



Abstraction, mimesis and the evolution of deep learning

Jon Eklöf^{1,3} · Thomas Hamelryck¹ · Cadell Last² · Alexander Grima³ · Ulrika Lundh Snis⁴

Received: 23 June 2022 / Accepted: 4 May 2023
© The Author(s) 2023

Abstract

Deep learning developers typically rely on deep learning software frameworks (DLSFs)—simply described as pre-packaged libraries of programming components that provide high-level access to deep learning functionality. New DLSFs progressively encapsulate mathematical, statistical and computational complexity. Such higher levels of abstraction subsequently make it easier for deep learning methodology to spread through mimesis (i.e., imitation of models perceived as successful). In this study, we quantify this increase in abstraction and discuss its implications. Analyzing publicly available code from Github, we found that the introduction of DLSFs correlates both with significant increases in the number of deep learning projects and substantial reductions in the number of lines of code used. We subsequently discuss and argue the importance of abstraction in deep learning with respect to ephemeralization, technological advancement, democratization, adopting timely levels of abstraction, the emergence of mimetic deadlocks, issues related to the use of black box methods including privacy and fairness, and the concentration of technological power. Finally, we also discuss abstraction as a symptom of an ongoing technological metatransition.

Keywords Deep learning · Evolution of deep learning · Abstraction · Mimesis

1 Introduction

Deep learning is a branch of machine learning that plays a vital role in science, medicine, technology and governance. In the past decade, deep learning software frameworks (DLSFs) have emerged that make it possible to develop applications that previously required in-depth knowledge of computer science, statistics and mathematics. Typically, DLSFs are open source, i.e., they are publicly available and free to use, which promotes the spread of new deep learning methodologies through mimesis. Abstraction (from the Latin *abs*, meaning “away from” and *trahere*, meaning to

“draw”) is the process of removing characteristics in order to focus on essential characteristics (Ganascia 2015). DLSFs can be interpreted as attempts to provide increasingly higher levels of abstraction to developers of deep learning. In this article, we first quantify the progress of abstraction in deep learning by examining software projects on Github, a large software repository (GitHub 2022a, b), and subsequently discuss some of its implications.

1.1 Abstraction in deep learning

Abstraction is the process to derive general rules and concepts from individual facts or situations and from these build formal descriptions of the underlying facts or situations. “An abstraction” is the outcome of this process. Instead of accounting for every detail of each fact or situation, a higher abstraction level generalizes the relevant patterns that characterize a larger set (Langer 1953). Abstractions help us understand and explain reality in a simpler way, and can be useful and powerful when used correctly. However, they can also limit our understanding. Since abstractions simplify reality to help us focus on certain parts, they risk leaving out other important aspects. Consequently, certain nuances may be lost as particular features are distilled through abstraction.

✉ Jon Eklöf
jon.g.eklof@gmail.com
Cadell Last
https://philosophyportal.online

¹ Department of Computer Science, University of Copenhagen, Copenhagen, Denmark
² Center Leo Apostel (CLEA), Vrije Universiteit, Brussels, Belgium
³ GKN Aerospace Engines, Trollhättan, Sweden
⁴ School of Business, Economics and IT, University West, Trollhättan, Sweden

The opposite of abstraction is specification, which describes the process of breaking down general rules and concepts into concrete facts or situations. Ganascia provides the following illustrative description of abstraction and the meaning of higher abstraction levels:

“[...] figures such as triangles, spheres or pyramids are abstractions of shapes of objects, geometry is an abstraction of figures and algebraic geometry is an abstraction of geometry. In the same way, integers are abstractions of sets of objects, real numbers are abstraction of measures and associative rings—algebraic structures—are abstractions of integers and real numbers” (Ganascia 2015).

Abstraction is critical for the development of science, as a scientific theory is a kind of abstraction. Mechanics, thermodynamics, electromagnetism, relativity and quantum mechanics all correspond to various levels of abstraction in physics. Abstract concepts such as gene expression, transcription and regulation are important in biology. Also in the humanities, abstractions are widely used. Historians use concepts that abstract past reality, such as Max Weber’s conceptual apparatus, with the notions of rationalization and secularization (Ganascia 2015).

