Michael Herrmann

Generieren wir eine Logik der Entdeckung durch Machine Learning?

Abstract

In der Literatur werden weitrechende Behauptungen aufgestellt, die Machine Learning eine Leistungsfähigkeit attestieren, die bisher der traditionellen wissenschaftlichen Methode zugeschrieben worden ist. So soll z.B. eine automatisierte Entdeckung von physikalischen Gesetzen möglich sein. Diese starke Behauptung wird in diesem Aufsatz einer Kritik unterzogen. Lernende Algorithmen können zwar als induktive Methode charakterisiert werden, die dann unsichere Schlüsse auf neue Hypothesen erlauben - und somit für den Menschen neues Wissen bereithalten können. Sie regulieren also die Entstehungs- und Entdeckungsbedingungen von wissenschaftlichen Hypothesen in Teilen der modernen Wissenschaft, die mit Machine Learning arbeiten.

Der Aufsatz wird dann aber erstens aufzeigen, welche Rolle maschinell lernende Algorithmen für den Entdeckungszusammenhang einer Hypothese spielen können. Zweitens wird argumentiert, dass Machine Learning nicht an die Stelle von wissenschaftlicher Theorie- und Hypothesenbildung treten kann. Die Generierung von Hypothesen kann nicht vollständig einem computerisierten Automatisierungsprozess übergeben werden, da dieser in Teilen von strukturellen Vorannahmen abhängt, die durch menschliche Eingriffe zustande kommen.

Recent publications link the efficiency of machine learning methods to more standard scientific methods . Under this interpretation, automated discovery of physical laws by machine learning methods could become feasible. In this paper, I critically examine this claim. To this end, I first characterize machine learning as an inductive method that facilitates the creation of a standard scientific hypothesis. I then show which role machine learning methods typically play in the context of discovery. I argue that machine learning cannot substitute traditional ways of theory construction based on the idea that the generation of a hypothesis depends on structural assumptions made by humans and therefore cannot fully be handed over to a computerized automated process

1 Einführung

Rechtfertigungsstrategien in der Wissenschaft zu untersuchen und sie dann zu formalisieren, war und ist eines der Interessensgebiete der Wissenschaftsphilosophie. Mit deduktiver und induktiver Logik können wir die Schlussweisen studieren, mit denen Ansprüche auf neues Wissen begründet werden. Logisch gültige Schlussformen, wie zum Beispiel der Modus ponens, sind in einem regelbasierten Kalkül des natürlichen Schließens formalisierbar, das es erlaubt, wahre Aussagen aus wahren Aussagen abzuleiten. Induktive Schlussweisen sind hingegen stets einem Hume'schen Skeptizismus ausgesetzt und können nicht mit deduktiver Sicherheit logisch rekonstruiert werden. Parallel zu den Rechtfertigungsstrategien ist es dennoch

lohnenswert, zu fragen, ob wir auch bei der Entstehung wissenschaftlicher Hypothesen ein (induktives) Regelsystem vorfinden, das steuernd die Hypothesenbildung selbst vorantreibt? Solche Fragen waren durch den Einfluss des logischen Empirismus im 20. Jahrhundert in den Hintergrund gerückt. Der Entdeckungsakt selbst sei der logischen Analyse unzugänglich, denn hierfür sei Folgendes nicht vorhanden:

> »[…] a mechanical procedure analogous to the familiar routine for the multiplication of integers, which leads, in a finite number of predetermined and mechanically performable steps, to the corresponding product.«[1]

Auch Hans Reichenbach hält wenig davon, philosophisch die Wege zu studieren, die zur Bildung einer Hypothese führen, denn

> »[e]s gibt keine logischen Regeln, auf deren Grundlage eine Entdeckungsmaschine gebaut werden könne, die die schöpferische Funktion des Genies übernehmen würde.«[2]

Eine wissenschaftsphilosophische Analyse sollte erst bei der fertig vorliegenden Hypothese zum Tragen kommen. Dieser Gedankengang geht zurück auf die wirkmächtige Unterscheidung Reichenbachs zwischen Entdeckungs- und Begründungszusammenhang.[3] Der Entdeckungszusammenhang umfasst dabei die Umstände und Bedingungen, die zur Formulierung einer Hypothese führen, ohne aber Aussagen über die inhaltliche Korrektheit der Hypothese eine Aussage zu machen. Die Geltungsgründe kommen erst im Rechtfertigungszusammenhang zum Tragen, der Aufschluss darüber gibt, auf welche Weise für eine Behauptung argumentiert, mit welchen Strategien eine Hypothese untermauert wird.

Die junge Disziplin des maschinellen Lernens scheint nun über interessante automatisierbare Strategien und Regelsysteme zur Generierung von Hypothesen zu verfügen. Hiermit soll eine automatisierte Entdeckung von physikalischen Gesetzen oder die Bestimmung neuer erklärender Variablen (die sich selbst nicht aus der Kombination schon bekannter Variablen ergeben) möglich sein. Man kann diese automatisierbaren Strategien deutlicher fassen, wenn man Machine Learning als eine Methode des unsicheren Schließens begreift. Ich möchte an dieser Stelle nicht ausarbeiten, inwiefern Inferenz im maschinellen Lernen als eine Form des induktiven oder abduktiven Schließens zu beschreiben ist. Für diese Arbeit genügt es, Methoden des maschinellen Lernens als unsichere Schlüsse aufzufassen, um zu verstehen, welchen Beitrag ein algorithmisch orientierter Zugang zur Logik der Entdeckung leisten kann; ich folge hier Jantzen, der schreibt:

---

1 Carl G. Hempel: *Philosophy of natural science*. [Nachdr.]. Upper Saddle River, NJ 1966 (Prentice-Hall foundations of philosophy series), S. 14.

2 Hans Reichenbach: *Der Aufstieg der Wissenschaftlichen Philosophie*. Wiesbaden 1968 (Wissenschaftstheorie Wissenschaft und Philosophie, 1), S. 260.

3 Hans Reichenbach: *Experience and prediction: an analysis of the foundations and the structure of knowledge*, Chicago 1938.

> »A logic of discovery is a computable method or procedure for generating one or more hypotheses from a set of empirical facts, preexisting theories, and theoretical constraints, where each hypothesis generated is significant in that it is both consistent with existing data and likely to be projectible over a substantial range of data not previously in evidence. In other words, a hypothesis is significant just if it is likely to make true predictions about previously unknown cases in its intended domain. […] As my definition suggests, I understand limited projectibility to be a necessary condition for a hypothesis to be significant or to constitute a discovery. Note that I am not insisting that an assessment or test of limited projectibility be part of a logic of discovery.«[4]

Die Bestimmung einer Logik der Entdeckung fokussiert hier also auf die Generierung von Hypothesen. Ebenso folge ich Jantzen, wenn er schreibt, dass

> »The given definition also leaves open the form and content of a logic of discovery. Such a logic need not resemble simple, 'enumerative induction', e.g., every raven we've seen has been black, therefore hypothesize that all ravens are black. Nor need the method be elegantly axiomatizable in some first-order language.«[5]

Ein lernender Algorithmus darf dabei theoretisch beliebig komplexe mathematische Konstruktionen und Ausdrücke verwenden, solange dieser Algorithmus berechenbar (im Sinne der algorithmischen Komplexität) ist. Das bedeutet, dass gegeben eine endliche Datenmenge, die Berechnung des Algorithmus nach endlicher Zeit aufhört und als Resultat eine Hypothese formuliert, die begrenzte Voraussagekraft besitzt.

