Anthropomorphised Algorithms.

Are humanisations of machine learning algorithms suitable for shedding light on the social impacts of artificial intelligence?

Joachim Heller *

5. 12. 2024

Abstract

Anwendungen maschinellen Lernens zu betrachten als Algorithmen, die eigenständige, aber absolut undurchschaubare gesellschaftliche Akteure sind, ohne die damit verbundenen theoretischen Annahmen aus einer technischen Perspektive zu überprüfen, verstellt den Blick auf den eigenen Anteil an der Ohmachtsempfindung gegenüber «computationaler Autorität». Einseitige Erklärungsansätze entfalten so auch keine politisch-emanzipierende Wirkung in der Medienbildung. Diese These wird vorliegend anhand der Rezeption eines Aufsatzes entwickelt, in dem die technische Perspektive als ungeeignet für die Ermittlung der Verstärkungswirkung von Diskriminierungstendenzen verworfen wird. Zur Kontrastierung dieser Auffassung wird die Kybernetik als methodischer Rahmen vorgeschlagen, der eine transparentere Terminologie bietet, um auch medienpädagogisch im Grundschulunterricht anschließen zu können. Dazu findet eine kritsche Auseinandersetzung mit dem im in Bezug genommenen Beitrag vorgefundenen Algorithmusbegriff statt, ein aufgeführtes Anwendungsbeispiel für die maschinen-habituelle und posthumanistische Analyse vermeintlich opaker Algorithmen wird technisch analysiert und aufgeklärt, neurale Netzwerke und klassische Programmiertechniken werden erörtert und der historische Rahmen wird umrissen, anhand der den Grundlagen Technikfolgenabschätzung nachgespürt wird.

One's own contribution to the feeling of powerlessness in the face of «computational authority» is obscured, if applications of machine learning are considered as algorithms that are independent, inscrutable social actors, without questioning the associated theoretical assumptions from a technical perspective. Biased explanatory

*. *Humboldt-Universität zu Berlin*, M.Ed. student in lateral entry to primary school teaching, contact: joachim.heller@hu-berlin.de

approaches therefore do not have a politically emancipatory effect in media education. This thesis is developed on the basis of the reception of an essay in which the technical perspective is rejected as unsuitable for determining the reinforcing effect of discrimination tendencies. To contrast this view, cybernetics is proposed as a methodological framework that offers a more transparent terminology in order to be able to integrate media education in primary school teaching. To this end, a critical discussion of the concept of algorithm found in the article referred to is carried out, an application example for the machine-habitual and posthumanistic analysis of supposedly opaque algorithms is technically analysed and explained, neural networks and classic programming techniques are discussed and the historical framework is outlined, based on which the foundations of technology assessment are traced.

Contents

List of Figures

1 Introduction

There are two perspectives for defining the term 'algorithm': one from a mathematical-technical perspective and one from everyday language. The latter encourages analogies that are given social scientific meaning and are intended to be compatible with media education. In particular, the image of a quasi-living entity is legitimised, whose cultural influence is supposedly made transparent through these analogies. In the following, an exposition written within this paradigm will be used to demonstrate how this strategy itself obscures the understanding of algorithms and, when applied in media education, can amplify inequalities. In section 2 the framework set up by the exposition's author is first considered, then in 3 his fallacies are discussed and in 4 their effects are traced.

2 Habitual-posthumanistic-neomaterialistic realm

Waldmann assumes that algorithms in software applications of machine learning are irreducibly opaque (Waldmann 2024, 1, 3). Notably, the posthumanist-neo-materialistic position based on *Barad* seems to have resonated with the author himself: he uses its terminology throughout (see Barad 2003). That position suggests a definition of 'algorithm' in which a recognisable self-consciousness (cf. Hegel 2014, 145–147) emerges, which, in conjunction with a habitually focused position, is a socialisable *relatively independent actor* with *unpredictable power to act* (Waldmann 2024, 2). From a non-posthumanist perspective, such an idea can be viewed as humanisation. The peculiar thing is that the anthropomorphisation of hardware components mentioned by *von Foerster* in 1970, (von Foerster 2002, 171–173), are obviously completely ignored in the argument and the humanisation takes place solely on the software level. This may initially be a 'Cartesian cut', with which the mind (the algorithm) *is considered* separated from the matter (the hardware) (Primas 1994, 610). In addition, the 'agential cut' is then supposed to have the effect that performatively, i. e. during and through presentation of the posthumanist observation, a previously non-existent 'relatum' is *created*, here the artificial subject 'algorithm', which is related to the analysis and independently *materialises itself* in the world thus reconfigured; in this case, the analysis is the 'apparatus' with which the cut is made 'intra-actively' and the components of the relationship only gain meaning for the analysts – perhaps also the hardware as a "mechanical apparatus" (constructed according to Waldmann 2024, 8, 11–13; Everth and Gurney 2022, 16; Barad 2003, 815).

Based on this, Waldmann would like to use a *posthumanist skepticism* as an alternative or complement to a sociologically influenced media pedagogical *triad* (Waldmann 2024, 14-19), that enables a position with which the opacity of algorithms (established as a given) can be adequately addressed in order to accept their output not doubtfully but skeptically as a possible truth with contingency expectations and not as certainty (19-21).

Cova sees a need for more substantial definitions of the digital in order to be able to more precisely target social scientific instruments for determining its social impact; he suggests paying more attention to the aspect of the generation of discrete (digital) and continuous (analog) signals and their transformation from one form to another and to reconsider structuralism for this (Cova 2016). Structuralism involves the recognition of connecting patterns, but has been criticized for not answering questions about the possibilities for changing power structures (Graf 2010, 212–213).

Waldmann assumes a *structural definition* of 'algorithm' (Waldmann 2024, 3), which he suspects of unreflectively reproducing the habitus and power structures of *white *cis men with academic backgrounds* (5). As far as the focus on power structures is to be sharpened, structuralism seems difficult to convey.

Nevertheless, it is necessary to find another theoretical approach, since Waldmann's analysis is technically incorrect and pedagogically problematic. His approach not only does not offer any possibility of changing power structures, but also has the destructive potential of inadvertently supporting what is to be avoided.

Another approach is Luhmannian systems theory. In view of the frequent criticism that it neglects humans (cf. for example Luhmann 2013, p. 35, note 47), it should at least offer the possibility for convergences from a skeptical *post*-humanist position itself. It is also just as stimulating and eclectic an epistemological project as Barad's quantum physical analogies.

The sociological arsenal of concepts shaped by systems theory leads back to cybernetics. The latter's coherent terminology for analysing and describing the behaviour of existing and hypothetical machines of any complexity (admittedly rejected by Barad in favour of her concept of "apparatus", Barad 2003, 816) in a wide variety of contexts, including social systems (for an overview, see Ashby 1957, 2–6), has been absorbed into sociology and numerous other disciplines (in mathematics it remained as *control theory*, cf. Russell and Norvig 2010, 15). «Artificial intelligence», which is also an application field of cybernetics (see for example Anderson 2024, 16; Sieniutycz 2020, 1–2), can be described more accurately and comprehensibly with the widely well-tried cybernetic conceptual system, from which terms such as 'determinate machine', 'black box', 'variety' in relation to control systems, etc. come. This also enables a connection to computer science. Since basic arithmetic is sufficient for their elementary understanding (Ashby 1957, v), their knowledge also offers potential for media education, especially for educational offerings at primary schools.

In the following, its terminology will first be incorporated into the explanation of Waldmann's erroneous reasoning.