DLSFs are software libraries containing programming building blocks that provide generic deep learning functionality. They shield their users from many mathematical, statistical and computational complexities and make it possible to utilize advanced hardware such as graphics processing units (GPUs) (Demirović et al. 2018; Shi et al. 2016). DLSFs make it possible to take advantage of other programmers’ work, with only a summary understanding of the underlying principles or technologies. If needed, a DLSF’s functionality can be selectively changed by additional user-written code to provide application-specific functionality. However, this is often a complex process, as it requires the user to go below the current abstraction level.

1.2 Mimetic theory

The American-French anthropologist and cultural theorist René Girard (1923–2015) posited that human desire is mimetic: humans tend to desire objects that are desired by those around them (Girard et al. 2002; Girard 1965). Mimetic theory indicates that mimesis is a complex and generative phenomenon at the heart of human culture, religion, economy, politics and technology. Desire has a triangular nature, involving a subject, a model and an object of desire. The subject looks to a model to learn about what is desirable. In addition, desire is mutually reinforcing; one person’s desire can render an object more attractive for another person, which in turn increases the interest of the first person. This aspect of desire is called double mediation

and lies at the heart of many hypes, fashions, bubbles, trends and frenzies. Advertisement is easily understood from the point of view of mimetic theory: it aims to provide credible and attractive models that induce subjects to desire certain objects through triangular mimesis. Humans are embedded in vast networks of desire in which they act as models and subjects of desire to each other, and compete over objects. Today, such networks of desire increasingly exist in the digital realm of the internet (Girard 2009). In the context of this article, we consider the interplay of mimesis and abstraction. We argue that abstraction, as for example in the case of DLSFs, makes it easier to adopt the concepts, methods and algorithms of mimetic models that are successful, or that are perceived as successful.

2 Objective

In this article, we quantify the increase of abstraction in deep learning by examining publicly available code on Github. Our hypothesis is that, if abstraction levels within deep learning increase, the number of lines of code in deep learning projects should decrease. We view the reduction in the number of lines of code as an indication of the ongoing advancements in AI and the increasing accessibility of AI technology. The quantification of abstraction using the number of lines of code used for deep learning applications serves as a starting point for discussing its philosophical and societal implications. The main contribution of the paper is to provide an analysis of the opportunities and challenges resulting from increased abstraction. We discuss and argue the importance of abstraction in deep learning with respect to ephemeralization, technological advancement, democratization, the need to use timely abstractions, the emergence of mimetic deadlocks, issues related to the use of black box methods including privacy and fairness, and the concentration of technological power. Finally, we also discuss abstraction as a symptom of an ongoing technological metatransition.

3 Methods

Github.com is a provider of Internet hosting for software development and version control where developers store their software projects. In January 2021, when we started to collect the data for this study, Github had over 57 million users and contained over 160 million projects stored in so-called repositories (GitHub 2022a, b). To identify deep learning repositories, we searched through all these repositories using the following keywords: ‘*deeplearning*’, ‘*deep learning*’ and ‘*deep-learning*’. We identified 605 403 repositories that fit these descriptions. The intention was to

download and analyze a random sample amounting to half of these 605,403 repositories. To ensure a representative sample, we employed uniform sampling using the `np.random.choice` command in Python. This command selects a random subset of data from a larger dataset, with each data point having an equal probability of being chosen. However, after we constructed the random sample and initiated the download process (that took over eight months to complete), some of the repositories of the sample had changed from public to private or been deleted. We therefore ended up with a random sample of 317,428 repositories. After removing all instances of forked repositories (i.e., repositories copied from other repositories) we ended up with a final dataset of 37,915 repositories. From this dataset, we extracted the following information: “*date*”—when the repository was created, “*last commit*”—when the repository was last changed, “*language*”—what programming language that was used for the project and “*code_count*”—the number of lines of code used in the repository. First, we investigated the number of deep learning projects based on when they were created. Second, we investigated the evolution of the median of the number of lines of code. In Fig. 2, we plot the median of the number of lines of code (y-axis) of all projects whose last activity on GitHub occurred in the given year (x-axis). Finally, we investigated the increase in the number of projects and the reduction in number of lines of code in view of the initial release dates of the major DLSFs: Keras (version 0.2.0), TensorFlow (version 0.5.0), PyTorch (alpha-1), Jax (version 0.1.47) and TensorFlow 2.0.