Machine Learning ist also für Wissenschaftsphilosophen interessant, weil hier vielleicht eine Disziplin im Aufbau begriffen ist, die so etwas wie eine Logik der Entdeckung begründen könnte. In der Literatur werden sogar weitreichendere Behauptungen aufgestellt, die der Methode Machine Learning eine Leistungsfähigkeit attestieren, die bisher der traditionellen wissenschaftlichen Methode zugeschrieben worden ist. Hintergrund solcher Zuschreibungen (auf die ich gleich eingehen werde) ist die immer noch offene philosophische Debatte, ob und wie Disziplinen wie Computersimulation und Machine Learning Wissenschaft und die empirische wissenschaftliche Methode verändert. Paul Humphreys lieferte zu dieser Frage wichtige Beiträge, in denen er zu zeigen versucht, wie sich Mathematik und Technik in den oben genannten Disziplinen auf eine besondere Weise miteinander verschränken und Computersimulationen das Verhältnis der Wissenschaftler zu ihrer Methode verändern.[6]

---

4 Benjamin C. Jantzen: »Discovery without a ›logic‹ would be a miracle«, in: *Synthese* 193 (2016), Heft 10, S. 3209–3238, S. 3211.

5 Ebd., S. 3212.

6 Vergleiche hierzu Paul Humphreys: *Extending ourselves*. *Computational science, empiricism, and scientific method*, Oxford 2004, sowie Paul Humphreys: »The philosophical novelty of computer simulation methods«, in: *Synthese* 169 (2009), Heft 3, S. 615–626.

Aus philosophischer Sicht ist die neue Disziplin Machine Learning deswegen interessant, weil sowohl zum Teil Autoren aus den Ingenieurswissenschaften, der Data Science wie der Philosophie behaupten, dass der Entdeckungszusammenhang von wissenschaftlichen Theorien bzw. die Herstellung von Hypothesen mithilfe von Machine-Learning-Algorithmen automatisiert werden könne. So schreibt zum Beispiel Freno:

> »Machine Learning algorithms deliver *models* of the data they are trained on […]. My claim is that such models are nothing but computational counterparts of what we commonly regard as scientific *theories*.«[7]

Den Begriff Theorie entledigt er seiner philosophischen Bedeutungsschwere, wenn er weiter schreibt:

> »Theories are tools for performing (different kinds of) inference. Although a loose notion of theory as an inferential device may fall short of the expectations of philosophers of science, that notion has the important effect of encouraging us to turn our interest from the reflection on the notion of theory to the identification of rational strategies for developing (extending, revising, etc.) scientific theories. Viewing theories broadly as inferential devices allows us to realize how machine learning methods are nothing but methods for automating the construction of scientific theories.« [8]

Wenn dies richtig ist, dann müsste folgerichtig auch behauptet werden, dass der relative Anteil intellektueller menschlicher Arbeit bei der Generierung wissenschaftlicher Thesen herabgesetzt ist. Aufgrund von Äußerungen wie z.B. von Chris Anderson, dass mit der Möglichkeit von maschinell lernenden Algorithmen »the end of theory« erreicht und »the scientific method obsolet« sei,[9] scheint es mir ratsam, zu klären, erstens was Machine Learning ist, zweitens wie in dieser Disziplin der Anteil an intellektueller menschlicher Aktivität zu begreifen ist. Und drittens möchte ich klären, ob mit dieser Disziplin eine Logik der wissenschaftlichen Entdeckung gegeben ist, wie sie oben definiert worden ist.

Ich werde im Folgenden einerseits für die Existenz einer solchen Logik argumentieren. Andererseits werde ich aber auch die Reichweite und Grenzen einer solchen Logik aufzeigen: Machine-Learning-Algorithmen sind keine ›Wundermaschinen‹. Sie beschreiben einen nicht-trivialen diskreten Prozess, neue Struktur- oder Abhängigkeitsbeziehungen in Daten zu erfassen. Die Qualifikation als begriffliche Neuheit

7 Antonio Freno: »Statistical Machine Learning and the Logic of Scientific Discovery«, in: *Iris. European Journal of Philosophy and Public Debate* (2009), Heft 2, S. 375–388, S. 377.
8 Ebd., S. 378.
9 Chris Anderson: »The End of Theory: The Data Deluge Makes the Scientific Method Obsolete«, in *Wired,* 23.6.2008, https://www.wired.com/2008/06/pb-theory/ (aufgerufen: 26.7.2018).

ist dabei aber in zweifacher Weise rückbezogen auf eine externe, hier menschliche Bewertungsinstanz. Erstens weiß der Mensch in der Regel vorher noch nicht von diesen Strukturen oder Mustern in den Daten. Und zweitens generiert ein maschinell lernender Algorithmus eine Hypothese über neue Muster auf der Grundlage von Variablen, die im Vorhinein bekannt sind bzw. durch deren Wahl Daten beschrieben werden. Daraus folgt, dass die begriffliche Neuheit des Inhalts einer maschinell-algorithmisch generierten Hypothese prinzipiell aus den beschreibenden Variablen abgeleitet ist. Die Eigenschaft der (begrifflichen) Neuheit ist damit explanatorisch reduzibel.

Um überzogene Erwartungen an das Feld des Machine Learning zu dämpfen, wäre es sicherlich lohnenswert, die Beziehung von Statistik und Machine Learning zu bestimmen. Wollen beide Disziplinen nicht dasselbe, nämlich Methoden entwickeln, um aus (großen) Datenmengen etwas zu lernen und zu versuchen, Unsicherheit diesbezüglich zu quantifizieren und zu kontrollieren? Bilden diese beiden Disziplinen die Pole eines Kontinuums, das sich durch die Handhabe großer Datenmengen und sehr hoher Rechenleistung moderner Computer aufgespannt hat? Sind also Machine-Learning-Probleme im Wesentlichen statistische Probleme, die man im Prinzip per Hand lösen könnte, dies aber praktisch nicht tut, weil schlicht zu viel Rechenarbeit vonnöten wäre?[10]

Flach beginnt sein Buch über Machine Learning damit, allgemein die Methodik von Machine Learning mit dem folgenden Slogan zu beschreiben: »Machine Learning is all about using the right features to build the right models that achieve the right tasks.«[11]

10 Eine Antwort auf diese Fragen kann an dieser Stelle aus Platzgründen leider nicht geleistet werden und bleibt Aufgabe zukünftiger Arbeiten.

11 Peter Flach: *Machine learning. The art and science of algorithms that make sense of data*, Cambridge 2012, S. 13.

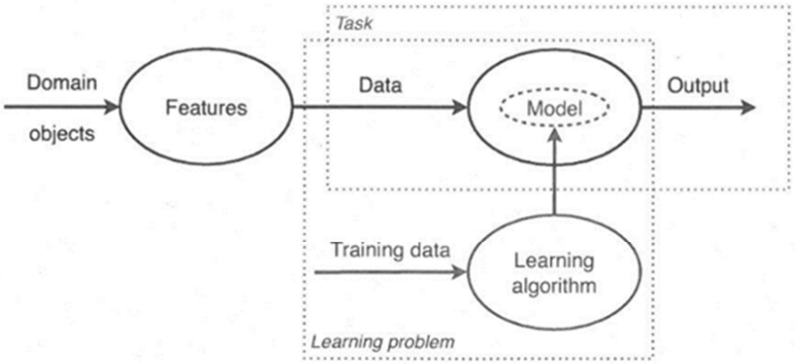

Figure 3. An overview of how machine learning is used to address a given task. A task (red box) requires an appropriate mapping – a model – from data described by features to outputs. Obtaining such a mapping from training data is what constitutes a learning problem (blue box).