3 Misconceptions

Waldmann concludes from an asserted fact:

> "Entstehung und Effekte algorithmischer Ungleicheit [können] **nicht** allein durch bewusst herbeigeführte Entscheidungen, die technische Artefakte betreffen, beobachtet und erklärt werden. [emphasis added]" (Waldmann 2024, 4)
>
> *The emergence and effects of algorithmic inequality can**not** be observed and explained solely by consciously made decisions concerning technical artifacts.*

to an ought intended by him:

> "[D]ie figurativen, diskursiven und relational-performativen Wirkungsweisen von Algorithmen [**sind**] genauer im Hinblick auf hierarchisierende und ausschliessende

> Logiken zu analysieren **und nicht** von handlungszentrierten Debatten über Transparenz und Kontrolle. [emphasis added]" (Waldmann 2024, 4)
>
> *The figurative, discursive and relational-performative effects of algorithms* ***are to be*** *analysed more closely with regard to hierarchical and exclusionary logics* ***and not*** *of action-centered debates about transparency and control.*

The chosen subjunction (material conditional) is in itself simply arbitrary: the entire statement is only false if the first claim is true, but the second partial statement is untenable. The equivalent would be to assert adjunctively: "*At least* with one of them, the relevant here can be observed and explained, *or* with the other it should be analysed." Now, a normative sentence cannot be true and so Waldmann is not even saying anything gratuitous at this point, but nothing at all. His further conclusions based on this therefore hang up in the air.

3.1 Algorithms

There is no "structural definition" of 'algorithm'. The only source referred to by Waldmann, from which he believes he has taken this information as a consensus in computer science and which at the same time, in line with his normative specification, leads to Barad, is *Burke*, who again overemphasizes the intellectual movement of *structured programming* (Burke 2019, 2), whose concept the latter mistakenly considers "more-or-less" as a precursor to other programming concepts such as object-oriented and functional programming (3), and he characterises the dogma of the separation of instructions ("code") and data as a reality-forming agential cut (3–4), as well as the summary of algorithms in program libraries (8). The magical thinking of "a disempowered user outside an opaque decision-making system" (9) from which an alternative anthropomorphised, "imprecise" algorithm concept arises (10), forms the basis for non-technical discourses on the cultural impact of algorithms (10–12): the *systemic definition* in Waldmanns sense (Waldmann 2024, 3). In certain cases (for example didactic ones), this simplification can be informally useful even in technical discourse (Burke 2019, 10).

However, the opacity of algorithms and their effects usually becomes clearer to the extent that the willingness to deal with them more precisely *also* as technical artifacts increases. In terms of programming, there are basically three forms of representation of algorithms. They can be formulated using everyday language, for example as shown in Figure 1.

Algorithm S (*Selection sampling technique*). To select n records at random from a set of N, where $0 < n \leq N$.

S1. [Initialize.] Set $t \leftarrow 0$, $m \leftarrow 0$. (During this algorithm, m represents the number of records selected so far, and t is the total number of input records that we have dealt with.)

S2. [Generate U.] Generate a random number U, uniformly distributed between zero and one.

S3. [Test.] If $(N - t)U \geq n - m$, go to step S5.

S4. [Select.] Select the next record for the sample, and increase m and t by 1. If $m < n$, go to step S2; otherwise the sample is complete and the algorithm terminates.

S5. [Skip.] Skip the next record (do not include it in the sample), increase t by 1, and go back to step S2. ▮

Figure 1: Generally formulated algorithm (Knuth 1998, 142)

This representation can then be translated more or less literally into source code as a second form of representation (figure 2) which is converted into machine language and becomes executable code.

```
(defun algorithm-s (n max) ; max is N in Knuth's algorithm
  (let (seen               ; t in Knuth's algorithm
        selected           ; m in Knuth's algorithm
        u                  ; U in Knuth's algorithm
        (records ()))      ; the list where we save the records selected
    (tagbody
     s1
       (setf seen 0)
       (setf selected 0)
     s2
       (setf u (random 1.0))
     s3
       (when (>= (* (- max seen) u) (- n selected)) (go s5))
     s4
       (push seen records)
       (incf selected)
       (incf seen)
       (if (< selected n)
           (go s2)
           (return-from algorithm-s (nreverse records)))
     s5
       (incf seen)
       (go s2))))
```

Figure 2: Literally translated, executable algorithm (Seibel 2011b, 251)

The third form of representation can be considered as transposition into a version corresponding to the structure of this language – as shown in figure 3.

```
(defun s-n-creator (n)
  (let ((sample (make-array n :initial-element nil))
        (i 0))
    (lambda (item)
      (if (<= (incf i) n)
          (setf (aref sample (1- i)) item)
        (when (< (random i) n)
          (setf (aref sample (random n)) item)))
      sample)))

(defun algorithm-s ()
  (let ((*random-state* (make-random-state t))
        (frequency (make-array '(10) :initial-element 0)))
    (loop repeat 100000
          for s-of-n = (s-n-creator 3)
          do (flet ((s-of-n (item)
                      (funcall s-of-n item)))
               (map nil
                    (lambda (i)
                      (incf (aref frequency i)))
                    (loop for i from 0 below 9
                          do (s-of-n i)
                          finally (return (s-of-n 9))))))
    frequency))
```

Figure 3: Idiomatically transposed executable algorithm (Collective 2024)

Depending on the means of abstraction, the algorithm can appear completely detached in the equivalent third representation (figure 4).

```
(defun algorithm-s (n max)
  (loop for seen from 0
     when (< (* (- max seen) (random 1.0)) n)
     collect seen and do (decf n)
     until (zerop n)))
```

Figure 4: Maximally simplified algorithm (Seibel 2011b, 252)

What is formulated as a compact loop in this programming language (Common Lisp) is translated by the automatic system in the background for the computer into jump instructions that correspond to the original formulation of the algorithm. The structure of this algorithm fits well into the internal workings of computers and ends with a result (as intended by Knuth 1998, v).

In a procedural *machine-oriented* programming language (see e.g. Knuth 2005, iv) the translation of the algorithm can look like in figure 5.

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

struct s_env {
    unsigned int n, i;
    size_t size;
    void *sample;
};

void s_of_n_init(struct s_env *s_env, size_t size, unsigned int n)
{
    s_env->i = 0;
    s_env->n = n;
    s_env->size = size;
    s_env->sample = malloc(n * size);
}

void sample_set_i(struct s_env *s_env, unsigned int i, void *item)
{
    memcpy(s_env->sample + i * s_env->size, item, s_env->size);
}

void *s_of_n(struct s_env *s_env, void *item)
{
    s_env->i++;
    if (s_env->i <= s_env->n)
        sample_set_i(s_env, s_env->i - 1, item);
    else if ((rand() % s_env->i) < s_env->n)
        sample_set_i(s_env, rand() % s_env->n, item);
    return s_env->sample;
}

int *test(unsigned int n, int *items_set, unsigned int num_items)
{
    int i;
    struct s_env s_env;
    s_of_n_init(&s_env, sizeof(items_set[0]), n);
    for (i = 0; i < num_items; i++) {
        s_of_n(&s_env, (void *) &items_set[i]);
    }
    return (int *)s_env.sample;
}

