\}^n \rightarrow \{0, 1\}$ einfacher und daher kostengünstiger. Das Quine–McCluskey Verfahren geht von der Funktionsdarstellung in kanonischer disjunktiver Normalform (KDNF) aus. Wenn man z.B. die Boolesche Funktion ($n = 3$): $f(a, b, c) = abc + \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}\bar{b}\bar{c}$ mit dem Quine–McCluskey Verfahren minimiert, erhält man $f(a, b, c) = \bar{a}b + c$. Abbildung 3(a) zeigt für die o.g. Beispielfunktion das Quine–McCluskey Verfahren mit dem Hasse-Diagramm des Suchgraphen. Abbildung 3(b) zeigt die Minimierung eines Beispiels mit einer Booleschen Funktion $f(a, b, c)$ ausgehend aus ihrer Wahrheitstabelle mit Don't Cares ('d' statt '0' oder '1'). Die Gütehierarchie der Überdeckungen der zu minimierenden Funktion wird auch unten gezeigt: von nicht-Prim-, Prim-, irredundante und minimale Überdeckung(en). Ein Primterm oder Primimplikant ist ein Implikant minimaler Länge, der also durch weiteres Vereinfachen oder Weglassen von Variablen kein Implikant mehr wäre. Ein Kernprimterm oder Kernprimimplikant ist ein Primterm, die Minterme enthält, die in keinem anderen Primterm vorkommen. Er muß in jeder minimalen disjunktiven Normalform vorkommen. Das Quine–McCluskey Verfahren bestimmt das Minimum als eine Überdeckung be-

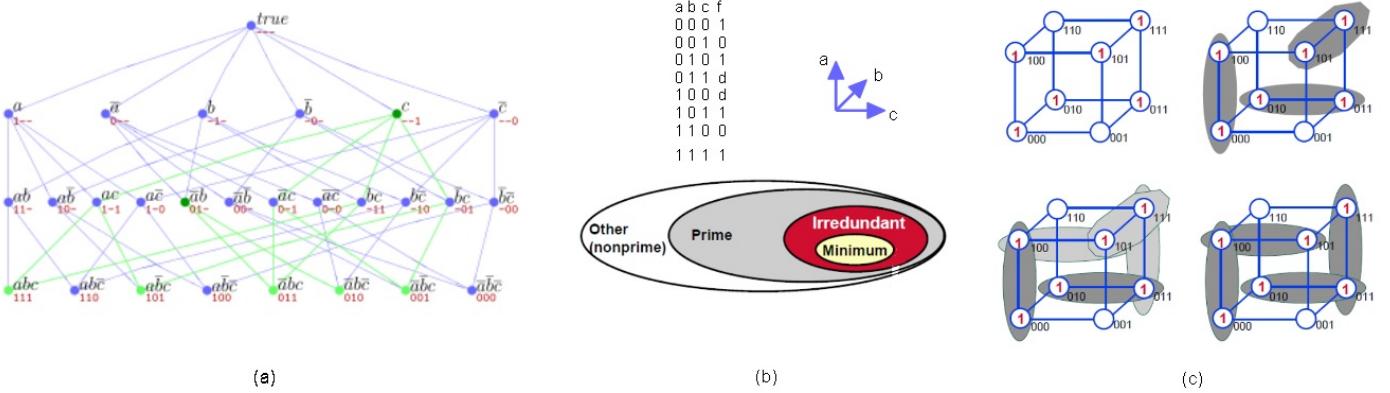


Abbildung 3: Logische Minimierung Boolescher Funktionen (a) Quine–McCluskey Verfahren – Beispiel – Hasse-Diagramm des Suchgraphen (b) Wahrheitstabelle mit Don't Cares ('d' statt '0' oder '1') der Booleschen Funktion $f(a, b, c)$ [CMU] (c) Überdeckungen der Booleschen Funktion $f(a, b, c)$

stehend aus Kernprimimplikanten in KDNF. Abbildung 3(c) zeigt verschiedene Überdeckungen der Booleschen Funktion $f(a, b, c)$. Oben links wenn Don't-Care-Werte in '1'-Werte überführt wurden: $f(0, 1, 1) = f(1, 0, 0) = 1$. Oben rechts die Minimum-Überdeckung von f . Unten links eine redundante Überdeckung. Schließlich, unten rechts eine minimale irredundante Überdeckung, aber nicht das Minimum. Da bei der Minimierung Boolescher Funktionen das zugrunde liegende Problem NP-vollständig ist, gibt es kein effizientes Verfahren. Die Rechenzeit wächst exponentiell mit der Anzahl der Variablen n . Allein die Anzahl der Primimplikanten für eine Funktion mit n Variablen kann so groß sein wie $\frac{3^n}{\sqrt{n}}$ [14]. Deshalb wurden KI-Heuristische Verfahren zur logischen Minimierung für die VLSI-Synthese, so z.B. von mir am Karlsruher Institut für Technologie (KIT) [15] entwickelt. Die Algorithmen waren im mathematischen Sinne beweisbar und ihre Implementierung basierte auf rekursive Funktionen. Als ich zur Entwicklungsbereitung von Siemens AG Automatisierung wechselte, benutzten wir im Entwicklungsteam mein KI-für-VLSI-Programm, um ASICs als Bestandteile von Spezialprozessoren zu entwerfen, die wir in der Gruppe entwickelt haben. Danach habe ich auch das dazu passende Echtzeitbetriebssystem für Multiprozessorsysteme geschrieben. Anschließend, in 1985, habe ich mit einer Gruppe von etwa 15 Technologen das deutsche VLSI-für-KI-Komitee in Berlin gegründet, um auch die Entwicklung von KI-Chips voranzutreiben.