Abbildung 1: Ein schematischer Überblick von Flach über die Vorgehensweise von maschinell lernenden Algorithmen.[12]

Features ist der Machine-Learning-Begriff für die Variablen, mit denen wir die Daten beschreiben. Die *Task* beschreibt die Aufgabe, die ein maschinell lernender Algorithmus ausführt. Die Aufgabe kann zum Beispiel darin bestehen, aufgrund von Trainingsdaten eine Regressionsanalyse durchzuführen. Man versucht hierbei eine Struktur in den Daten durch eine Regressionsfunktion näher zu charakterisieren. Mit einer Lernstrategie kann dann ein Algorithmus die noch unbekannten Parameter der Regressionsfunktion bestimmen und erzeugt somit das Modell (das hier durch die Regressionsfunktion mit ihrem gefundenen Parameter gegeben ist).

Ein *Learning Problem* besteht dann darin, mithilfe von Trainingsdaten ein Modell zu finden, das die gewählte *Task* ›korrekt‹ ausführt.

Das grundlegende Ziel von Machine Learning besteht also darin, Algorithmen zu entwickeln, die *Learning Problems* lösen. Aus dieser Definition wird bereits ersichtlich, dass ein sogenannter ›*data-driven approach*‹, wie Machine Learning, immer noch auf der Wahl einer Lernstrategie beruht. Und diese Wahl ist von Menschenhand und bedarf einer Rechtfertigung.

Welche Probleme Machine Learning angehen kann und wie genau hier das *Learning Problem* gelöst wird, möchte ich anhand von linearen Methoden zeigen,

12 Ebd., S. 11.

die für Vorhersagen (darunter z.B. Klassifizierung, Wahrscheinlichkeitsschätzer und Regressionsanalysen) eingesetzt werden.[13]

Diese Form von *Learning Problems* erinnert an eine induktive Schlussform, in der von einer empirischen Datenbasis auf eine generische Regel hypothetisch geschlossen wird. Mit Machine Learning scheinen wir eine Methode zur Hand zu haben, mit der zumindest (kontext-abhängige) Regeln angegeben werden können, wie wir aus Daten und einem Modell auf eine Hypothese schließen können. In diesem Sinne möchte ich Machine-Learning-Algorithmen als eine Logik der Entdeckung begreifen.

Setzt man voraus, dass zwischen Statistik und Machine Learning keine klar definierte Grenze verläuft, ist es legitim, die Aufgabe der Klassifikation als die einfachste Machine-Learning-Methode anzusehen. Klassifikation beschreibt den Prozess, einen Input (z.B. ein Bild) einer Klasse (z.B. Hund oder Katze) zuzuordnen. Und tatsächlich beruht zum Beispiel moderne Bilderkennung auf elaborierten Formen dieser einfachen linearen Regression, die ich nun vorstelle.

Die Aufgabe bei der linearen Regression besteht darin, Inputdaten $x = (x_1, x_2, \ldots, x_n)$ einem Label y über eine noch unbekannte Funktion $f(\cdot)$ zuzuordnen, die von weiteren Parametern $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ abhängt:

$$y = f(x, \omega)$$

In einer binären Klassifikation gibt es nur zwei Klassen, in die eingeteilt wird. Die Entscheidungsfunktion oder Hypothese $h(x)$ wäre dann die Signumsfunktion $\mathrm{sgn}(f(x))$, definiert als:

$$h(x) = sng(f(x)) = \begin{cases} 1, \text{ falls } f(x) \geq 0 \\ -1 \text{ sonst} \end{cases}$$

Diese Funktion f entscheidet dann über die Zuordnung. Die Aufgabe für den Menschen besteht nun darin, erstens auszuwählen, welche Features in den Daten x relevant sind, zweitens die geeignete Funktionenklasse für f zu wählen und drittens die richtigen Parameter ω zu finden.

Feature-Wahl

Auch wenn dieser Prozess in Teilen automatisiert werden kann, muss der Mensch diejenigen Variablen spezifizieren, die zur Problemlösung herangezogen werden sollen. In der Statistik nennt man dies die Wahl der Prädiktoren. Ebenso müssen Grö-

13 Die Eigenschaft der Linearität ist mathematisch gut verstanden, was diesen Ansätzen eine gewisse Einfachheit und Stabilität hinsichtlich Auswirkungen von Rauschen in den Trainingsdaten verleiht.

ßen in den Daten oft normalisiert werden, bevor man sie einem algorithmischen Verfahren übergeben kann.

Wahl der Funktionenklasse für f

Die Wahl der parametrisierten Funktion *f* setzt fest, nach welcher Vorschrift unsere Daten einem Label *y* zugeordnet werden. Wenn die Aufgabe z.B. in einer Vorhersage für die Nachfrage nach einem bestimmten Produkt besteht, dann müssen Funktionenklassen mit periodischen Eigenschaften gewählt werden, um zeitlich sich wiederholende Ereignisse abzubilden. Grundsätzlich kann *f* jede Form einer parametrisierten Funktion annehmen. Zum Beispiel kann man sich *f* als ein Polynom dritter Ordnung vorgeben:

$$f(x) = \omega_0 + \omega_1 x + \omega_2 x^2 + \omega_3 x^3$$

Das *Learning Problem* besteht hier nun also darin, die unbekannten Parameter $\omega_0, \ldots, \omega_3$ zu finden, um das Modell (hier die Funktion *f*) eindeutig zu bestimmen, das die gegebenen Daten ›gut‹ repräsentiert.

Parameterwahl

Die Parameterwahl ist eine Optimierungsaufgabe (im Englischen *loss function*), wie man sie bei ganz vielen Machine-Learning-Algorithmen antrifft. Hierbei sucht man eine Objektfunktion zu bestimmen, die die Güte der Funktion *f* hinsichtlich der Trainingsdaten angibt. In unserem Beispiel haben wir die Trainingsdaten $x_1, \ldots x_n$ und die entsprechenden labels $y_1, \ldots y_n$ gegeben. Mit welcher Zuordnungsvorschrift x_i nun auf y_i abgebildet wird, ist das Problem. Wir müssen also die unbekannten Parameter $\omega_0, \ldots, \omega_3$ schätzen.

Für den bekanntesten Schätzer benutzt man die Methode der kleinsten Fehlerquadrate (*least squares method*), die auf Carl Friedrich Gauß zurückgeht. Die Differenz zwischen dem Wert y_i und dem Funktionswert der Regressionsfunktion an der Stelle x_i, d.h. $|f(x_i) - y_i|$, ist der Fehler, den die Regressionsanalyse an einem Datenpunkt (x_i, y_i) macht.