int main()
{
    unsigned int i, j;
    unsigned int n = 3;
    unsigned int num_items = 10;
    unsigned int *frequencies;
    int *items_set;
    srand(time(NULL));
    items_set = malloc(num_items * sizeof(int));
    frequencies = malloc(num_items * sizeof(int));
    for (i = 0; i < num_items; i++) {
        items_set[i] = i;
        frequencies[i] = 0;
    }
    for (i = 0; i < 100000; i++) {
        int *res = test(n, items_set, num_items);
        for (j = 0; j < n; j++) {
            frequencies[res[j]]++;
        }
    free(res);
    }
    for (i = 0; i < num_items; i++) {
        printf(" %d", frequencies[i]);
    }
    puts("");
    return 0;
}
```

Figure 5: Algorithm translated into low(er)-level programming language *C* (Collective 2024)

In the last defined procedure `main`, the sub-procedure `test` is called, which calls previously defined procedures, which in turn call previously defined procedures. The users of such languages have significantly fewer automatisms at their disposal, so that a data structure suitable for the solution is also user-defined at the beginning. On a small scale, the hierarchical and sequential structure of such very concrete languages becomes clear and it is measurable how complex more far-reaching problem solutions can become. The concept of structured programming is intended for these languages – precisely *without* jumps (Dijkstra 1970, 1968; Criticism of the generalisation: Harvey 1997a, 233–238), which of course are also implemented in the later machine code. This concept is the only programming 'paradigm' in the sense of binding "techniques . . . shared by the members of a given community" (Kuhn 1970, 175) as a mindset, implemented in programming languages that enforce the use of a specific programming style. With those that allow different strategies for organising source code components for different classes of problems, the term shifts to a more linguistic meaning, just as 'verb stem' is a paradigm of a syntagma into which a multitude of corresponding word components fit.

A language like the one demonstrated in figures 2 to 4 hides more of the technical peculiarities of computers and thus offers other ways of dealing with complexity, which add the equivalent *bottom-up design* (Graham 1993, v–vi) in computer science to the *top-down programming principle* (Waldmann 2024, 2). They and their relatives also manage without the dogmatic separation of data and code at the syntactical level. Data and code are structurally identical there, which enables further abstraction tools to reduce complexity (about the property initially Mooers and Deutsch 1965, 232).

Functional programming in languages such as the one demonstrated is a programming style based on mathematical function composition to exploit *recursive* processes and increase the reliabilty of complex program systems, especially in parallel processes (Turner 2013; Bauer 2004, 247–248), what is particularly accessible to cybernetic observation (von Foerster 2015, 73). Recursive processes, on the other hand, are avoided as much as possible in the less automated machine-oriented languages due to the more complex memory management (Bauer 2004, 240). However, recursive procedures are not unstructured or irregular (opposite opinion apparently Waldmann 2024, 3). They are generalised loop constructions that repeat the rules for working through tree-like, i.e. equally recursive data structures for each individual element until a stop condition is reached (Domkin 2021, 159; on the special case of tail-call recursion/iteration, Teschl and Teschl 2013, 221; Abelson, Sussman, and Sussman 1996, 45–47).

There is no such thing as *the* object-oriented programming. Nonetheless, the different models have in common that the user-defined data-structures are associated with code in a more dynamic way than in structured programming (introductory Seibel 2011a, 189–191). These models were developed *to overcome* the limitations of structured programming at even higher levels of complexity

All of these methods for organizing program source code support different problem-solving strategies in which algorithms are represented differently. Inspite of that, algorithms remain what they were before digital computers existed (Cormen et al. 2009, xv): A step-by-step procedure for solving a precisely described computational problem that, if followed correctly, leads to a correct result (5–6) – which is already true for ancient methods of Euklid or Heron (cf. e.g. Teschl and Teschl 2013, pp. 94–95, 177; Landa 1969, 22–24).

To use them reliably, you only need to know how they behave. Written division, for example, does not require any understanding of numbers (Padberg and Benz 2021, 226–227). Procedures are *always* used as *black boxes* (Ashby 1957, 86), which, as *determinate machines* (24), show *reliable* transformations from appropriate input to output values (11–16, 86–87). A good behaviour description also gives programmers an indication of the side effects, but otherwise the internal processes are usually irrelevant unless correctness or efficiency are put to the test.

The insights Waldmann believes he has gained from Burke about the nature of algorithms are not tenable in this respect. Nevertheless, he rejects the technical perspective as trivial; its knowledge would lead to *biased results* and to *cultural phantasms* (Waldmann 2024, 3, 5).

3.2 Raising awareness of decision options that affect technical artifacts

Complex software solutions are multiply coupled systems (Ashby 1957, 48–53), in which repeated transformations take place (16–23), Markov chains may be implemented for machine learning (165–173), other methods based on Bayesian probability statistics are used (Downey 2012, 3–6; Russell and Norvig 2010, 495–499) and numerous other approaches – but no *truth* (Waldmann 2024, 20). Two things become obvious: (1) The principles on which machine learning applications are based are explainable. (2) Complex chains with multiple transformations cannot be described in detail by the transition of the respective input values from one component to the next. However, it remains undeniable that the input of a self-learning system is transformed into an output, as well as that this transformation takes place on a rational basis, namely, as a rule, statistical procedures (also Burke 2019, 8–9).

3.2.1 Alleged categorical subjection

Using the representative example of a study on the use of software to exploit precariously employed delivery service employees in false self-employment by the actors in an ideal-typical anti-labour law start-up company (Waldmann 2024, 9), Waldmann attempts to demonstrate the posthumanist-neomaterialist analysis tool presented and to show the inevitable opacity of algorithms in applications of artificial intelligence.

For this purpose, on the one hand, a *machine habitus* is assumed, which represents the habitual characteristics of the people involved in the development and the automated learning processes, which is referred to as 'primary' and 'secondary' *machine socialisation*, from which path dependencies arise, which ultimately form a *practical sense of judgment* of the machines, albeit one that is difficult to predict (Waldmann 2024, 5–6). The sociological interpretation of the fictitious, representative case study should then show that the software originally used *constitutes a techno-social field* in terms of its habitus (10). The laws of that field are incomprehensible to all people involved, but the software realises symmetries at the level of the habitual participation of all actors, which then also includes the machine (11). On the other hand, the neo-materialist interpretation is intended to show, in contrast to this, but also in addition, that machines create reality through *algorithms as agential cuts* and are thus also independent participants in the discourse on inequalities that is interwoven with matter (13–14), which preserves the opacity of the algorithms.

Nevertheless, Waldmann himself gives the example of a correct output hypothesis in this context by drawing attention to the options for action hence decision in connection with the technical artifact, where actual courier drivers were able to specifically and successfully influence the output of their orderer's software in their favour (18). As heuristic *transactional practices* on a subversive level, such an approach is considered a playful, productive misuse of algorithms (according to referenced Allert and Asmussen 2017, 34–35). In this respect, algorithmic inequality continues to be understood as an overall irrevocable relationship of subordination in which only minimal scope remains for *micropolitically* effective actions (Waldmann 2024, 18).

3.2.2 Decision options shown by technical analysis