Im Bereich des Maschinellen Lernens hat man über die Jahre verschiedene Lernmethodentypen eingesetzt. Zunächst, wurden überwachte Lernmethoden (auf Englisch: Supervised Learning methods) eingesetzt, um Ausgangsdaten auf der Basis von vorher ungesehenen Eingangsdaten zu generieren nachdem Trainings-Eingangs- und -Ausgangsdaten mit bekannten Bezeichnungen verwendet wurden, um ein Ein-Ausgangs-Modell zu bilden. Unüberwachte Lernmethoden (auf Englisch: Unsupervised Learning methods) halfen, Muster in Daten ohne bekannte Bezeichnungen aufzudecken. Beide Lernmethodentypen, überwacht und unüberwacht, waren im Prinzip ungeeignet, Sequenzen von Daten zu erlernen, und ihr Einsatz in Bereichen wie in der Robotik begrenzt [16]. Fokussierte Erweiterungen der neuronalen Netzwerkarchitekturen habe ich dann eingeführt [17], so daß auch Datensequenzen erlernt werden konnten. Sie basierten auf dem Konzept der Aufmerksamkeit (auf Englisch: Attention). Eine Demonstration dieser Fähigkeit folgt. Abbildung 4(a) zeigt Attraktoren für verschiedene Werte des Parameters τ der Mackey–Glass-Gleichungen [18], eine Familie von nichtlinearen Differentialgleichungen mit Zeitverzug. Das entspricht der Visualisierung von $(P(t), P(t-\tau))$. Sie werden wie folgt definiert (hier nur eine von zwei Gleichungen): $\frac{dP(t)}{dt} = \frac{\beta_0 \theta^n P(t-\tau)}{\theta^n + P(t-\tau)^n} - \gamma P(t)$. Sie werden zur Modellierung physiologischer Systeme eingesetzt und zählen zu einem der beliebtesten Standard-Benchmark-Datensätze für solche Aufgaben, d.h. die Prädiktion komplexer Zeitreihen. Abbildung 4(b) zeigt oben und in der Mitte jeweils eine gesunde ($\tau = 6$) und eine pathologische ($\tau = 20$) Veränderung der Blutzelldichte. Ganz unten wird die Mackey–Glass-Zeitreihe und ihre Parameterwerte gezeigt, die zum Bestätigen des erfolgreichen Aufmerksamkeitsmechanismus in der neuronalen Netzwerkarchitektur verwendet wurden. Abbildung 4(c) zeigt jeweils oben und

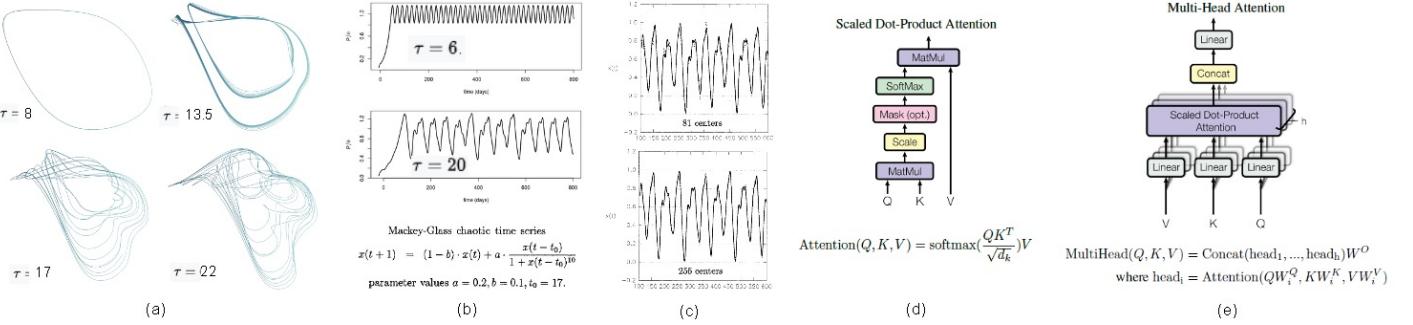


Abbildung 4: Zeitreihenanalyse – Aufmerksamkeit in neuronalen Netzwerkarchitekturen (a) Attraktoren der Mackey-Glass-Gleichungen (b) Mackey-Glass-Zeitreihen (c) Ergebnisse – Erlernen von Datensequenzen mit neuronalen Netzwerkarchitekturen mit Aufmerksamkeit [17] (d) Transformer-Aufmerksamkeit: das skalierte Skalarprodukt (e) die Mehrkopf-Aufmerksamkeits-Funktion [19]