Als Objektfunktion E wählt man hierbei die Summe der quadrierten Fehler $(f(x_i) - y_i)^2$: Der Fehler $|f(x_i) - y_i|$ soll nun für alle $i = 1 \ldots n$ klein werden. Folglich sucht man ein Minimum der Objektfunktion. Hierfür müssen wir die partiellen Ableitungen nach $\omega_0, \omega_1, \ldots, \omega_3$ bilden und diese Null setzen:

$$\frac{\partial}{\partial w_0} E(w_0, w_1, w_2, w_3) = 0$$
$$\frac{\partial}{\partial w_1} E(w_0, w_1, w_2, w_3) = 0$$
$$\frac{\partial}{\partial w_2} E(w_0, w_1, w_2, w_3) = 0$$
$$\frac{\partial}{\partial w_3} E(w_0, w_1, w_2, w_3) = 0$$

Hierdurch entsteht ein lineares Gleichungssystem, dessen Lösung die gesuchten Parameter $\omega_0, \omega_1, \ldots, \omega_3$ liefert.

Hennig und Kutlukaya betonen dabei, dass die Wahl der Objektfunktion keine mathematische Aufgabenstellung ist, sondern dass es sich vielmehr um ein Übersetzungsproblem aus einem nicht-formalen Kontext in eine mathematische Formel handelt:

> »[…] the task of choosing a loss function is about the translation of an informal aim or interest that a researcher may have in the given application into the formal language of mathematics. The choice of a loss function cannot be formalized as a solution of a mathematical decision problem in itself, because such a decision problem would require the specification of another loss function. Therefore, the choice of a loss function requires informal decisions, which necessarily have to be subjective, or at least contain subjective elements.«[14]

So legen die beiden Autoren in einer Fallstudie dar, wie z.B. bei einer Regressionsanalyse die (subjektive) Wahl zwischen dem quadrierten Fehler $(f(x_i) - y_i)^2$ und dem einfachen Fehler $|f(x_i) - y_i|$ starke Auswirkungen hatte auf die Güte verschiedener Regressionsmethoden. Auf das subjektive Moment in einer sonst formal ablaufenden Wissenschaft, wie der Statistik, gehen die beiden Autoren am Ende ihres Aufsatzes ein:

> »We use ›subjectivity‹ here in a quite broad sense, meaning any kind of decision which can't be made by the application of a formal rule of which the uniqueness can be justified by rational arguments. ›Subjective decisions‹ in this sense should take into account subject-matter knowledge, and can be agreed upon by groups of experts after thorough discussion, so that they could be called ›intersubjective‹ in many situations and are certainly well-founded and not ›arbitrary‹. However, even in such situations different groups of experts may legitimately arrive at different decisions. This is similar to the impact of subjective decisions on the choice of subjective Bayesian prior probabilities.«[15]

14 Christian Hennig und Mahmut Kutlukaya: »Some thoughts about the design of loss functions«, in: *Revstat Statistical Journal* 5 (2007), Heft 1, S. 19–39, S. 21.
15 Ebd., S. 36.

Wie mit einer weiterentwickelten Methode, die auf den Einsichten der linearen Regression beruht, gearbeitet wird, will ich nun anhand einer binären Klassifikation darlegen. Ein Algorithmus lernt dabei anhand von Trainingsdaten neue Daten (die nicht mit den Trainingsdaten übereinstimmen) in eine von zwei Klassen einzuteilen.

In einem Beispiel wollen wir eine Voraussage entwickeln, ob es regnen wird, indem wir Tage als regnerisch oder trocken klassifizieren, um dann daraus zu schließen. Als Trainingsdaten liegen die Durchschnittstemperaturen und die Feuchtigkeitswerte pro Tag vor. Wir wollen nun eine Trennlinie finden, die idealerweise alle Regentage von den Trockentagen trennt. Mit dieser Trennlinie kann man dann, gegeben Temperatur- und Feuchtigkeitswerte, Voraussagen treffen, ob es sich um einen Regentag handelt oder nicht.

In der obigen Abbildung 2 werden die vorhandenen Temperaturdaten auf der *x*-, die Feuchtigkeit auf der *y*-Achse abgebildet. Die Kreise sind dabei Regentage, die Dreiecke repräsentieren trockene Tage. Die Linie, die die Daten in zwei Bereiche einteilt, repräsentiert eine Hypothese über den Zusammenhang der unabhängigen Variablen Temperatur, Feuchtigkeit und der abhängigen Variable Regen. Woher kommt nun diese Hypothese, wie finden wir die Grenzlinie? Ein wichtiger lernender Algorithmus für diese Aufgabe ist der SVM-Algorithmus.

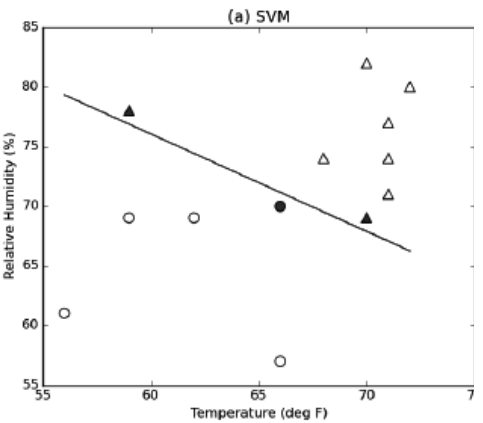

Abbildung 2: Ein *Support-Vector-Machine*-(SVM)-Algorithmus generiert eine Hypothese über den Zusammenhang zwischen Temperatur- und Luftfeuchtigkeitswerten, um sagen zu können, ob es in Zukunft regnen wird.[16]

16 Jantzen: »Discovery without a ‘logic’ would be a miracle«, in: *Synthese*, S. 8.

Vereinfacht ausgedrückt, löst dieser SVM-Algorithmus wieder ein (hier ein quadratisches) Optimierungsproblem. Und zwar wird die beste Wahl der Trennlinie diejenige sein, die den Abstand der Trainingsdaten zu der Trennlinie maximiert. Natürlich ist dieser ›Abstand‹ a priori keine Größe, die der Computer oder der Algorithmus versteht. Denn wie soll der Abstand zwischen einem Datenpunkt (x_i, y_i) und einer Trennlinie definiert werden? Es gibt unendlich viele Möglichkeiten. Wir müssen eine Wahl treffen und einen Abstandsbegiff definieren. In diesem Algorithmus wurde nun die Wahl getroffen, die kürzeste orthogonale Strecke zwischen einem Datenpunkt und einer Trennlinie als Abstand zu definieren. Als Abstand eines Trainingsdaten Sets wird dann der maximale Abstand aller Trennlinien bezeichnet.

In der folgenden Abbildung 3 sieht man zwei mögliche Trennlinien (rote und grün). Mit dem oben gewählten Abstandsbegriff sieht man, dass der Abstand z_2 größer als z_1 ist. Das bedeutet, dass in diesem Beispiel der Abstand von den Trainingsdaten zur grünen Linie maximal wird. Die Daten können so maximal voneinander separiert werden. Eingabedaten, die sich durch eine Trennlinie trennen lassen, werden linear separabel genannt.