Another example of methodical behavioural analysis is provided by two computer scientists from the group of "BIPoC" (which is surprisingly underrepresented in Waldmann's sources although explicitly mentioned, 4), with a research focus on bias and artificial intelligence (see Buolamwini 2017, 2016), who have the competence to adequately observe, describe and analyse the transformations `Input`→`Output` and thus the *reproducible* behaviour of applications of complex statistical methods of machine learning. Due to the necessity of using automatic facial recognition for law enforcement purposes in the USA and other countries, *Buolamwini* and *Gebru* have comprehensibly and reproducibly identified the inaccuracies in the facial recognition of three software products *using an appropriately prepared data source* and have thus *clearly* demonstrated that, as an undesirable side effect, the statistical weightings coded at the time of their investigation do *not* lead to correct results, and the scientists can provide recommendations for action to improve the product (Buolamwini and Gebru 2018, 12). In fact, the focus in programming and analysis is generally more on the *data* used and generated and their structure than on the algorithms, which, however, does not mean a separation in the sense of structured programming (other opinion Burke 2019, 8–9), but rather allows the complexity to be shifted from the program code to the transparent data (cf. Domkin 2021, 29–30). Even the hidden ratings are data whose tendencies can be measured by the error rate. The analysis by Buolamwini/Gebru also allows for interpretations: It may say something about the proportion of *white *cis men with academic backgrounds* at IBM if the error rate of their product is 0.0% when it comes to facial recognition of light-skinned women (Buolamwini and Gebru 2018, 10). Above all, however, Waldmann's paradigm is opposed: technical analysis is *particularly* suitable for making algorithmically generated inequality transparent and providing effective recommendations for action to correct errors – and is therefore *relevant* (currently e.g. Sabbatini and Calegari 2024; older contribution to further ethical problems, from a more humanistic perspective: Van der Loos 2014). Skepticism about the results of complex machine learning processes does not require a posthumanist basis with anthropomorphic algorithms (Waldmann 2024, 19–20) to be effective. It arises from experience with evaluation processes in any complex normative system, that is carried out for decision-making in issues that affect *humans.*

3.2.3 Complexity, folk spirits and invisible hands

Nine years after *Turing* and *Church*'s articles on computability (Turing 1936; Church 1936), an early theoretical draft for modeling an artificial neural network as a logical calculus was developed by a neuroscientist and a young logician, based on the structure of a biological neural network (McCulloch and Pitts 1943, 117), with reference to Turing's work (Piccinini 2020, 107). These fundamental texts of theoretical computer science and mathematics are part of a wide range of interdisciplinary research projects that were summarised at the time under the keyword 'cybernetics', which, among other things, showed mathematical ways of dealing with increasing complexity. Their findings were initially used for military purposes during the Second World War. After the war, the application fields expanded to include the control of a wide variety of systems with the support of progressively versatile computers, the development of which was based on this cybernetic work. This period also saw Turing's contributions to possible future computer thinking capabilities (introducing "Intelligence", Turing 2004a; introducing the "Imitation Game", Turing 1950). In one he follows the thought experiment of mathematically conceivable computers, "which will simulate the behaviour of the human mind very closely. They will make mistakes at times, and at times they may make new and very interesting statements, . . . " (Turing 2004b, 472). The idea of creating *artificial intelligence* soon became attractive and the term was coined by *McCarthy* in 1955 (Anderson 2024, 4) "in opposition to the term *cybernetics*" and the most prominent representative of that academic movement, *Wiener* (Anderson 2024, 16). Unlike the latter, McCarthy who was 18 years old in 1945 did not have to conduct his research for the military during the war and had no concerns that his research would continue to be funded from the military budget. He believed in artificial intelligence to be "'the best hope of 'objective' solutions to societal problems'", therefore he was "one of the strongest proponents of *not* addressing [societal] risks" of somehow intelligent machines until the technology is "well advanced" and irreversibly implemented (9-10). Wiener, on the other hand, did have concerns: He was 18 years old in 1914 and thus experienced most of the dark first half of the 20th century as an adult, worked for the military during the war, categorically ruled out any further involvement in military projects after the war (Wiener 1947) and had a dismal vision of the future with learning and self-reproducing machines (Wiener 1950, 214; 2021, 242–243), a robotics pioneer by himself (Wiener 1950, 191–194). Another prominent critic from the 1970s onwards was *Weizenbaum*, the developer of the first "conversation agent" in 1966 ELIZA, the reference point of all chatbots today (Berry 2023, 2). He was unpleasantly surprised at how easily users of the essentially simple program allowed themselves to be deceived in this first supposedly easy Imitation Game, especially as an automated, in fact mindless psychotherapist (Weizenbaum 1976, 3–8; Harvey 1997b, 147–150). He subsequently strongly recommended that the use of computers as decision-making aids should be limited as much as possible, regardless of the technical capabilities (Weizenbaum 1976, 13; Berry 2023, 3).

This brief overview of the first 40 years of computer development initially reveals the framework within which the technological consequences could already be estimated - reliably, from today's perspective, both in terms of their potential and their risks. Turing points out a specific characteristics of the mind: Making mistakes and making surprising statements. His mathematical model of a computer (an equivalent of Church's model) started with a human computer in mind, a person whose wage labour was rapid mental arithmetic (Copeland 2004a, 40). *McCulloch* and *Pitts*' idea of an artificial neural network maybe inspired him to develop an idea of "'interfering training'" (Copeland 2004b, 403; Piccinini 2020, 108). And the intelligent machine he was thinking about later, was not a precise, reliable calculator but *a more elaborated* model that intentionally is *capable* of making *its own* mistakes which comes along with the possibility of original utterances. So human consciousness appears as blueprint for the development of highly sophisticated machine intelligence. Human consciousness arises from a sizable biological neural network, according to current understanding of about $8.6 \cdot 10^{10}$ neurons in the brain, of which about $1.6 \cdot 10^{10}$ neurons form the cerebral cortex and about $6.9 \cdot 10^{10}$ neurons make up the cerebellum (Herculano-Houzel 2009, 7). Certainly the remaining neural connections in the body will expand the entire neural network in some extent and play their part in the emergence of consciousness. Of course, an artificial neural network is something different. An artificial neuron is a mathematical function, for instance $\mathtt{n}_0(\mathtt{x})$. Its network is therefore a complex composition of functions, here: $\mathtt{n_m}(\mathtt{n_{m-1}}(\mathtt{n_{m-2}}\ldots(\mathtt{n_2}(\mathtt{n_1}(\mathbf{n_0}(\mathbf{x}))))\ldots))$ respectively $(\mathtt{n_m} \circ \mathtt{n_{m-1}} \circ \mathtt{n_{m-2}} \circ \cdots \circ \mathtt{n_2} \circ \mathtt{n_1} \circ \mathbf{n_0})(\mathbf{x})$. Such a transitive structure, in huge clusters of possibly even a bit more neurons than a human brain contains, requires a lot of computing power from the hardware for training and thus consumes disproportionately more energy than a natural neural network could ever tolerate, enough to power households; also in much smaller applications they demand a lot of computing power (see Lucchioni, Jernite, and Strubell 2024). Artificial neural networks are apparently much less efficient and unbodied. They have no physical connection with the entity that discretely realises them. In turn, the biological neural network is an integral part of the entire body, which functions only through it – continuously in the low millivolt range. In addition, in a natural neural network, many neurons are likely preset to regulate bodily functions and instinctive behavior. Their interaction with the cerebral cortex should have a significant influence on the development of consciousness. An artificial neural network has no body to control and protect, no lineage to ensure, so these components can usually be missing, probably ("[unnecessary] baggage," Arkin 1998, 32). So, all artificially defined neurons can be focused on the task of learning to match specific patterns.