unten die Ergebnisse der neuronalen Netzwerkarchitektur mit jeweils 81 und 256 Neuronen in der Zwischenschicht eines RBF-Netzwerkes ohne wesentliche Unterschiede. Abbildung 4(d) zeigt den Aufmerksamkeitsmechanismus der Transformerarchitektur, der die Aufmerksamkeitsgewichte zwischen verschiedenen Elementen in einer Sequenz bestimmt. Schließlich, Abbildung 4(e) zeigt die Mehrkopf-Aufmerksamkeits-Funktion, die mehrere solche Berechnungen erlaubt, um gleichzeitig verschiedene Aspekte der Daten zu berücksichtigen.

Bei bestärkenden Lernmethoden (auf Englisch: RL = Reinforcement Learning) [20] erkundet ein Agent die Umgebung, um die optimale Strategie herauszufinden. Durch das Konzept der Markow-Kette [21] wurde die Anwendung des Gesetzes der großen Zahlen von Folgen unabhängiger Zufallsvariablen von Andrei A. Markow auf Folgen abhängiger Zufallsvariablen erweitert, die die Markow-Eigenschaft genügen, eine gedächtnislose Eigenschaft eines stochastischen Prozesses, nach der nur der gegenwärtige Zustand die Wahrscheinlichkeitsverteilung zukünftiger Zustände beeinflusst. Bei Systemmodellierungen eignen sich Markow-Ketten, um zufällige Zustandsänderungen eines Systems zu modellieren. Wenn man eine Markow-Kette um einen Agenten erweitert, der sich zwischen mehreren möglichen Aktionen entscheiden kann und positive oder negative Belohnungen als Rückmeldung erhält, dann hat man einen Markow-Entscheidungsprozess. Abbildung 5(a) zeigt die Definition einer Markow-Kette mithilfe der Markow-Eigenschaft. Abbildung 5(b) zeigt die Definition eines Markow-Entscheidungsprozesses und das Ziel des Agenten, der seine Aktionen mit Hilfe einer Strategie π auswählt. Abbildung 5(c) zeigt ein inverses Pendel montiert auf einem Wagen als Beispiel für einen Markow-Entscheidungsprozess. Ein Markow-Entscheidungsprozess (MEP) [22] ist ein Tupel (S, A, R, P, ρ) wo S, A der Zustandsraum und die Aktionsmenge jeweils sind. $R : S \times A \times S \rightarrow \mathbb{R}$ ist die Belohnungsfunktion die jeder Transition von einem Zustand zum Folgezustand durch eine spezifische Aktion eine reellwertige Belohnung zuordnet. $P : S \times A \rightarrow P(s_1|s_0a_0)$ ist die Transitions-wahrscheinlichkeitsfunktion. Schließlich, $\rho(s) \rightarrow [0, 1]$ ist die Startzustandswahrscheinlichkeitsverteilung, die die Wahrscheinlichkeit eines Systemstarts in einem gewissen Zustand beschreibt. Das System startet im Zustand s_0 mit Wahrscheinlichkeit $\rho(s)$ und führt eine Aktion a_0 aus, das System erreicht dann den Folgezustand s_1 mit Wahrscheinlichkeit $P(s_1|s_0a_0)$. Diese Transition vom Zustand s_0 zum Folgezustand s_1 erhält die Belohnung $r_0 = R(s_0, a_0, s_1)$, die die Wirksamkeit der Aktion im Hinblick auf ihren Beitrag zum Gelangen des Gesamtziels quantifiziert. Die Reihenfolge wiederholt sich vom Zustand s_1 aus und nachfolgend wird die Trajektorie $\tau = \{s_0, a_0, s_1; a_1, \dots\}$ generiert. Das Ziel des Gesamtprozesses ist den Gesamtertrag $R(\tau) = \sum_{t=0}^T \gamma^t R(s_t, a_t, s_{t+1})$ zu maximieren, mit T die Trajektorienlänge, auch Horizontlänge genannt, und $\gamma \in [0, 1]$ der Diskontierungsfaktor, der Belohnungen gewichtet, die kurzfristig erfolgen, höher als solche, die später erfolgen. Ein MEP, bei dem die Anzahl von Zuständen und Aktionen endlich ist und alle Transaktionswahrscheinlichkeiten und Belohnungen bekannt sind, kann mit Hilfe der dynamischen Programmierung [23] iterativ gelöst werden. Das Modell des inversen Pendels montiert auf einem Wagen ist ein Beispiel für einen MEP. Der Zustandsraum kann aus der Horizontallage des Wagens x und dem Winkel θ zwischen dem