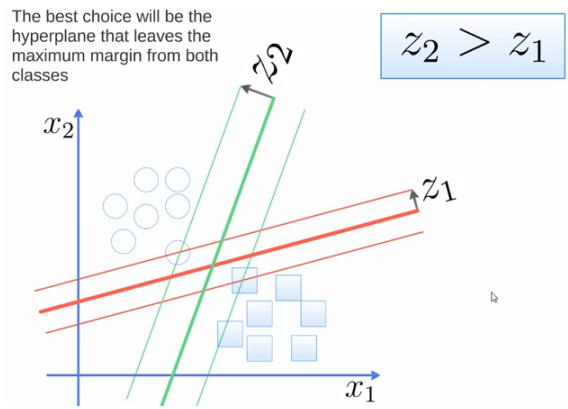

Abbildung 3: Im SVM-Algorithmus wird diejenige Trennlinie gewählt, durch die der Abstand der Trainingsdaten zu dieser Trennlinie maximal wird.[17]

Der SVM-Algorithmus lernt also anhand der Trainingsdaten, die im Sinne des beschriebenen Optimierungsproblems „beste“ Trennlinie für die binäre Klassifikationsaufgabe zu finden. In der folgenden Abbildung 4 ist dies die grüne Gerade $g(x)$.

17 Thales Sehn Körting: *How SVM (Support Vector Machine) algorithm works*, in: *YouTube*, 6.1.2014, https://www.youtube.com/watch?v=1NxnPkZM9bc&t=178s (aufgerufen am 26.7.2018).

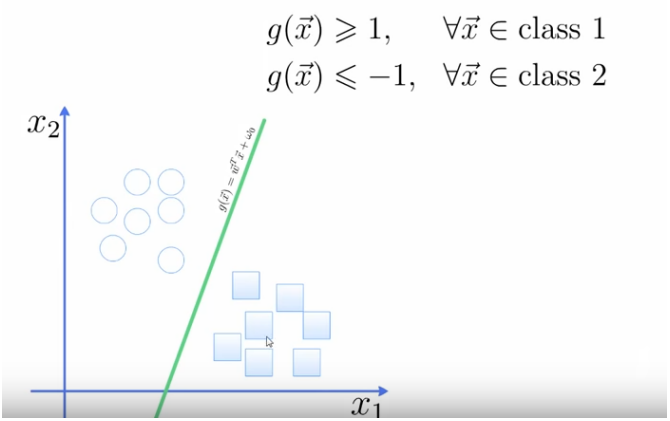

Abbildung 4: Der SVM-Algorithmus findet die beste Wahl der Hypothese, hier repräsentiert durch die grüne Linie.[18]

Die Entscheidungsfunktion oder Hypothese $h(x)$, die bestimmt, wie Daten $x = (x_1, \ldots, x_n)$ gemäß der Trennlinie binär klassifiziert werden, lautet dann wie weiter oben beschrieben:

$$h(x) = sng(g(x)) = \begin{cases} 1, & \text{für alle } x \text{ in Klasse 1} \\ -1, & \text{für alle } x \text{ in Klasse 2} \end{cases}$$

Diese Methode, über einen lernenden SVM-Algorithmus eine Hypothese zu generieren, findet zum Beispiel Anwendung in der Bioinformatik.[19] 2007 nutzen Han u.a. einen SVM-Algorithmus, um Proteine anhand ihrer Aminosäurenzusammensetzung, Volumen, Polarität und Hydrophobie hinsichtlich einem bestimmten Wirkstoffpotential zu beschreiben.[20]

Auch wenn der SVM-Algorithmus ein erfolgreich genutzter Hypothesen-Generierer ist, der wenig anfällig für *Overfitting* ist, sollte man bedenken, dass dieser Ansatz nur dann ein nützliches Werkzeug für die Hypothesenbildung ist, wenn sich die Daten linear separieren lassen. Überschneiden sich zum Beispiel die Datensätze, kann

18 Ebd.

19 Vgl. Zheng Rong Yang: »Biological applications of support vector machines«, in: *Briefings in Bioinformatics* 5 (2004), Heft 4, S. 328–338, http://dx.doi.org/10.1093/bib/5.4.328.

20 Lian Yi Han u.a.: »Support vector machines approach for predicting druggable proteins: recent progress in its exploration and investigation of its usefulness«, in: *Drug Discovery Today* 12 (2007), Heft 7, S. 304–313.

der SVM-Algorithmus keine Hypothese bestimmen, er endet in einer Endlosschleife.[21]

Wichtig hierbei ist aber, dass das Finden einer möglichst guten Trennung nicht separierbarer Daten mit der kleinsten Anzahl von Fehlern und Iterationsschritten *NP*-vollständig ist. Um dieses Problem zu umgehen, wird ein sogenannter ›Kerneltrick‹ verwendet, der es erlaubt, das Problem wieder lösbar zu machen und auch nicht-separierbare Daten zu klassifizieren.

4 Der Einspruch gegen die automatisierte Erkennbarkeit von Strukturen

Angesichts vorzeigbarer Erfolge von Machine-Learning-Algorithmen auf verschiedenen Gebieten ist es unbestritten, dass sie hilfreiche Werkzeuge sind bei der Formulierung von Hypothesen über empirische Regelmäßigkeiten und gesetzesähnliche Beziehungen. So haben wir im obigen Beispiel der linearen Klassifikation gesehen, dass wir über eine algorithmisierte Methode verfügen, eine Hypothese zu generieren, die eine gut gesicherte Vorhersagefähigkeit besitzt.

Darüber hinaus wird in der Literatur lernenden Algorithmen auf zwei Feldern eine erstaunliche Leistungsfähigkeit attestiert, die bisher ausschließlich der traditionellen wissenschaftlichen Methode zugeschrieben worden ist. So werden einerseits *Machine-Learning*-Algorithmen präsentiert, die eine automatisierte Entdeckung von physikalischen Gesetzen versprechen. Andererseits wird behauptet, dass *Machine-Learning*-Algorithmen neue erklärende Variablen bestimmen können, die sich selbst nicht aus der Kombination schon bekannter Variablen ergeben. Ich werde Beispiele für beide Fälle anführen, möchte jedoch erst darauf hinweisen, dass bereits früh in der Geschichte der modernen Wissenschaftstheorie Bedenken geäußert worden sind über Grenze und Reichweite algorithmisierter Methoden. So schrieb zum Beispiel bereits Hempel:

> »Scientific theories and hypotheses are usually couched in terms that do not occur at all in the description of the empirical findings on which they rest. […] A logic of discovery would have to provide a mechanical routine for constructing, on the basis of the given data, a hypothesis or theory stated in terms of some quite novel concepts, which are nowhere used in the description of the data themselves.«[22]

21 Eine Möglichkeit, dieses Problem zu umgehen, wäre, falsch klassifizierte Daten zuzulassen und nach einer Trennlinie zu suchen, die einerseits die Anzahl der falschen Klassifikationen und andererseits den Abstand dieser ›falschen‹ Daten zur Trennlinie minimiert. Auf diesen Ansatz wird hier aber aus Platzgründen nicht eingegangen.

22 Hempel: *Philosophy of natural science*, S. 14.

Ein ähnlicher Einwand findet sich etwas später bei Woodward:

> »A computer programmer armed with an arsenal of concepts and definitions can write a program that, employing random search procedures of varying sophistication, explores the various possible interrelationships of the variables contained in those concepts. ›New‹ concepts can be generated by clustering the variables already present in different ways. But such clustering will not produce conceptual shifts like that from the concept of impetus of the middle ages to the principle of inertia— the key idea in the Scientific Revolution of the 16th and 17th centuries.«[23]

Beide Autoren argumentieren dafür, dass aus *Machine-Learning*-Methoden keine *conceptual novelty* hervorgehen könne. Hempels und Woodwards Kritik könnte man so zusammenfassen: Ausgehend von bekannten Variablen V, die ein Phänomen beschreiben, können lernende Algorithmen keine neuen erklärenden Variablen V' generieren, sodass diese dann Voraussagen oder Erklärungen liefern für einen Phänomenbereich, der nun von der Vereinigung $V \cup V'$ beschrieben wird.