However, there are indeed similarities. Untouched natural and artificial neural networks behave more error-prone than more experienced ones and unexpected outcome can be surprising and interesting statements to foreign observers. Again another difference: a human person can be asked, why she acted the way she did. Perhaps the person herself is not clear about the reasons but it is also possible that she starts to question them by herself and comes up with sound explanations for her behaviour. That's how human people start to trust each other. If a person acts predominantly reasonable, her individual behaviour can become predictable to others over the length of time they know that person. The person becomes trustworthy, if the reasons she gives are consistent with the consequences of her actions. An artificial neural network cannot observe and question itself. Perhaps requiring self-observation from an artificial neural network would be the same as expecting it from the human cerebral cortex alone. (Besides, artificial neural networks can be much less complex than natural ones.) Also the human person cannot tell which intermediate steps in that cluster lead to the own decision. Ultimately, maybe some aspects of intelligence in Turing's sense are mechanised in the meantime, but not consciousness.

It is important to realise that most of the software development products in question are not neural network applications anyway. *Artificial intelligence*, the term used "'to nail the flag to the mast'" (Moore 2006, 87), aimed since its beginnings in 1955 to improve computer programming, develop language processing, "neuron nets" and self-improving machines, work on complexity of calculation and possibly heuristic creative methods (McCarthy et al. 2006), and already five years later *Minsky* summarised the state of development at that time in “five main areas: Search, Pattern-Recognition, Learning, Planning, and Induction” (Minsky 1961, 8). Of these, it is planning (and problem-solving) to which Waldmann’s remarks primarily refer – at that time, a first basic approach to automatically proof of propositional logic theorems with a simple heuristic scheme (21–22) and a second one as a “general planning heuristic” (26). Although Minsky states that the absence of machine intelligence does not necessarily equate to a lack of computers’ possible cogitation – with a much less reverent understanding of the concept of ‘thinking’ – (27), the approaches of programmed problem-solving strategies are blind processes which were already programmable when sophisticated and in fact opaque *deep* machine learning processes with artificial neural networks (see Raschka, Patterson, and Nolet 2020, 5) were only imaginable not implementable because of insufficient hardware (until the time around 2010, 21). At most, these ‘classic’ implementations reflect the intellectual effort of human programmers, which shows its consequences in the program flow. This made computer-aided problem-solving and planning already possible using much simpler techniques that could be carried out with much less powerful computers. The solutions reached already high levels of complexity in terms of size and depth of the source code, corresponding to the intricacy of the tasks to be completed. Nonetheless, they consume much less computing hence electricity power. Therefore such classic approaches are the foundation of a vast amount of software solutions until today. They do not develop strategies autonomously after being trained in their field of operation as an initially ‘virtually blank slate’. Instead, their developers analyse this field, at best inquire experts about the field’s characteristics, and create a scheme with predetermined decision options and “*structured data* in mind” (5). ELIZA is also an application based on structured data that enables easy pattern matching in the pretended conversation (cf. Harvey 1997b, 149–151) – without necessarily being programmed in the structured programming paradigm.

A software solution for the problem of *deploying and tracking bicycle couriers as time- and cost-efficiently as possible*, which is used in Waldmann’s fictitious example, is such a classic application. The following structured data can be easily assumed: (1) A finite set of couriers abstracted as tuples that identify the real people and the automated individual ratings. (2) A stack of orders. (3) A set of constants that represent *the employers explicit* ratings of time- and cost-efficiency as modifiers for the automated rating of the employees’ performance and working times (Waldmann 2024, 9). The algorithms involved perform the intended calculations and assign orders and couriers according to the computed ranking, ultimately as a Cartesian product.

The complexity does *not* lie in the algorithms. Also the data sets can be simple one-dimensional vectors – undercomplex. Essentially, this system could be halted at any time and the current state of any datum could be read. This is a reason, why the so-called ‘micropolitically effective actions’ mentioned above are fundamentally possible. Instead, the implementation becomes difficult to oversee because of the networked large number of peripheral devices, which continuously supply the simple system with incoming new orders and recorded delivery data in order to assign suitable courier and order pairs in line with the employer's understanding of a desirable break-even point – without ever coming to rest (until the business is dissolved). The software's main task is to manage the data flow and not to compute the ratings. The complexity arises from the operation of the technical system within the social environment, of which the company is a subsystem, which uses this software as a tool to reduce the complexity to the narrow corporate purpose of exploiting stakeholders as a set of key-performance-indicators to increase the shareholder value – still complex but nevertheless greatly simplified compared to the surrounding system.

Back in the 1960s and 1970s, Weizenbaum unexpectedly observed, that his skilfully engineered but in principle rather simple conversation agent was already inspiring psychiatrists and others to envision computer-aided or completely computer-driven mass-processing of patients; it tempted users to anthropomorphise the program with which they were having serious, intimate conversations that they requested to be kept confidential, even though they had witnessed the program's development and were aware that they were using a computer; others considered ELIZA "a general solution to the problem of computer understanding of natural language" (Weizenbaum 1976, 5–7). A possible interpretation of these observations could be that the deliberately reduced complexity as a section of reality for the creation of a model is confused with a complexity that supposedly emerges from within as a new reality.

To pursue the idea speculatively: The reason for this could lie in the natural neural network that mediates human consciousness. For *Hegel*, perception is an intimate movement of the consciousness that conveys the various aspects of a perceptible thing through sensory impressions as diverse experiences of difference, which are repeatedly synthesised into an overall impression – a continuous process of analysis, synthesis, internalisation and alienation, which ultimately results in the distinction between inner impression and external object. (Hegel 2014, 99-103). This process continues in the perception of another *self*-consciousness that is recognised as a similar mind but remains another self – an ego recognizes itself by perceiving another ego as different from itself and potentially appreciates the similarity of the other (146–147). This concept of a constructive exploration of the inner and outer worlds is more than two hundred years old and was developed in response to ancient Greek philosophy. Notwithstanding, it is plausible, can be easily transferred to the functioning of neural networks and suggests corresponding processing operations of the unstructured incoming data. Perhaps, anthropomorphisation is part of this apparently heuristic procedure according to the previous training: "I made experiences with entities that can utter sentences which make sense to me and successfully recognised them as other selfs that can think in complex ways, just like me. So maybe that entity which seems to show a similar behaviour is also the same." The knowledge of the potential for this possible structure of our thinking is apparently quite old, as is probably our susceptibility to this fallacy when it comes to animals, toys, even fantasy characters and also computer programs. This may also have been partly the reason for Wiener's skepticism. It should therefore come as no surprise how early Weizenbaum tried – figuratively speaking – 'to pull the emergency break.' Maybe according to Hegel, the reason for the fallacy could be an incomplete movement of the consciousness in which alienation was interrupted after internalisation.