4 Results

4.1 Python is the predominant programming language

Python has been the dominant programming language for deep learning projects on GitHub for as long as our data extends. As we can see in Fig. 1, between 2014 and 2018, its popularity increased steadily. Since then, Python has maintained its position as the programming language of choice for deep learning projects. Today, over 85% of the deep learning projects use Python and, as can be seen in Fig. 1, this number shows no signs of diminishing. Since Python accounts for such a large share of the deep learning projects on GitHub, we have focused our further investigations on this programming language only.

4.2 Significant increase of deep learning projects

Between 2013 and the end of 2015, the number of deep learning projects in the sample only increased slightly. Beginning in 2016, until 2018, the number of deep

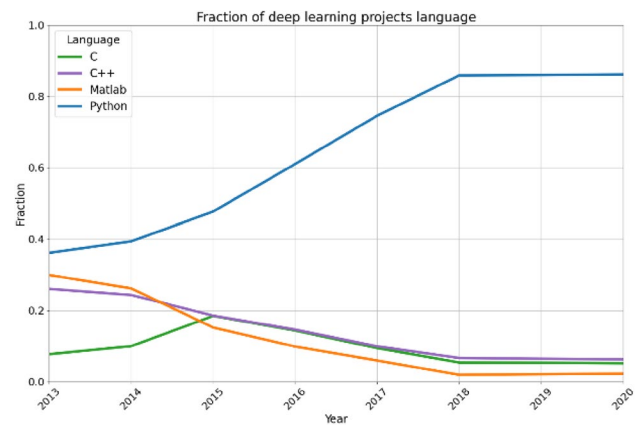


Fig. 1 Yearly development of the proportionate use of the programming languages C (green line), C++ (purple line), Matlab (orange line) and Python (blue line) for deep learning projects on GitHub. The y-axis shows the fraction of projects in each language; the x-axis shows the year of the first commit. A Github commit is a record of changes made to a code repository. It is a snapshot of the changes made to the code, typically with a brief description of the changes made

learning projects increased drastically. In Fig. 2, we can see changes in the rate at which the increase in the number of deep learning projects has occurred (orange dotted line). Some of these changes roughly coincide with the release of DLSFs in Python. The first increase in the beginning of 2015 coincides with the release of Keras (Keras 2016). The second coincides with the release of TensorFlow (Google AI 2015) and the third change that occurred at the end of 2016 coincides with the release of PyTorch, which was released in September 2016 (PyTorch 2018; GitHub and PyTorch 2022a, b).

4.3 Reduced number of lines of code

In this study, we aim to quantify the increase in abstraction using a reasonable proxy: we investigate the number of lines of code used in deep learning projects. Our hypothesis is that, as abstraction levels within deep learning increase, the number of lines of code in deep learning projects should decrease. The GitHub data showed exactly this. We could see that as the DLSFs were launched, the median number of lines of code of the deep learning projects changed. From 2014 until 2016, the median number of lines of code of active deep learning repositories on GitHub increased. However, since the end of 2016, the median number of lines of code of active deep learning repositories has decreased significantly. In 2016, the median number of lines of code used was about 3700. At the end of 2016, there was a sharp drop in the number of lines of code. By the end of 2017, the median had reduced to about 1200 lines of code. This sharp drop coincides with the release of TensorFlow and PyTorch