Ω : finite state space
 P : transition matrix $|\Omega| \times |\Omega|$

MC

Definition

A sequence of random variables (X_0, X_1, X_2, \dots) is a Markov chain with state space Ω and transition matrix P if for all $n \geq 0$, and all sequences $(x_0, x_1, \dots, x_n, x_{n+1})$, we have that

$$\begin{aligned} \mathbb{P}[X_{n+1} = x_{n+1} | X_0 = x_0, \dots, X_n = x_n] \\ = \mathbb{P}[X_{n+1} = x_{n+1} | X_n = x_n] = P(x_n, x_{n+1}). \end{aligned}$$

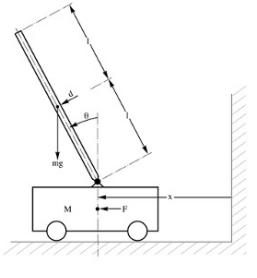
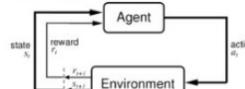
(a)

MDP (S, A, T, R, H)

- Given:
- S : set of states
 - A : set of actions
 - $T: S \times A \times S \times \{0, 1, \dots, H\} \rightarrow [0, 1]$, $T_t(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$
 - $R: S \times A \times S \times \{0, 1, \dots, H\} \rightarrow [-\infty, \infty]$, $R_t(s, a, s') = \text{reward for } (s_{t+1} = s', s_t = s, a_t = a)$
 - H : horizon over which the agent will act
- Goal:
- Find $\pi: S \times \{0, 1, \dots, H\} \rightarrow A$ that maximizes expected sum of rewards, i.e.,

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^H R_t(S_t, A_t, S_{t+1}) | \pi \right]$$

(b)



(c)

Abbildung 5: (a) Markow-Kette (auf Englisch: Markov Chain (MC)) [MIT] (b) Markow-Entscheidungsprozess (auf Englisch: Markov Decision Process (MDP)) [UCB] (c) Beispiel: das inverse Pendel montiert auf einem Wagen

Pendel und der Vertikalalachse bestehen oder aber aus dem Winkel θ und der Winkelgeschwindigkeit $\dot{\theta}$. Der Aktionsraum besteht typischerweise aus der Kraft \vec{F} mit der der Wagen balanciert wird oder aus einem entsprechenden Drehmoment \vec{M} . Die zu definierende Belohnungsfunktion R bietet einen Anreiz dafür, daß das Pendel in die aufrechte Position schwingt, möglicherweise auch mit Strafen für übermäßigen Kontrollaufwand. Die Transitionswahrscheinlichkeitsfunktion P wird mithilfe der Pendelsystemdynamik bestimmt. Auf dieser Grundlage können dann optimale Kontrollstrategien mittels z.B. bestärkende Lernmethoden errechnet werden.

Bei bestärkenden Lernmethoden alternieren sie zwischen dem Sammeln von Daten durch Interaktion mit der Umgebung und der Optimierung einer Ersatzzielfunktion. Strategiegradientenmethoden (auf Englisch: Policy Gradient (PG) methods) optimieren die Zielverlustfunktion $L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t | s_t) \hat{A}_t \right]$. $\hat{\mathbb{E}}_t$ gibt empirisch den Durchschnitt einer begrenzten Anzahl von Daten an, π_θ ist eine stochastische Strategie und \hat{A}_t ist eine Schätzung der Vorteilsfunktion zum Zeitpunkt t , die den relativen Nutzen der Durchführung von der Aktion a im Zustand s unter der Strategie π quantifiziert. Ein Nachteil dieser Methoden ist die kontraproduktive Generierung von großen Strategieaktualisierungen. Bei der Lernmethode Strategieoptimierung der Vertrauensregion (auf Englisch: TRPO = Trust Region Policy Optimization) wird folgendes uneingeschränktes Optimierungsproblem gelöst: $\max_{\theta} \hat{\mathbb{E}}_t \left[r_t(\theta) \hat{A}_t - \beta \cdot KL[\pi_{\theta_{old}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t)] \right]$, dabei stellt $r_t(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)}$ ein Strategieverhältnis dar, β ist ein einstellbarer Hyperparameter und die Kullback-Leibler(KL)-Divergenz [24] ist ein Maß für die Unterschiedlichkeit zweier Wahrscheinlichkeitsverteilungen, die numerisch berechnet werden muß. Eine Verbesserung dieser Lernmethode wurde durch die Lernmethode proximale Strategieoptimierung (auf Englisch: PPO = Proximal Policy Optimization) [25] eingeführt, die die Stabilität und Zuverlässigkeit beibehält, aber einfacher zu implementieren ist, denn sie ohne relativ komplexe KL-Divergenz-Berechnungen auskommt. Die PPO Lernmethode verbessert die Trainingsstabilität des Agenten, indem zu große Strategieaktualisierungen vermieden werden. Die PPO Zielfunktion lautet: $L_t(\theta) = \hat{\mathbb{E}}_t \left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t) \right]$, c_1, c_2 sind einstellbare Hyperparameter. Um die Aktualisierungen auf einen vordefinierten Bereich zu beschränken verwendet man $L_t^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right]$. L_t^{VF} ist der mittlere quadratische Verlust $(V_\theta(s_t) - V_t^{target})^2$. Um die Erkundung des Aktionsraums zu fördern addiert man einen Entropie-Bonus $S[\pi_\theta](s_t)$.