Dagegen argumentieren zum Beispiel die Autoren Schmidt und Lipson. In ihrem Artikel »Distilling Free-Form Natural Laws from Experimental Data« legen sie dar, wie auf der Basis von experimentellen Daten u.a. aus einem harmonischen Federpendel mit symbolischer Regression[24] die Bewegungsgleichungen abgeleitet werden können:

> »Without any additional information, system models, or theoretical knowledge, the search with the partial-derivative-pairs criterion produced several analytic law expressions directly from these data. [...] We have demonstrated the discovery of physical laws, from scratch, directly from experimentally captured data with the use of computational search. We used the presented approach to detect nonlinear energy conservation laws, Newtonian force laws, geometric invariants […] without prior knowledge about physics, kinematics, or geometry.«[25]

Es ist sicherlich richtig, dass durch diese automatisierte Entdeckung von Gesetzesbeziehungen eine menschenunabhängige Mechanisierung des Entdeckungszusammenhangs vollzogen wird und wir tatsächlich von einer Logik der Entdeckung sprechen können: Es gibt eine algorithmisierbare regelbasierte Methode zur Generierung einer Hypothese (hier einer Gesetzesbeziehung).

Mithilfe von Hempel und Woodwards möchte ich aber zeigen, dass der Anspruch der Autoren, »without any additional information analytic law expressions directly

23 James F. Woodward: »Logic of discovery or psychology of invention?«, in: *Found Phys* 22 (1992), Heft 2, S. 187–203, S. 200.

24 Bei der symbolischen Regression werden nicht nur die unbekannten Parameter einer vorher definierten Funktion bestimmt, wie wir es weiter oben in diesem Text gesehen haben. Vielmehr wird hier selbst die Form der mathematischen Funktion selbst noch variiert und gemäß einer Fehlerabschätzung dann die geeignetste ausgewählt.

25 Michael Schmidt und Hod Lipson: »Distilling free-form natural laws from experimental data«, in: *Science (New York, N.Y.)* 324 (2009) Heft 5923, S. 81–85, S. 82 u. 85.

from these data« zu generieren, falsch ist. Woodward sagt es deutlich, dass zwar »›new‹ concepts can be generated by clustering the variables already present in different ways. But such clustering will not produce conceptual shifts«. Und genau auf diese Weise gehen die Autoren vor, schreiben sie doch selbst:

> »Given position and velocity data over time, the algorithm converged on the energy laws of each system (Hamiltonian and Langrangian equations). Given acceleration data also, it produced the differential equation of motion corresponding to Newton's second law for the harmonic oscillator and pendulum systems.«[26]

Was Woodward also in Zweifel zieht, ist die Möglichkeit, über Machine-Learning-Algorithmen neue erklärende Variablen aus den Daten zu gewinnen. Und dies findet in der zitierten Arbeit nicht statt. Die betrachteten Variablen (das ist der Winkel der Auslenkung sowie die Winkelgeschwindigkeit des Pendels) sind im Vorhinein bekannt. Bei Schmidt und Lipson sind sogar die Gesetze, die approximiert werden sollen, bekannt. Die Identifikation und Spezifizierung von potentiell erklärenden Variablen selbst ist eine nicht-triviale Aufgabe, die einfach mechanisiert oder automatisiert werden kann. Ein relevanter Suchraum muss zu allererst durch theoretische wie praktische Einschränkungen eingegrenzt werden, um Kandidaten für Prädiktor-Variablen festzulegen. Erst dann können algorithmisierbare Heuristiken formuliert werden.

Anders fomuliert: *Machine-Learning*-Methoden hängen von strukturellen Vorannahmen ab. Welche potentiellen Prädiktor-Variablen zieht man heran, um Daten zu beschreiben? Wie kommt es zur Wahl der Modelle, mit denen Machine-Learning-Algorithmen *Learning Problems* lösen sollen? Handelt es sich z.B. um eine Optimierungsaufgabe? Was genau wird in dem Verfahren minimiert oder maximiert? Sind diese Verfahren eindeutig, in dem Sinne, dass immer ein Extremum gefunden wird? Werden nur lokale Extrema gefunden oder gibt es Sicherheit, dass wir das globale Extremum finden?

Als Konsequenzen aus diesen Überlegungen folgen zwei Dinge. Mit *Machine-Learning*-Algorithmen ist erstens eine praktisch durchführbare Logik der Entdeckung formuliert. Sie generiert regelgebunden Hypothesen mit beschränkter Voraussagekraft. Maschinelle Algorithmen regulieren die Entstehungs- und Entdeckungsbedingungen von wissenschaftlichen Hypothesen in Teilen der modernen Wissenschaft, die mit *Machine Learning* arbeiten.

Zweitens habe ich dafür argumentiert, dass datengetriebene Methoden jedoch nicht an die Stelle von wissenschaftlicher Theorie- und Hypothesenbildung treten können. Lernende Algorithmen detektieren nicht voraussetzungslos Strukturen, die in Daten vorhanden sind. Wir können Daten nur mit denjenigen Variablen analysieren, die wir vorher definiert haben. Typische Probleme von *Machine Learning* be-

26 Ebd., S. 83.

treffen Klassifikations- oder Regressionsprobleme. Hier will man die Abhängigkeitsbeziehung einer Outputvariable von (meist mehreren hochdimensionalen) Prädiktorvariablen auf der Grundlage von Daten herausfinden. Die Spezifizierung dieser Abhängigkeitsbeziehung kann einem computerisierten Automatisierungsprozess übergeben werden. Aber welche Prädiktorvariablen und strukturellen Vorannahmen bezüglich Modellwahl diesem Automatisierungsprozess dann übergeben werden, ist immer noch eine Aufgabe menschlich-wissenschaftlicher Praxis.

Man kann dann die Woodward'sche Kritik auch als ein Kreativitätsargument auffassen. Um dies zu tun, weise ich darauf hin, dass bereits Alan Turing in »Computing Machinery and Intelligence« das Argument behandelt, dass Computermethoden nicht verantwortlich für konzeptuelle Neuheit sein können.

> »A variant of Lady Lovelace's objection states that a machine can ›never do anything new‹. […] A better variant of the objection says that a machine can never ›take us by surprise‹.«[27]

Turing interpretiert diesen Einwand so, dass der Kritiker mit dem Überraschungsmoment ein Vermögen von Kreativität voraussetzt, das (fundamental) neue Konzepte hervorbringen kann.

> »He will probably say that surprises are due to some creative mental act on my part, and reflect no credit on the machine.«[28]

Und Turing fährt nun fort, weiter zu entfalten, was ›sein Kritiker‹ hier mit Kreativität meint:

> »The view that machines cannot give rise to surprises is due, I believe, to a fallacy to which philosophers and mathematicians are particularly subject. This is the assumption that as soon as a fact is presented to a mind all consequences of that fact spring into the mind simultaneously with it. It is a very useful assumption under many circumstances, but one too easily forgets that it is false. A natural consequence of doing so is that one then assumes that there is no virtue in the mere working out of consequences from data and general principles.«[29]

Es spricht aus meiner Sicht vieles dafür, Woodwards Kritik analog zu dem Kreativitätsargument zu lesen, gegen das Turing argumentiert. Das Argument von Woodward als Machine-Learning-Kritiker lautete, dass Machine-Learning-Methoden – im Gegensatz zu Menschen – keine neuen Konzepte hervorbringen können bzw. nicht

27 Alan Turing: »Computing Machinery and Intelligence«, in: *Mind* 59 (1950), S. 433–460, S. 450.
28 Ebd., S. 451.
29 Ebd., S. 451.

imstande sind, konzeptuelle Veränderungen wie z.B. das Trägheitsprinzip von Newton zu generieren.