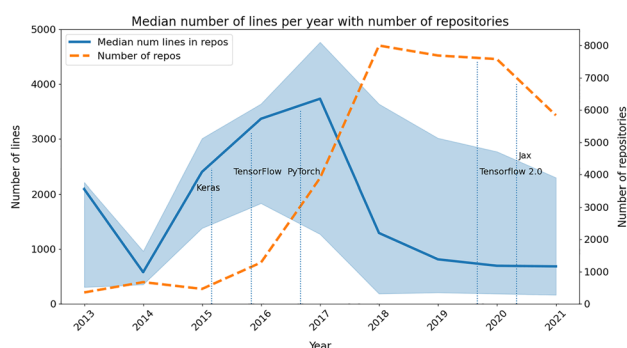


Fig. 2 Development of the median of the number of lines of Python code of deep learning projects (blue solid line) together with the number of deep learning repositories (orange dotted line) on Github. The blue shaded area shows the spread of the median number of lines using the 25th and 75th quartile of the sample. The left y-axis shows the median number of the lines of code of projects whose last commit on GitHub was in the year specified on the x-axis; the right y-axis shows the number of repositories initiated in the year specified on the x-axis. The blue dotted vertical lines display the initial release dates of the stable versions of important DLSFs (Keras 0.2.0, TensorFlow 0.5.0, PyTorch alpha-1, TensorFlow 2.0, and Jax 0.1.47)

(Google AI 2015; PyTorch 2018; GitHub and PyTorch 2022a, b). The next drop occurred at the end of 2017. After 2017, there have been no major reductions in the number of lines of code used similar to those in the end of 2016 and 2017. The decrease continues, but at a slower rate. By the end of 2020, the median was down to about 600 lines of code. This means that over the past years, the median number of lines of code has reduced by more than 80%.

5 Implications

The launch of open source DLSFs such as TensorFlow and PyTorch correlates both with significant increases in the number of deep learning projects as well as with major reductions in the number of lines of code used to build deep learning models. The reduction in the number of lines of code is an indication of the ongoing advancements in AI technology. An increasing amount of people use deep learning solutions based on abstractions that allow them to do more with less. We call this ongoing process “abstraction explosion”. We discuss the possible implications of the progress of abstraction in deep learning below.

5.1 Abstraction contributes to ephemeralization

The architect and systems theorist Buckminster Fuller (1895–1983) coined the term ephemeralization—the ability through technological advancement to do “*more and more with less and less until eventually you can do everything with nothing*” (Fuller 1938). DLSFs make it possible to build

deep learning models that can perform increasingly difficult tasks (Wang et al. 2020; Oneto et al. 2019) while requiring less and less actual code. This can be interpreted as the ephemeralization of deep learning. Even if our results indicated that the reduction of the number of lines of code used for deep learning models seems to have halted over the past years, deep learning models based on the existing DLSF can solve increasingly difficult tasks (Wang et al. 2020; Oneto et al. 2019). One explanation for the success of abstraction is the increase in the amount of available data. From 2010 to 2020, the amount of data created, captured, copied, and consumed in the world increased by almost 5000% growth, from 1.2 trillion gigabytes to 59 trillion gigabytes (Press 2020). Deep learning models rely upon large amounts of data and their predictions become more accurate the more data they are fed (Marcus 2018).

5.2 Technological advancement through feedback loops

Today, deep learning models typically run on graphics processing units (GPUs) since this shortens the training and inference time of the models compared to running the models on central processing units (CPUs) (Demirović et al. 2018; Shi et al. 2016). Using GPUs for deep learning models used to be a cumbersome process that required a lot of manual work to distribute the calculations on the GPU. Now, GPUs are easy to use for deep learning models because their complexity is effectively abstracted away (Demirović et al. 2018). As a result, more and more people use GPUs to run deep learning models. This has created a feedback loop that has promoted the further development of GPUs. This can be interpreted as an example of “the law of accelerating returns” proposed by Kurzweil; as a process becomes more effective, greater resources are available to progress that process further (Kurzweil 2001).