So wie wir bereits einen Fall in der Booleschen Algebra besprochen haben wo durch KI-Heuristiken Lösungen ermöglicht wurden, besprechen wir zunächst noch einen Fall in der abstrakten Algebra wo neue Erkenntnisse zu einem noch nicht bewiesenen Problem mittels maschinellen Lernens gewonnen werden konnten. Auf dem Gebiet der kombinatorischen Gruppentheorie gibt es wie sonst in der Mathematik eine Fülle von ungelösten Problemen, so wurde z.B. die noch nicht bewiesene Andrews–Curtis(AC)-Vermutung in ihrem Beitrag über freie Gruppen und Henkelkörper eingeführt [26]. Eine ausgewogene Präsentation ist eine Präsentation mit derselben Anzahl n an Erzeuger und Relationen: $P = \langle x_1, \dots, x_n | r_1, \dots, r_n \rangle$. Die Andrews–Curtis-Vermutung lautet: irgendeine

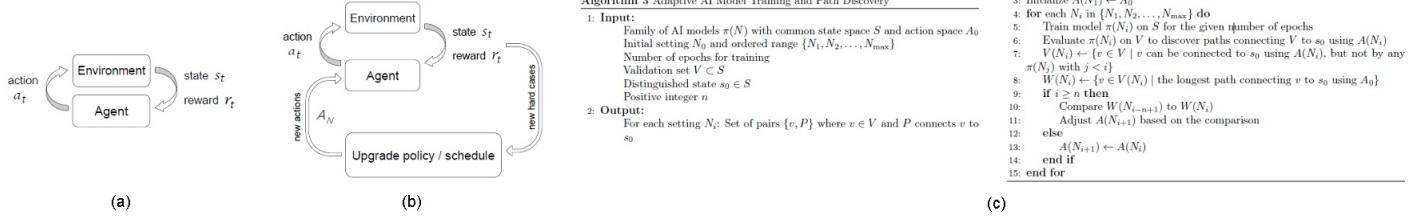


Abbildung 6: Untersuchung der Andrews–Curtis(AC)-Vermutung [28] (a) Zyklus der bestärkenden Lernmethode (b) Erweiterung mit adaptivem Aktionsraum (c) Adaptives Modelltrainieren und Wegfindung