5 Künstliche neuronale Netze als neue Form von maschinellem Lernen?

Zur Klasse der maschinell lernenden Algorithmen gehören auch die sogenannten künstlichen neuronalen Netze (*artificial neural networks*, kurz: ANN). Ich möchte hier noch darauf eingehen, wie diese Methode mit den in diesem Aufsatz erwähnten maschinellen Algorithmen für Klassifizierungsaufgaben zusammenhängt. Ich argumentiere im Folgenden dafür, dass sich ANN mit linearer bzw. polynomialer Regression und *Support Vector Machines* (SVM) hinsichtlich des methodischen Vorgehens gut vergleichen lassen. Zudem verfolgen beiden Methoden das gleiche Ziel, unbekannte Datenmengen zu klassifizieren. Unter dieser Annahme lässt sich daher meine im Aufsatz entwickelte These auch auf künstliche Netze ausdehnen.

Die Bezeichnung ›neuronale Netze‹ ist irreführend. Diese geht zurück auf die ursprünglich biologische Motivation bei der Entwicklung der ersten künstlichen Netze in den 1940er Jahren, mit diesen Netzen die Funktionsweise des menschlichen Gehirns besser zu verstehen. Man beschreibt diese Methode tatsächlich mathematisch präziser mit dem Begriff Funktionennetz, das benutzt wird, um ein funktionales Verhältnis von gegebenen Input- und Output-Daten zu approximieren.

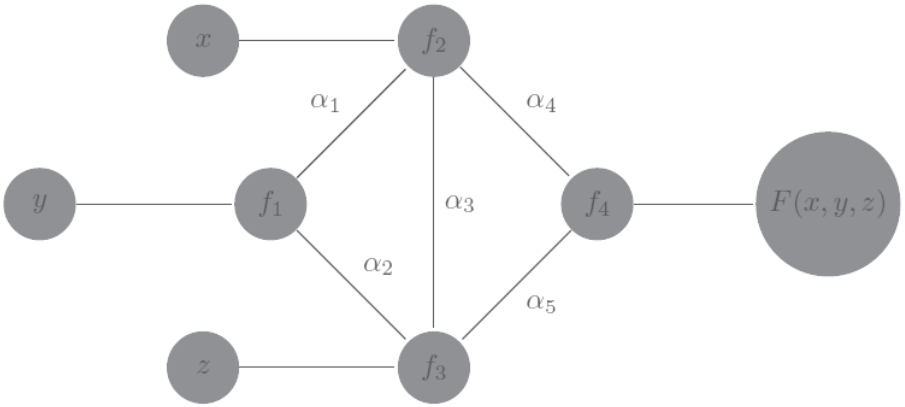

Abbildung 5: Neuronale Netze sollten als ein Funktionennetz verstanden werden, das sich hier aus den Aktivierungsfunktionen f_1, f_2, f_3 und f_4 zusammensetzt.[30]

Künstliche neuronale Netze haben eine wie in Abbildung 6 abgebildete Struktur. Das Funktionennetz kann hier als die Funktion $F(x, y, z)$ interpretiert werden. Die Funktion F selbst ist die Komposition der Funktionen f_1, f_2, f_3 und f_4. Jede Funktion f_i $(i = 1,2,3,4)$ steht für die Aktivitätsfunktion eines Neurons, das auch Knoten genannt wird. Ein ANN liefert nun eine Approximation an diese Funktion F. Knoten sind durch Kanten verbunden. Die Stärke der Verbindung zweier Knoten wird durch ein sogenanntes Gewicht (in der Abbildung $\alpha_1, \alpha_2, \alpha_3, \alpha_4$) angegeben. Der Input, den ein Knoten von einem anderen empfängt, hängt von zwei Werten ab, die in der Regel multiplikativ miteinander verknüpft sind: dem Output des übertragenden Knoten und dem Gewicht der entsprechenden Kante zwischen beiden Knoten. Die Schicht(en) an Neuronen, die sich zwischen Input- und Output-Neuronen befinden, nennt man verborgene Schicht(en). In der sogenannten Trainingsphase treffen Inputwerte auf die erste Schicht von Neuronen, werden dann gemäß den Gewichten und der Aktivierungsfunktion verändert und laufen so unter sukzessiver Veränderung an jedem Knoten aller Schichten von Neuronen entlang bis ein Outputwert generiert wird.

30 Raúl Rojas und R.-H. Schulz (ed.): *Was können neuronale Netze?,* Mannheim 1994 (Mathematische Aspekte der Angewandten Informatik), S. 55–88.

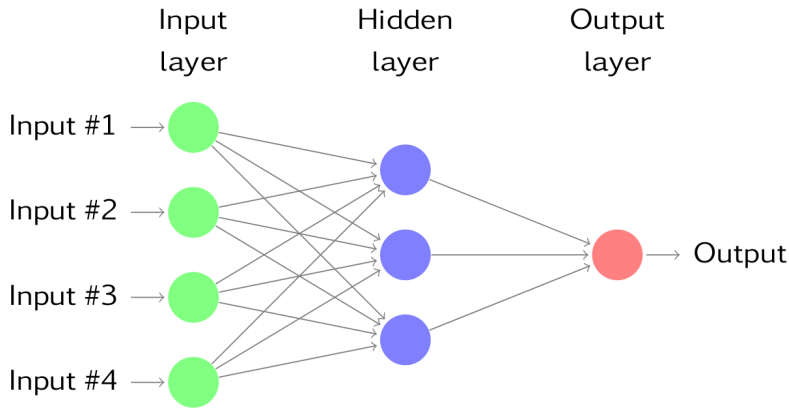

Abbildung 6: Ein schematischer Überblick eines einfachen neuronalen Netzes.[31]

Eine Menge von gegebenen Input/Output-Daten $(x_1, y_1), \ldots, (x_n, y_n)$ sei Trainingsmenge genannt. Das Lernproblem für ANN besteht nun darin, jene Funktion F zu finden, die die Inputwerte ›am genauesten‹ den Outputwerten zuordnet. Wir suchen also die Wahl der ›optimalen‹ Gewichte, die die beste Approximation an die Input/Output-Daten erlaubt. Um die Gewichte in der Trainingsphase zu ändern, benötigt man eine Lernregel, die angibt, wie eine Änderung vorgenommen werden soll. Diese Regel besteht in einem Algorithmus, der ermittelt, welche Gewichte des neuronalen Netzes wie stark erhöht oder reduziert werden müssen. Hierfür hat sich der sogenannte „Backpropagation-Algorithmus“[32] (etabliert, der ein (lokales) Minimum der folgenden Fehlerfunktion E sucht. Der Fehler E berechnet die quadratische Differenz des korrekten Wertes der gegebenen Outputwerte $y_1, \ldots, y_n$ und den durch das neuronale Netz berechneten Output $F(x_1), \ldots, F(x_n)$:

$$E(\omega_1, \omega_2, \ldots, \omega_l) = \sum_{i=1}^{n} (y_i - F(x_i))^2$$

Der *Backpropagation*-Algorithmus versucht nun die Gewichte so zu verändern, dass der resultierende Gesamtfehler E möglichst klein ausfällt. Das Netz wird mit zufälligen Anfangsgewichten initialisiert. Mithilfe des Gradientenabstiegsverfahrens werden nun rückwärts in jeder Schicht von Neuronen die Gewichte ein kleines Stückchen in die Richtung angepasst, die den Fehler kleiner macht (Abstieg in die negative Gradientenrichtung von E). Nachdem alle Gewichte angepasst wurden, erfolgt ein erneuter Durchlauf des neuronalen Netzes mit den gegebenen Inputwerten und der Fehler E wird erneut gemessen. Für die neu generierte Liste an Gewichten wird