5.3 Fast succession of paradigm shifts

Abstraction can stimulate the emergence of new paradigmatic technologies such as deep learning, deep probabilistic programming (Bingham et al. 2019), quantum computing (Havlíček et al. 2019) and causal inference (Pearl 2000) by making them comparatively easy to use. New solutions involving so-called “low coding” or even “no coding” (low-code, no-code) are already replacing conventional software development (Kaye 2022; Economist 2022). For example, through its ChatGPT model, OpenAI has launched a system that is able to translate natural language into multiple programming languages (OpenAI 2023). This takes coding to a completely different level since it makes it possible to build models without any actual writing of code. To some extent, it even renders our method in this study, trying to quantify

the abstraction explosion by counting lines of code, obsolete. Tying such high levels of abstraction with emerging technologies can promote the emergence of rapid and decisive paradigm shifts.

5.4 Democratization of deep learning

Many DLSFs, including TensorFlow and PyTorch, are open source DLSFs, which means that they are free for anyone to use, modify and redistribute. The release of open source software, along with a broader socio-cultural shift towards participation in media and cultural production, increases the opportunities for democratization of production, governance and knowledge exchange (Powell 2012). The DLSFs have made it much easier for people to collaborate and build on each other's work, allowing for a surge in technological innovation and progress (Bommasani et al. 2021).

In addition, DLSFs provide access to technologies that previously required in-depth knowledge of mathematics, statistics and programming, which lowers the demands with respect to required education. This, in turn, facilitates faster adoption of technology by a wider range of users of DLSFs, as specialized or in-depth knowledge is not required. We note that, although the DLSFs have democratized deep learning, access to the most advanced deep learning models are moving in the opposite direction. Models such as GPT-3 are only accessible through an API, which is shared with a limited pool of people (Bommasani et al. 2021).

5.5 Adopting timely levels of abstraction

The rapid progress of abstraction in DLSFs makes it increasingly important to use the latest programming frameworks that allow adopting timely, appropriate levels of abstraction. Failing to do so, companies or whole societies could waste time and resources using obsolete abstractions and ultimately risk being outcompeted by those who use a more timely level of abstraction. Because of this rapid development, keeping up with the abstraction explosion is a challenge in itself and requires a commitment to learning about new, emerging levels of abstraction. This commitment requires a proactive mindset and the aptitude to acknowledge one's limitations in order to identify the need for further knowledge. In addition, the commitment entails fostering critical thinking skills and the ability to go beyond superficial understandings of technological advancements to evaluate their potential implications and limitations. Such commitment requires adequate resources in terms of money, time and cognitive capacity. Keeping up with the progress of abstraction will remain a challenging task in governance, industry, medicine, the military and academia.

5.6 Mimetic deadlocks

Increased abstraction has simplified the adoption and use of new methods and the democratization has made them publicly available. This risks leading to dead ends if solutions are solely sought within the familiar and convenient limits of the abstractions that everybody is using. We call this kind of unproductive herd behavior a "mimetic deadlock". The reason we risk ending up in these deadlocks is that we let others guide our behavior. Girard describes our tendency to rely on others to determine how we should behave:

"Man is the creature who does not know what to desire, and he turns to others in order to make up his mind. We desire what others desire because we imitate their desires." (Girard 1988).

To some extent, the DLSFs that we discuss in this article are a result of beneficial mimesis, where groups of individuals commonly engage in collective practices to generate new knowledge and understanding. The application interfaces of TensorFlow, PyTorch and Jax all are based on the older software framework NumPy. In 1935, Ludwik Fleck (1896–1961), a Polish physician, microbiologist, and philosopher of science, referred to such a mimetic process in science as a "*thought collective*" (Sady 2001). The democratization of deep learning and the rise of collaboration and community-driven development point to the formation of such a thought collective. As more people are able to access and contribute to the development of deep learning, new thought collectives might emerge that may shape the future of the field.