ausgewogene Präsentation der Trivialgruppe $\langle x_1, \dots, x_n \mid r_1, \dots, r_n \rangle$ kann in die Trivialpräsentation $\langle x_1, \dots, x_n \mid x_1, \dots, x_n \rangle$ überführt werden durch eine endliche Folge von Elementartransformationen, auch AC-Züge genannt, und wie folgt definiert: (1) einige r_i durch $r_i r_j, i \neq j$ zu ersetzen, (2) einige r_i mit r_i^{-1} auszutauschen und (3) einige r_i in $g r_i g^{-1}$ zu überführen mit g ein Erzeuger oder seine Inverse. Die Länge einer Präsentation errechnet sich als die Summe der Wortlängen von allen Relationen. Präsentationen die durch eine Folge von AC-Zügen ineinander transformiert werden können nennt man AC-äquivalent. Eine Präsentation heißt AC-trivial wenn sie AC-äquivalent zu einer Trivialpräsentation ist. Um eine bestimmte Präsentation auf Trivialität zu untersuchen kann man systematisch den gesamten Raum möglicher Sequenzen von AC-Zügen absuchen, bis eine Sequenz die Präsentation trivial macht. Dieser Raum wächst aber exponentiell mit der Anzahl an Sequenzen. Die Gesamtzahl der Sequenzen von AC-Zügen der Länge k für eine Präsentation mit n Erzeugern ist $(3n^2)^k$, d.h. ein Brutalkraftansatz ist unpraktikabel. Präsentationen aus der infiniten Akbulut-Kirby(AK)-Reihe [27] möglicher Gegenbeispiele: $AK(n) = \langle x, y \mid x^n = y^{n+1}, xyx = yxy \rangle, n \geq 2$ sind über die Jahre untersucht worden. $AK(2)$ der Länge 11 ist AC-trivial. $AK(3)$ der Länge 13 hat sich aber bis dato nicht trivialisieren lassen. Das der Andrews-Curtis-Vermutung zugrundeliegende Problem kann als Markow-Entscheidungsprozess (MEP) formuliert werden. Präsentationstrivialisierungen mithilfe u.a. von bestärkenden Lernmethoden werden in [28] untersucht. Durchsuchen der Graphenknoten erfolgt zunächst durch Breitensuche und einen Greedy-Suchalgorithmus, der die Zustände in einer Prioritätswarteschlange hält, deren Ordnung durch den Tupel (k, l) bestimmt wird, mit k und l jeweils die Präsentationslänge und die Pfadlänge zwischen dem Zustand und dem Initialzustand. Der Zustand mit dem kleinsten Wert von k wird gewählt. Wenn es mehr Zustände gibt, entscheidet der kleinste Wert von l . Dann wird eine bestärkende Lernmethode, die proximale Strategieoptimierung (auf Englisch: PPO = Proximal Policy Optimization), angewandt, um AC-Trivialisierungen ausgewogener Präsentationen zu finden. Die Ergebnisse werden mit denen verglichen, die mithilfe der o.g. Suchalgorithmen bestimmt wurden. Tiefgreifende Eindrücke konnten gewonnen werden, z.B. beim Testen der Ansatzwirksamkeit mithilfe von Präsentationen aus der Miller-Schupp-Reihe [29] möglicher Gegenbeispiele: $MS(n, \omega) = \langle x, y \mid x^{-1}y^n x = y^{n+1}, x = \omega \rangle$. So z.B. sind für $n = 3$ und $\omega_1 = y^{-1}x^{-1}yxy$ die Präsentationen $MS(n, \omega_1)$ der Länge 15 und $AK(n)$ AC-äquivalent. Die PPO Lernmethode wird eingesetzt, um die Strategiefunktion π zu ermitteln. Wenn aber die Horizontlänge konstant gehalten wird, können komplexe Präsentationen entweder nicht gelöst werden, wenn die Horizontlänge zu klein ist, oder erhebliche Rechenressourcen sind erforderlich, damit das Training konvergiert, wenn die Horizontlänge zu groß ist. Deshalb wurde die Horizontlänge in den Experimenten dann variable gehalten, mit Werten steigend von 200 auf 1200 für Umgebungsinteraktionen von jeweils 10 auf 50 Mio. Der PPO-Agent konnte dann zunächst einfache Präsentationen lösen, um danach mit erhöhter Fähigkeit komplexere Miller-Schupp-Präsentationen mit größerer Horizontlänge zu verarbeiten, einschließlich welche, die die Suchverfahren nicht imstande waren zu lösen. Es empfiehlt sich also das Hinzufügen von zunehmend längeren Sequenzen elementarer AC-Zügen zum Aktionsraum zu ermöglichen. Abbildung 6(a) zeigt den grundlegenden Zyklus der bestärkenden Lernmethode. Abbildung 6(b) zeigt die Erweiterung des Modells mit einem adaptiven Aktionsraum. Abbildung 6(c) zeigt einen Algorithmus zum adaptiven Modelltrainieren und Wegfindung.

Wie hier konkret gezeigt wurde, können manche Mathematik-Probleme eine geeignete Platform darbieten, um selbstverbessernde KI/ML-Systeme auf dem Weg zur Künstlichen Generischen In-

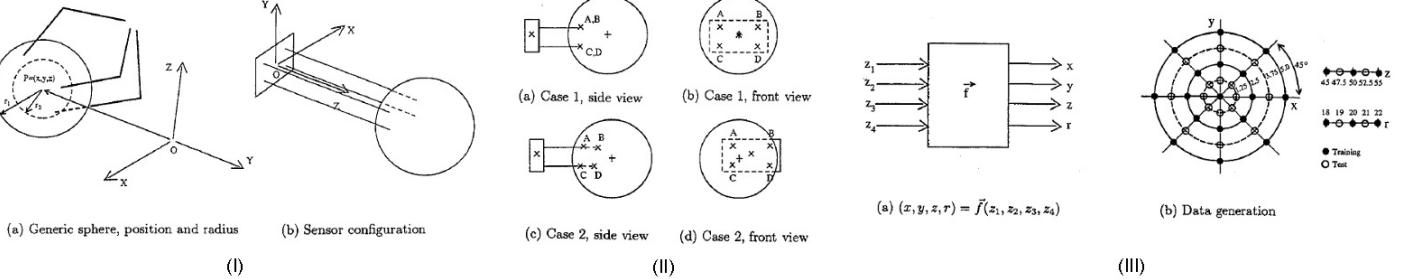


Abbildung 7: Künstliche Generische Intelligenz (auf Englisch: Artificial Generic Intelligence (AGI) – Robotisch autonomes Greifen einer generischen Kugel [17]