This preliminary assumption of an inherent need for recognition could then be reapplied at a higher level to deal with complexity. Again in the 19th century, intersubjective and intergenerational hermeneutic legal discourses could be considered as a *folk spirit* that produced the true law that could be received from this spirit through jurisprudence (Lahusen 2013, 71–72). Less romantic and probably just a choice of identifier influenced by Hegelian terminology (Schnädelbach 1983, 159) an *objective spirit* was introduced in general hermeneutics but understood as the intersubjective and intergenerational discourse with participants who were aware of the influence of their own socialisation (Danner 1998, 47–50). Thus, since about two hundred years at least, the dynamic aspect of networks of complexly interacting people has also been taken into account with metaphors that suggest a collective being, perhaps only as substitute for more appropriate descriptions to develop. *Smith*'s invisible hand, which takes care for everyone who is only concerned with their own advancement (Smith 1937, 423), comes from the previous century. Hobbes Leviathan is even older.

Later, the inherent dynamics of such collective processes however were described more connectable, related to the legal science as the Struggle for the Law whereby a metaphysical concept was abandoned in favour of the analysis and generalisation of concrete legal disputes (von Ihering 1992, 66–68). However, at the beginning of the computer age the idea of *objective* solutions to societal problems apparently proved to be an attractive means of reducing complexity and a convincing (selling) argument, along with the traditional notion of invisible hands or a collective, blindly formed, caring mind. The scepticism of the representatives of cybernetics, who were closer to the the late 19th century and who were increasingly marginalised by the enthusiasts of the artificially intelligent countermovement from the mid-1950s onwards, may have been due to the fact that, on the one hand, their socialisation probably took place in an academic discourse in which metaphysical simplifications were critically examined (see for instance Husserl 2009), whose propagandistic instrumentalisation as a means of warfare they had experienced as contemporary witnesses partly already in the First, but at least in the Second World War and its pre-war period (also a legacy of the late 19th century, see for example Schnädelbach 1983, 184–186).

As a result, technology assessment has been in line with existing historical experience of human nature from the very beginning. The proof is provided by the anthropomorphisation of the dynamics of networked computer systems that prevails in the *age of digitality* (arguably firstly outlined from Negroponte, according to Korsgaard Sorensen and Kjærgaard 2016, 4; Negroponte 1995) and is geared towards a quasi-religious view of algorithms that are vaguely equated with artificial neural networks, whose actual functionality is of far less interest than their convenience. The *invisible hand* does not show itself as a caregiver that realises the objectively best solution to social problems. It is reflected in the predictably biased results of machine learning processes, which reflect the bias of the human trainers who compile the training data sets, just as biased findings in every other discipline of science, politics and law. And it is reflected in the dynamic of mass hysteria, inherent in human history and therefore well known, amplified by limitless communication, exploited for warfare which is also refined by sophisticated, artificially intelligent, autonomous weapon systems in continuuos use, plus increased consumption of electricity that can only be met by nuclear energy and fossiel fuels – Wiener's vision has come true.

The implications of confusing the computer-assisted human Struggle for the Law with an *algorithmic authority*, which appears as the core of Waldmann's position, are briefly examined in the subsequent section in order to provide a starting point for media education, especially in primary schools.

4 Responsibility

The humanisation of algorithms follows, as seen in Burke, from feelings of powerlessness and a lack of willingness to acquire agency – based on an "argument from ignorance" (Walton, Reed, and Macagno 2013, 327): *If these algorithms were comprehensible, I could understand them. But I cannot.* or a fallacy of generalisation: *I cannot understand it because nobody can.*

This may have something to do with *movement*: in the old *Common Law* of England, the ancient legal concept that a thing could become *deodand* survived until the 19th century (Holmes 1991, 7–10). By 'thing' is meant everything that is used as a 'dependent' thing by 'fully accepted' people: tools, means of transport, farm animals and, earlier, slaves. If objects were brought to 'life' by movement or animals passed through and this caused significant damage, they were in certain cases *forfeited* and *fell to God* (*deodandum*) or to the crown or the public or even the injured party as compensation for the misfortune that had occurred; it was about *no-fault liability* in the case of damage that was not directly attributable (e.g. Waldmann 2024, 7), but which was nevertheless intended to affect *people* closely associated with the event to a clearly defined extent (Holmes 1991, 24–26). The owners were not directly obliged to pay compensation and were not subject to physical punishment or condemnation, but they lost this part of their property. Punishment and ostracism could instead affect the forfeited item, including in the form of its disposal.

This traditional strategy to solve the problem seemed increasingly absurd (House of Commons 1846), which involved pretending that a sailing ship would start to have a life of its own if (and only if) it broke free from its mooring without any further intervention, i.e. as if by itself, and was set in motion by the river current and damaged a jetty, causing injury to a person (Holmes 1991, 22–23, 26–27).

There are more legally appropriate solutions than the detour via the anthropomorphisation of a 'scapegoat'. In liability law, this example shows the increasing rationalisation of a fundamental social problem through history. With a *habitual and powerful, intraactive reality decomposition apparatus*, the opposite path is taken to the idea of a thing coming to life through (recursive loop) movement (see also Turkle 2010, 8–10). Symptomatically, a work slave, *Roboti* (*Karel Čapek*), could become a *deodand* again. A construct of the *machine habitus* or a general, posthumanist-neomaterialist-based skepticism towards the output does not lead to the cause: the enabling of *algorithmic authority* (Waldmann 2024, 11; with clearly human responsibility: Bryson 2010). There is no doubt that the cross-product of the dataset of couriers and the dataset of orders creates social facts that are concretised in the shaping of the respective working conditions of individual employees. But this does not turn the mechanical execution of the mathematical operation to implement the employer's will into a reality-transforming agential cut of an apparatus as a social actor. However, assuming this anyway does not reveal a 'deus ex machina', but rather conjures a rabbit out of a top hat that *creates* the impression of being at the mercy of computers.

From a pedagogical point of view, it seems to me to be more promising if, on the one hand, data and its algorithmic processing and generation are made tangible and describable in order to be able to critically understand the transformation `Input`→`Output`. And if, on the other hand, awareness of personal responsibility is raised by which decision-making *tools* (computer *aided* design/planning/decision-making) are effective in which way, and also to what extent individuals permit to submit to the communication processes carried out with them or to exploit them for their own benefit and to the detriment of others.

5 Conclusion and outlook