31 Diese Graphik wurde der Webseite https://becominghuman.ai/artificial-neural-networks-and-deep-learning-a3c9136f2137 entnommen.
32 Vgl. Rojas 1996, Kapitel 7.

abermals der Gradient bestimmt und damit eine Modifikation der Gewichte erreicht. Dieses Verfahren wiederholt man so lange, bis ein lokales Minimum gefunden wird oder eine vorher festgelegte Anzahl an Iterationsschritten erreicht worden ist.

In der anschließenden Testphase wird auf der Grundlage der bereits ermittelten Gewichte aus der Trainingsphase untersucht, ob das neuronale Netz von den Trainingsdaten abstrahieren kann und auch korrekte Klassifizierungsergebnisse für bisher nicht gelernte Input-Daten liefern kann.

Das sogenannte *universal approximation theorem* besagt, dass ein neuronales Netz mit nur einer verborgenen Schicht mit endlichen vielen Neuronen, unter gewissen Bedingungen an die Aktivierungsfunktion, lokal jede stetige Funktion approximieren kann.[33] Da neuronale Netze also auch sehr komplizierte Funktionen hinreichend gut annähern können, sollte das Problem des *Overfitting* beachtet werden. Denn am Ende will man nicht die perfekte Funktion für die gegebenen Input-/Output-Daten finden. Das Netz soll vielmehr in der Testphase in der Lage sein, Inputdaten, die nicht Teil der Trainingsdaten waren, richtig zu klassifizieren.

Ein Problem des *Backpropagation*-Algorithmus ist, dass ihm nur die lokale Umgebung des Gradienten bekannt ist. Daher weiß der Algorithmus zum Beispiel nicht, ob er ein lokales oder absolutes Minimum gefunden hat. Dies ist insbesondere bei Netzen mit sehr vielen Verbindungen zwischen den Neuronen der Fall. Ebenso kann aufgrund flacher Plateaus des Fehlergraphen der Gradient beim Gradientenabstiegsverfahren so klein werden, dass das nächste ›Tal‹‹ gar nicht mehr erreicht wird. Stagnation des Verfahrens ist die Folge. Oder es können Oszillationen des Fehlergraphen entstehen, sodass der *Backpropagation*-Algorithmus weder ein globales noch ein lokales Minimum entdeckt. Durch (menschlichen) Eingriff in das Gradientenabstiegsverfahren können diese Probleme aber angegangen werden (u.U. treten dann Folgeprobleme auf).

Bei einer linearen oder allgemein polynomialen Regressionsanalyse liegt ein ähnliches Lernproblem vor wie bei künstlichen neuronalen Netzen. Denn hier ging es darum, ein Polynom *n*-ten Grades bzw. deren Koeffizienten zu finden. Auch dieses Lernproblem führt auf eine Optimierungsaufgabe, möglichst diejenigen Koeffizienten zu finden, sodass der Gesamtfehler im Abgleich mit den Testdaten ausreichend gering ist. Xi Cheng u.a. argumentieren sogar in einem Preprint auf ArXiv.org, dass »neural networks actually *are* polynomial regression models.«[34] Argumentativer Kern ihres Aufsatzes ist, dass die Aktivierungsfunktionen in neuronalen Netzen allesamt durch Polynome approximiert werden können. Infolge wird dann auch die

33 George Cybenko: »Approximation by superpositions of a sigmoidal function«, in: *Math. Control Signal Systems* 2 (1989), Heft 4, S. 303–314. DOI: 10.1007/BF02551274.

34 Xi Cheng u.a.: »Polynomial Regression As an Alternative to Neural Nets«, in: *ArXiv.org* 29.6.2018, https://arxiv.org/abs/1806.06850, (aufgerufen: 28.6.2018), S. 3, Hervorhebung im Original.

Funktion *F* ein Polynom sein, das durch polynomiale Regressionsmodelle approximiert werden kann.

Trotz methodischer Ähnlichkeiten bin ich vorsichtig, beide Ansätze als äquivalent zu betrachten. Ein unterscheidendes Merkmal scheint die Vorhersageleistung bei der Klassifikation unbekannter Datensätze zu sein. Hier scheinen künstliche neuronale Netze deutlich besser zu sein. Eine Erklärung hierfür könnte in der nicht-polynomialen Struktur künstlicher Netze im Gegensatz zu linearer bzw. polynomialer Regression liegen.

Ich habe in diesem Aufsatz aufzuzeigen versucht, welche Rolle maschinell lernende Algorithmen für den Entdeckungszusammenhang einer Hypothese spielen. Reichweite und Grenze dieser (wissenschaftlichen) Methode wurden in diesem Aufsatz so bestimmt, dass maschinelle Algorithmen zwar einen positiven Beitrag bei der Aufstellung einer Hypothese leisten, überzogene Ansprüche an eine Automatisierung wissenschaftlicher Entdeckung und Hoffnungen auf eine automatisierte Theorie- und Hypothesenbildung allerdings zurückgewiesen wurden.

Machine-Learning-Methoden liefern Berechnungsmodelle, um Klassifikations- und Regressionsprobleme (approximativ) zu lösen. Dabei können diese algorithmischen Ansätze sehr effizient darin sein, Abhängigkeitsbeziehungen einer Outputvariablen zu (mehreren) Inputvariablen aufzudecken. Und damit verfügen wir über einen computerisierten Automatisierungsprozess, Hypothesen über Zusammenhänge in Daten zu formulieren. In diesem Sinne habe ich Machine Learning als Regulator der Entstehungs- und Entdeckungsbedingungen von (wissenschaftlichen) Hypothesen aufgefasst. Ein begründeter Optimismus in die Reichweite dieser Technik darf aber nicht vergessen lassen, dass lernende Algorithmen nicht voraussetzungslos Strukturen in Daten detektieren können. Zuallererst sind dabei strukturelle Vorannahmen zu treffen, die z.B. eine Auswahl der Prädiktorvariablen festlegen, mit denen wir Daten überhaupt erst beschreiben können. Welche theoretischen und praktischen Einschränkungen ergeben sich, damit ein Computer eine Optimierungsaufgabe im Sinne des gestellten Problems ausführt? Mithilfe der zuletzt genannten Woodward-Turing Analogie hinsichtlich einer ›Kreativität‹ von Computermethoden, sind maschinell lernende Algorithmen immer in den Begriffen, mit denen sie initialisiert wurden, ›gefangen‹. Eine vorschnelle Verabschiedung menschlich-wissenschaftlicher Praxis in der computergestützten Naturwissenschaft wären daher verfrüht.