telligence (auf Englisch: Artificial Generic Intelligence AGI) zu untersuchen und weiterzuentwickeln, nachdem Große Sprachmodelle (auf Englisch: Large Language Models LLMs) [30] sich mit beinahe menschlicher Kompetenz unterhalten können, die neuesten Modelle schließen u.a. OpenAI GPT4.5 und o3-mini [31] sowie DeepSeek V3 and R1 [32] ein. Grundlagen der LLM Theorie und Praxis können z.B. in [33] nachgeschlagen werden. Wenn wir über die Sprache hinaus zum Verhalten übergehen, d. h. wenn wir KI/ML-Systeme einschließlich intelligente, autonome, lernende Robotersysteme der nächsten Generation [34] entwickeln, dann benötigen wir Große Verhaltenmodelle (auf Englisch: Large Behavior Models LBMs) oder verwandte, die aus Expertendemonstrationen vielfältige Aufgaben einschließlich Handlungsmuster und kontext-sensitive Interaktionen lernen. Sie bieten in der Regel erweiterte RL-Fähigkeiten in entsprechenden Anwendungen, insbesondere mit Rückkopplung aus der realen Welt, aber nicht nur. Der Stand der Technik im Bereich der Künstlichen Generischen Intelligenz und der Künstlichen Super Intelligenz (auf Englisch: Artificial Super Intelligence ASI) wurde in [35] kritisch betrachtet, mit einer nüchternen Sichtweise, welche intelligente Maschinen mit menschenähnlichem Verhalten oder sogar übermenschlichen Fähigkeiten entwickelt worden sind oder derzeit auf dem Weg sind. Abbildung 7 zeigt noch ein Beispiel im Bereich der multivariaten Approximationstheorie, einem Bestandteil der Funtionalanalysis in der Mathematik, und dann im Bereich Robotik eingesetzt, wie Lernmethoden und Systeme der Künstlichen Generischen Intelligenz entwickelt werden können. Abbildung 7(I) zeigt die idealisierte Umgebung: ein Roboterdreifingergreifer mit 4 Entfernungsmessern und die zu greifende generische Kugel. Abbildung 7(II) zeigt verschiedene Situationen wie die Entfernungsmessungen von der Kugeloberfläche vorkommen können, Vorder- und Seitenansichten werden angezeigt. Abbildung 7(III) zeigt links die multivariate Funktion $(x, y, z, r) = \vec{f}(z_1, z_2, z_3, z_4)$, die mittels ein selbstentwerfendes neuronales Netzwerk approximiert wurde und rechts die Trainings- und Testdatengenerierung, die mittels einer geeigneten Simulationsumgebung realisiert wurde. Der Ansatz kann auf andere zu greifende Körper angewandt werden. Das Generalisieren des Generalisierens auf irgendwelche zu greifenden Objekte erfordert z.T. das Umgebungsurkunden und adaptive Lernmethoden wie hier z.T. skizziert. Während Schachspielpartien nur ein paar Dutzend Züge dauern, die KI-Systeme bereits beherrschen, erfordern komplexe Lösungen für viele mathematische Probleme je nach Komplexität Millionen, Milliarden oder sogar mehr Schritte. Erste KI/ML-Systeme, die dabei behilflich sein können, befinden sich bereits in der FuE, die insbesondere lange, schwer zu findende Schrittfolgen entdecken, d.h. die Ausreißer in der statistischen Verteilung der Lösungen, analog zum automatischen Entdecken von Ausreißern [36] in zur Verfügung gestellten Daten beim völlig automatischen Entwerfen von neuronalen Netzwerkarchitekturen und deren Training durch assoziierte Lernmethoden [17]. Im Kern basieren viele der KI/ML-Lösungsmethoden auf Suchstrategien und liefern je bessere Lösungen desto bessere Suchstrategien sie beinhalten, wie z.B. in [37] eindeutig beschrieben. In diesem Bericht werden verschiedenartige Mathematikprobleme untersucht, wo der Einsatz von KI-ML-Methoden vehemente Vorteile gegenüber dem Stand der Technik einführen, z.T. um ungelöste Probleme der Mathematik herauszufordern und ggf. zu lösen. Zumaldest einiger dieser Lösungen werden uns erlauben, weiterhin KI-ML-Systeme mit menschenähnlichem Verhalten oder sogar übermenschlichen Fähigkeiten [38] zu entwickeln, wie der Verfasser seit Jahrzehnten meistert.