Two provoking and interesting concepts, the construction of a machine habitus and agential cuts, are used to try to sharpen a blunt idea of algorithms without allowing for technical expertise that could dampen one's own normative reservations and prove preferred ideas and the arguments based on them to be unsustainable.

This creates an "irreducible opacity" that is not necessarily inherent in algorithms in the context of machine learning, but is created by speaking about *the effect* of the technology in an external way without engaging with its nature. This is the reproduction and amplification of the self-limiting powerlessness from which the everyday term 'algorithm' arose.

The possibilities of cybernetic observation were suggested as an alternative approach to encourage an examination of the origins of the established concepts, supplemented by a connection to the historical development and the foundations in terms of the history of ideas. Without error-prone translation into and back translation from less established terminology they put the technical side into words that can be taken up in social science disciplines with understanding and may potentially enable a transdisciplinary dialogue. The path to this dialogue can already be prepared with the content-rich cybernetic terms, appropriately didactically reduced, in primary school education and linked to political education for recognizing and shaping power relations.

References

Abelson, Harold, Gerald Jay Sussman, and Julie Sussman. 1996. *Structure and Interpretation of Computer Programs.* 2nd ed. Cambridge, MA, USA: MIT Press. Accessed January 21, 2017. https://web.mit.edu/6.001/6.037/sicp.pdf.

Allert, Heidrun, and Michael Asmussen. 2017. "Bildung als produktive Verwicklung." In *Digitalität und Selbst. Interdisziplinäre Perspektiven auf Subjektivierungs- und Bildungsprozesse,* edited by Heidrun Allert, Michael Asmussen, and Richter Christoph, 27–68. Bielefeld, GER: transcript Verlag. Accessed December 6, 2024. https://doi.org/10.1515/9783839439456-004.

Anderson, Marc M. 2024. "AI as Philosophical Ideology: A Critical Look Back at John McCarthy's Program." *Philosphy & Technology* 37 (44). Accessed July 14, 2024. https://doi.org/10.1007/s13347-024-00731-1.

Arkin, Ronald C. 1998. *Behavior-Based Robotics.* Cambridge, MA, USA: MIT Press.

Ashby, W. Ross. 1957. *An Introduction to Cybernetics.* 2nd ed. London, GBR: Chapman and Hall Ltd. Accessed July 14, 2024. https://philarchive.org/go.pl?id=ASHAIT-6&proxyId=&u=https%3A%2F%2Fphilpapers.org%2Farchive%2FASHAIT-6.pdf.

Barad, Karen. 2003. "Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter." *Signs: Journal of Women in Culture and Society* 28 (3, Gender and Science: New Issues): 801–831. Accessed July 22, 2024. JSTOR: 10.1086/345321. https://www.jstor.org/stable/10.1086/345321.

Bauer, Friedrich L. 2004. "Die Algol-Verschwörung." In *Geschichten der Informatik. Visionen, Paradigmen, Leitmotive,* edited by Hans Dieter Heilige, 237–254. Berlin/Heidelberg, GER: Springer-Verlag. Accessed December 3, 2023. https://link.springer.com/chapter/10.1007/978-3-642-18631-8_10.

Berry, David M. 2023. "The Limits of Computation: Joseph Weizenbaum and the ELIZA Chatbot." *Weizenbaum Journal Of The Digital Society* 3 (3): 1–24. Accessed December 6, 2024. https://doi.org/10.34669/WI.WJDS/3.3.2.

Bryson, Joanna. 2010. "Robots Should Be Slaves." In *Close Engagements with Artificial Companions: Key Social, Psychological, Ethical and Design Issues,* edited by Yorick Willks, 63–74. Natural Language Processing. Amsterdam, NED: John Benjamins Publishing Company. Accessed July 26, 2024. https://doi.org/10.1075/nlp.8.11bry. https://www.researchgate.net/publication/250333956_Robots_Should_Be_Slaves.

Buolamwini, Joy. 2016. *Unmasking Bias.* Accessed September 8, 2023. https://medium.com/mit-media-lab/the-algorithmic-justice-league-3cc4131c5148#.lh6ftfaoa.

———. 2017. "Gender Shades: Intersectional Phenotypic and Demographic Evaluation of Face Datasets and Gender Classifiers." PhD diss., Massachusetts Institute of Technology. Accessed September 8, 2023. https://dspace.mit.edu/bitstream/handle/1721.1/114068/1026503582-MIT.pdf.

Buolamwini, Joy, and Timnit Gebru. 2018. "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification." In *Proceedings of Machine Learning Research,* edited by Sorelle A. Friedler and Christo Wilson, 81:1–15. New York, USA: PMLR. Accessed July 24, 2024. http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf.

Burke, Adam. 2019. "Occluded Algorithms." *Big Data & Society,* no. July–December, 1–15. Accessed July 20, 2024. https://doi.org/10.1177/2053951719858743.

Church, Alonzo. 1936. "An Unsolvable Problem of Elementary Number Theory." *American Journal of Mathematics* 58 (2): 345–363. Accessed December 6, 2024. https://doi.org/10.2307/2371045.

Collective, Rosetta. 2024. "Knuth's Algorithm S." *Rosetta Code,* accessed July 24, 2024. https://rosettacode.org/wiki/Knuth%27s_algorithm_S.

Copeland, B. Jack. 2004a. "Computable Numbers: A Guide." In *The Essential Turing: The Ideas That Gave Birth to the Computer Age,* edited by B. Jack Copeland, 5–57. Oxford, GBR/New York, USA: Clarendon Press.

Copeland, B. Jack. 2004b. “Intelligent Machinery (1948): Introduction.” In *The Essential Turing: The Ideas That Gave Birth to the Computer Age,* edited by B. Jack Copeland, 395–409. Oxford, GBR/New York, USA: Clarendon Press.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms.* 3rd ed. Cambridge, MA, USA: MIT Press.

Cova, Victor. 2016. “Digitality, Analogicity, and Computation.” *Theorizing the Contemporary, Fieldsights,* no. March, 24, accessed May 30, 2024. https://culanth.org/fieldsights/digitality-analogicity-and-computation.

Danner, Helmut. 1998. *Methoden geisteswissenschaftlicher Pädagogik. Einführung in Hermeneutik, Phänomenologie und Dialektik.* 4th ed. München, GER/Basel, CHE: Ernst Reinhardt Verlag.

Dijkstra, Edgar W. 1968. “Go To Statement Considered Harmful.” *Communications of the ACM* 11 (3): 147–148. Accessed July 15, 2024. https://homepages.cwi.nl/~storm/teaching/reader/Dijkstra68.pdf.

———. 1970. *Notes on Structured Programming.* Technical report T.H.-Report 70-WSK-03. Eindhoven, NED: Technische Hogeschool Eindhoven. Accessed July 15, 2024. https://www.cs.utexas.edu/~EWD/ewd02xx/EWD249.PDF.

Domkin, Vsevolod. 2021. *Programming Algorithms in Lisp: Writing Efficient Programs with Examples in ANSI Common Lisp.* New York, USA: Apress.

Downey, Allen B. 2012. *Think Bayes: Bayesian Statistics in Python.* Needham, MA, USA: Green Tea Press. Accessed March 9, 2024. https://www.greenteapress.com/thinkbayes/thinkbayes.pdf.

Everth, Thomas, and Laura Gurney. 2022. “Emergent Realities: Diffracting Barad within a Quantum-Realist Ontology of Matter and Politics.” *European Journal for Philosophy of Science* 12 (51). Accessed July 23, 2024. https://doi.org/10.1007/s13194-022-00476-8.

Graf, Erich Otto. 2010. “Strukturalismus.” In *Wissenschaftstheorie,* edited by Detlef Horster and Wolfgang Jantzen, 1:211–214. Behinderung, Bildung, Partizipation Enzyklopädisches Handbuch der Behindertenpädagogik. Stuttgart, GER: Kohlhammer.

Graham, Paul. 1993. *On Lisp.* Hoboken, New Jersey, USA: Prentice Hall. Accessed November 2, 2017. https://paulgraham.com/onlisptext.html.

Harvey, Brian. 1997a. *Computer Science Logo Style: Volume 1. Symbolic Computing.* 2nd ed. Cambridge, MA, USA: MIT Press. Accessed August 18, 2015. http://people.eecs.berkeley.edu/~bh/pdf/v1ch13.pdf.

———. 1997b. *Computer Science Logo Style: Volume 2. Advanced Techniques.* 2nd ed. Cambridge, MA, USA: MIT Press. Accessed August 18, 2015. http://people.eecs.berkeley.edu/~bh/pdf/v2ch09.pdf.

Hegel, Georg Wilhelm Friedrich. 2014. *Phänomenologie des Geistes.* 13th ed. Vol. 3. Werke. Frankfurt am Main, GER: Suhrkamp Verlag.

Herculano-Houzel, Suzana. 2009. "The Human Brain in Numbers: A Linearly Scaled-up Primate Brain." *Frontiers in Human Neuroscience* 3 (31): 1–11. Accessed December 6, 2024. http://dx.doi.org/10.3389/neuro.09.031.2009.

Holmes, Oliver Wendell jr. 1991. *The Common Law (Reprint of the 1st Edition of 1881).* New York, USA: Dover Publications, Inc.

House of Commons. 1846. "Deodands Abolition (No. 2) Bill." *House of Commons Debate (HC Deb) 11 August 1846,* Vol. 88, cc. 624-7. Accessed July 21, 2024. https://api.parliament.uk/historic-hansard/commons/1846/aug/11/deodands-abolition-no-2-bill.

Husserl, Edmund. 2009. *Philosophie als strenge Wissenschaft. Mit einer Einleitung herausgegeben von Eduard Marbach.* Vol. 603. Philosophische Bibliothek. Hamburg, Deutschland: Felix Meiner Verlag.

Knuth, Donald E. 1998. *Seminumerical Algorithms.* 3rd ed. Vol. 2. The Art of Computer Programming. Reading, MA, USA et. al.: Addision-Wesley. Accessed July 3, 2024. https://archive.org/details/artofcomputerpro0000knut/mode/2up.

———. 2005. *MMIX: A RISC Computer for the New Millennium.* Vol. 1. Fascicle 1. The Art of Computer Programming. Upper Saddle River, New Jersey, USA: Addision-Wesley.