Literatur

- [1] V.D. Sánchez. *IEEE Fellow Award – "for leadership in neural and parallel computation, and pioneering contributions to autonomous space robots"*. 1995.
<https://profdrvdsaphd.lima-city.de/documents/IEEEFellow.pdf>
- [2] V.D. Sánchez. *Neurocomputing 50th volume anniversary*. Neurocomputing, 50, ix, 2003
- [3] V.D. Sánchez. *Personal written communication with the Office of the German Federal Minister of Research and Technology BMFT – Start of the First German National AI/ML Research and Technology Development Program*. 1988.
- [4] M. Dehn. *Papers on Group Theory and Topology*. Translated from German by J. Stillwell, Springer 1987.
- [5] W. Dyck. *Gruppentheoretische Studien*. Mathematische Annalen 20, März 1882, 1–44.
- [6] J. Meier. *Groups, Graphs and Trees: An Introduction to the Geometry of Infinite Groups*. Cambridge University Press 2008.
- [7] D.L. Johnson. *Presentations of Groups*. Cambridge University Press 1990.
- [8] R.C. Lyndon and P.E. Schupp. *Combinatorial Group Theory*. Springer 2001.
- [9] W. Magnus et al. *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations*. Revised edition, Dover 2004.
- [10] A. Arutyunov. *Combinatorial Group Theory – Lecture Notes*. December 2024. https://www.researchgate.net/publication/387518435_Combinatorial_group_theory_Lecture_Notes
- [11] W.v.O. Quine. *The Problem of Simplifying Truth Functions*. The American Mathematical Monthly, 59 (1952) 8, 521–531.
- [12] E.J. McCluskey Jr. *Minimization of Boolean functions*. Bell System Technical Journal 35 (1956) 6, 1417–1444.
- [13] R.K. Brayton et al. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers 1984.
- [14] A.K. Chandra and G. Markowsky. *On the number of prime implicants*. Discrete Mathematics 24 (1978) 1, 7–11.
- [15] V.D. Sánchez. *Analysis and Implementation of Heuristical Algorithms for Two-Level Logic Minimization of Digital Circuits, 3 volumes (in German)*. Final Report, Karlsruhe Institute of Technology (KIT), February 1986.
- [16] V.D. Sánchez and G. Hirzinger. *The State of the Art of Robot Learning Control Using Artificial Neural Networks*. In O. Khatib, J.J. Craig, and T. Lozano Perez (Eds.), *The Robotics Review 2*, Cambridge, Massachusetts, The MIT Press, 261–283, 1992.
- [17] V.D. Sánchez. *Beiträge zum überwachten Lernen reellwertiger Funktionen mithilfe neuronaler Netze*. Deutsches Zentrum für Luft- und Raumfahrt (DLR) Bericht, unveröffentlicht, 1993.
- [18] M.C. Mackey and L. Glass. *Oscillation and Chaos in Physiological Control Systems*. Science 197 (1977) 4300, 287–289.
- [19] A. Vaswani et al. *Attention Is All You Need*. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

- [20] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd Edition, 2018.
- [21] A.A. Markov. *Extension of the Law of Large Numbers to Dependent Events (in Russian)*. Bulletin of the Society of the Physics Mathematics, Kazan, Russia, 2, 155–156.
- [22] R. Bellman. *A Markovian Decision Process*. Journal of Mathematics and Mechanics 6 (1957) 5, 679–684.
- [23] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [24] S. Kullback and R.A. Leibler. *On Information and Sufficiency*. The Annals of Mathematical Statistics 22 (1951), 79–86.
- [25] J. Schulman et al. *Proximal Policy Optimization Algorithms*. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, December 4–9, 2017.
- [26] J.J. Andrews and M.L. Curtis. *Free groups and handlebodies*. Proc. Amer. Math. Soc. 16 (1965), 192–195.
- [27] S. Akbulut and R. Kirby. *A potential smooth counterexample in dimension 4 to the Poincare conjecture, the Schoenflies conjecture, and the Andrews–Curtis conjecture*. Topology 24 (1985) 4, 375–390.
- [28] A. Shehper et al. *What makes Math problems hard for reinforcement learning: A case study*. Preprint, 11 February 2025. <https://arxiv.org/pdf/2408.15332v2>
- [29] C.F. Miller III and P.E. Schupp. *Some presentations of the trivial group*. In Contemporary Mathematics, volume 250 – Groups, Languages and Geometry, Editor: R.H. Gilman, American Mathematical Society 1999, 113–116.
- [30] V.D. Sánchez. *Advanced Gen AI & LLM Foundations and Applications – Paving the way to a more powerful and diverse ML –*. Dezember 2023.
<https://profdrvdsaphd.lima-city.de/documents/AdvancedGenAILLMs.pdf>
- [31] OpenAI. *Pioneering research on the path to AGI*. März 2025. <https://openai.com/research/>
- [32] DeepSeek. *DeepSeek-R1*. März 2025. <https://github.com/deepseek-ai/DeepSeek-R1>
- [33] U. Kamath et al. *Large Language Models: A Deep Dive – Bridging Theory and Practice*. Springer 2024.
- [34] V.D. Sánchez. *AI/ML Methods to Develop Superior Next Gen Autonomous Learning Robot Systems for Industrial Terrestrial and Space Applications*. Dezember 2024.
<https://profdrvdsaphd.lima-city.de/documents/AIMLNextGenAutonLearnRobots.pdf>
- [35] V.D. Sánchez. *On Artificial General Intelligence (AGI) and Artificial Super Intelligence (ASI)*. Januar 2025. <https://profdrvdsaphd.lima-city.de/documents/OnArtificialGeneralIntelligenceAndArtificialSuperIntelligence.pdf>
- [36] V.D. Sánchez. *Robustization of a Learning Method for RBF Networks*. Neurocomputing, 9, 85–94, 1995.
- [37] V.D. Sánchez. *Searching for a solution to the automatic RBF network design problem*. Neurocomputing, 42 (1-4), 147–170, 2002.
- [38] V.D. Sánchez. *Enabling Robots to become more human and obtain superhuman capabilities for terrestrial and space applications using advanced AI/ML*. November 2024. <https://profdrvdsaphd.lima-city.de/documents/EnablingSuperhumanRobotsUsingAdvancedAIML.pdf>