Korsgaard Sorensen, Elsebeth, and Thomas Kjærgaard. 2016. "Designing for Dialogue and Digitality in Higher and Continuing Education." In *Proceedings of the 10th International Conference on Networked Learning,* edited by Sue Cranmer, Nina Bonderup Dohn, Marten de Laat, Thomas Ryberg, and Julie Anne Sime. Lancaster, GBR. Accessed July 19, 2024. https://www.researchgate.net/publication/303278598_Designing_for_Dialogue_and_Digitality_in_Higher_and_Continuing_Education.

Kuhn, Thomas S. 1970. *The Structure of Scientific Revolutions.* 2nd ed. Chicago, USA: The University of Chicago Press.

Lahusen, Benjamin. 2013. *Alles Recht geht vom Volksgeist aus. Friedrich Carl von Savigny und die moderne Rechtswissenschaft.* Berlin, GER: Nicolaische Verlagsbuchhandlung.

Landa, Lev N. 1969. *Algorithmierung im Unterricht. Autorisierte Übersetzung und Bearbeitung von Elske Däbritz (Original: Moskau 1966).* Berlin: Volk und Wissen Volkseigener Verlag.

Lucchioni, Sasha, Yacine Jernite, and Emma Strubell. 2024. "Power Hungry Processing: Watts Driving the Cost of AI Deployment?" In *FAccT '24: Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency,* 85–99. New York, USA: Association for Computing Machinery. Accessed December 6, 2024. https://doi.org/10.1145/3630106.3658542.

Luhmann, Niklas. 2013. *Das Recht der Gesellschaft.* 6th ed. Frankfurt a.M., Deutschland: Suhrkamp Verlag.

McCarthy, John, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. 2006. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955." *AI Magazine* 27 (4): 12–14. Accessed July 14, 2024. https://doi.org/10.1609/aimag.v27i4.1904.

McCulloch, Warren S., and Walter Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 5:115–133. Accessed July 9, 2024. https://link.springer.com/article/10.1007/BF02478259.

Minsky, Marvin. 1961. "Steps toward Artificial Intelligence." *Proceedings of the IRE* 49 (1): 8–30. Accessed July 15, 2015. https://courses.csail.mit.edu/6.803/pdf/steps.pdf.

Mooers, Calvin N., and L. Peter Deutsch. 1965. "Programming Languages for Non-Numeric Processing—1: TRAC, a Text Handling Language." *ACM '65: Proceedings of the 1965 20th national conference,* no. 4. August 1965, 229–246. Accessed July 24, 2024. https://doi.org/10.1145/800197.806048.

Moore, James. 2006. "The Dartmouth College Artificial Intelligence Conference: The Next Fifty Years." *AI Magazine* 27 (4): 87–91. Accessed July 8, 2024. https://www.researchgate.net/publication/220605256.

Negroponte, Nicholas. 1995. *Being Digital.* New York, USA: Alfred A. Knopf.

Padberg, Friedhelm, and Christiane Benz. 2021. *Didaktik der Arithmetik. fundiert, vielseitig, praxisnah.* 5th ed. Berlin, Deutschland: Springer Spektrum.

Piccinini, Gualtiero. 2020. *Neurocognitive Mechanisms: Explaining Biological Cognition.* Oxford, GBR: Oxford Academic. Accessed December 6, 2024. https://doi.org/10.1093/oso/9780198866282.003.0006.

Primas, Hans. 1994. "Realism and Quantum Mechanics." In *Logic, Methodology and Philosophy of Science,* edited by Dag Prawitz, Brian Skyrms, and Dag Westerståhl, 609–631. Amsterdam, NED: Elsevier Science. Accessed July 23, 2024. https://philsci-archive.pitt.edu/951/1/Realism%26QuantumMechanics.pdf.

Raschka, Sebastian, Joshua Patterson, and Corey Nolet. 2020. "Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence." *Information* 11 (4): 1–44. Accessed December 8, 2024. https://doi.org/10.3390/info11040193.

Russell, Stuart J., and Peter Norvig. 2010. *Artificial Intelligence: A Modern Approach.* 3rd ed. Upper Saddle River, New Jersey, USA: Pearson. Accessed December 20, 2019. https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf.

Sabbatini, Federico, and Roberta Calegari. 2024. "On the Evaluation of the Symbolic Knowledge Extracted from Black Boxes." *AI and Ethics* 4:65–74. Accessed July 27, 2024. https://doi.org/10.1007/s43681-023-00406-1.

Schnädelbach, Herbert. 1983. *Philosophie in Deutschland 1831-1933.* Frankfurt a.M., Deutschland: Suhrkamp Verlag.

Seibel, Peter. 2011a. "Object Reorientation: Generic Functions." In *Practical Common Lisp.* New York, USA: Apress. Accessed February 11, 2019. https://gigamonkeys.com/book/object-reorientation-generic-functions.

———. 2011b. "The Special Operators." In *Practical Common Lisp.* New York, USA: Apress. Accessed February 11, 2019. https://gigamonkeys.com/book/the-special-operators.

Sieniutycz, Stanisław. 2020. *Complexity and Complex Thermo-Economic Systems.* Amsterdam, NED: Elsevier. Accessed July 24, 2024. https://doi.org/10.1016/B978-0-12-818594-0.00001-5.

Smith, Adam. 1937. *An Inquiry Into The Nature And Causes Of The Wealth Of Nations.* New York, USA: Random House.

Teschl, Gerald, and Susann Teschl. 2013. *Mathematik für Informatiker. Band 1. Diskrete Mathematik und Lineare Algebra.* 4th ed. Berlin/Heidelberg, Deutschland: Springer Vieweg.

Turing, Alan M. 1936. "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proceedings of the London Mathematical Society* s2-42 (1): 230–265. Accessed April 9, 2024. https://doi.org/10.1112%2Fplms%2Fs2-42.1.230.

———. 1950. "Computer Machinery and Intelligence." *Mind* 59 (236): 433–460. Accessed September 8, 2023. JSTOR: 2251299. http://www.jstor.org/stable/2251299.

———. 2004a. "Intelligent Machinery (1948)." In *The Essential Turing: The Ideas That Gave Birth to the Computer Age,* edited by B. Jack Copeland, 410–432. Oxford, GBR/New York, USA: Clarendon Press. Accessed December 6, 2024. https://doi.org/10.1093/oso/9780198250791.003.0016.

———. 2004b. "Intelligent Machinery, A Heretical Theory (1951)." In *The Essential Turing: The Ideas That Gave Birth to the Computer Age,* edited by B. Jack Copeland, 472–475. Oxford, GBR/New York, USA: Clarendon Press. Accessed December 6, 2024. https://doi.org/10.1093/oso/9780198250791.003.0018.

Turkle, Sherry. 2010. "In Good Company? On the Threshold of Robotic Companions." In *Close Engagements with Artificial Companions: Key Social, Psychological, Ethical and Design Issues,* edited by Yorick Willks, 3–10. Natural Language Processing. Amsterdam, NED: John Benjamins Publishing Company. Accessed July 30, 2024. https://doi.org/10.1075/nlp.8.11bry. http://web.mit.edu/people/sturkle/pdfsforstwebpage/Turkle_In%20Good%20Company.pdf.

Turner, David A. 2013. "Some History of Functional Programming Languages (Invited Talk)." In *Trends in Functional Programming: 13th International Symposium, TFP 2012 St. Andrews, UK, June 12-14, 2012 Revised Selected Papers,* edited by Hans Wolfgang Loidl and Ricardo Peña, 1–21. Heidelberg, GER/New York, USA/Dordrecht, NED/London, GBR: Springer. Accessed March 10, 2024. https://link.springer.com/chapter/10.1007/978-3-642-40447-4_1.

Van der Loos, H. F. Machiel. 2014. "Sherry Turkle. Alone Together: Why We Expect More from Technology and Less from Each Other. Basic Books, New York, 2012. ISBN-13: 978-0465031467 (Book Review)." *Science and Engineering Ethics* 20:5–6. Accessed July 28, 2024. https://doi.org/10.1007/s11948-014-9524-1.

von Foerster, Heinz. 2002. "Thoughts and Notes on Cognition." In *Understanding Understanding: Essays on Cybernetics and Cognition,* edited by Heinz von Foerster, 169–189. New York, USA: Springer-Verlag. Accessed July 20, 2024. https://www.alice.id.tue.nl/references/foerster-2003.pdf.

———. 2015. "Kybernetik." In *Wissen und Gewissen. Versuch einer Brücke,* 9th ed., edited by Siegfried J. Schmidt, 72–76. Frankfurt am Main, GER: Suhrkamp Verlag.

von Ihering, Rudolf. 1992. *Der Kampf um's Recht. Zum hundertsten Todesjahr des Autors herausgegeben von Felix Ermacora.* Berlin, GER: Propyläen Verlag.

Waldmann, Maximilian. 2024. "Abwerten, Aussortieren, Separieren. Algorithmische Ungleichheit als neuartiges Gegenstandsfeld soziologischer, neomaterialistischer und medienbildungstheoretischer Positionen." *MedienPädagogik. Zeitschrift für Theorie und Praxis der Medienbildung* Themenheft 61:1–23. Accessed July 20, 2024. https://doi.org/10.21240/mpaed/61/2024.06.10.X.

Walton, Douglas, Chris Reed, and Fabrizio Macagno. 2013. *Argumentation Schemes.* 6th ed. New York, USA: Oxford University Press.

Weizenbaum, Joseph. 1976. *Computer Power And Human Reason.* New York/San Francisco, USA: W.H. Freeman and Company.

Wiener, Norbert. 1947. "A Scientist Rebels." *The Atlantic Monthly,* accessed December 6, 2024. https://cdn.theatlantic.com/media/archives/1947/01/179-1/132381596.pdf.

———. 1950. *The Human Use of Human Beings: Cybernetics and Society.* Cambridge, MA, USA: Riverside Press. Accessed July 15, 2024. https://monoskop.org/images/9/90/Wiener_Norbert_The_Human_Use_of_Human_Beings_1950.pdf.

———. 2021. *Cybernetics: Or the Control and Communication in the Animal and the Machine (Reprint of 2nd Edition 1961).* 2nd ed. Eastford, Connecticut, USA: Martino Fine Books. Accessed July 20, 2024. https://doi.org/10.7551/mitpress/11810.001.0001.