As Fleck already understood, mimesis also risks narrowing interest and constraining innovation. Bommasani et al. point out that we see a homogenization in the selection of models based on these DLSFs and how they are used (Bommasani et al. 2021). Rather than finding the best way to solve a problem, developers risk converging to the use of algorithms and methods used by successful models around them. This can lead to a mimetic deadlock: a convergence to an agreed convenient solution that is not necessarily optimal or even good. Abstraction exacerbates this phenomenon as it makes it easy to copy without understanding the underlying issues.

Further, the interplay between abstraction and mimesis risks creating a feedback loop where the simplified representations of reality are further imitated. This can lead to a gradual reduction of nuance and complexity, contributing to a homogenized and streamlined perception of reality. As people are exposed to these simplified and imitative versions of reality, they may begin to adopt a more uniform way of thinking and perceiving the world. This can hinder their ability to appreciate the full complexity and diversity of reality, ultimately limiting their understanding and perspectives.

5.7 The black box problem

Often, we cannot account fully for how deep learning algorithms make their decisions (Castelvecchi 2016) but are only informed of the final decisions made by the algorithm (Holweg et al. 2022). Higher abstraction levels hide part of the computational complexity of the lower levels. This makes it increasingly difficult both to confirm that the models do what they are supposed to do and to troubleshoot them. Bommasani et al. argues that any flaws in the lower levels of abstraction “*are blindly inherited*” by the higher levels of abstraction (Bommasani et al. 2021). For example, rounding errors can change the topological properties of activation functions in deep networks, paradoxically resulting in improved performance (Naizat et al. 2020). In other words, the improved performance results from an error deeply hidden in layers of abstraction. Such phenomena obviously make the correct interpretation of the performance of machine learning models challenging.

Not knowing if the abstraction level works as intended could have serious implications, particularly where deep learning applications make decisions that are directly affecting human well-being or are used in areas where malfunctions could have fatal consequences, such as the aeronautics or nuclear industry (Holweg et al. 2022; Ganascia 2015). It can also amplify already existing problems with deep learning in terms of fairness, privacy and quality assurance problems.

5.8 Fairness, privacy and quality assurance problems

Decreased explainability in combination with the democratization of deep learning increases the risk of fairness, privacy and quality assurance problems. Even without the democratization of deep learning, the algorithms can amplify societal stereotypes (Wang et al. 2019; Bommasani et al. 2021). For example, a recruiting tool for the fields of science, technology, engineering and math jobs believes men are more qualified and shows bias against women (Kiritchenko and Mohammad 2018), and facial recognition software has proven to perform poorly for females with darker skin (Buolamwini 2018). However, there might be multiple conditions of fairness that (with mathematical certainty) cannot be satisfied at the same time (Kleinberg et al. 2017). Increasing fairness with respect to one condition can thus very well result in decreasing fairness with respect to other conditions (Kleinberg et al. 2017).

When deep learning becomes increasingly accessible and easier to use, and more organizations use it to increase efficiency and effectiveness, the number of cases where its application violates social norms and values rises (Holweg et al. 2022). Through the abstraction explosion, non-experts

have access to tools to create deep learning solutions. However, non-experts often lack fundamental skills in standard processes for debugging as well as testing for quality control and reliability. They also lack adequate strategies to deal with the effects of malfunctioning algorithms. Most organizations are not strategically prepared on how to respond to AI failures. If people are using the increased abstraction levels without being able to understand if they work as intended, or what they may or may not be used for, they can unintentionally unleash “unfair” algorithms.

Attempts have been made trying to combat this increased risk of “unfair” algorithms. Singapore has released a software toolkit aimed at helping financial institutions ensure they are using AI responsibly (Yu 2022). Although this to some extent may work to increase the transparency of algorithms, it does not correct the root cause of the problem where the algorithms in some way are malfunctioning. Correcting errors of deep learning models requires an understanding of the underlying abstraction levels, which in its turn often requires an in-depth understanding of both math and programming.

5.9 Abstraction introspection and informative error messages

The difficulty in correcting malfunctioning algorithms is amplified further by the fact that the error messages generated by malfunctioning code are often vague, imprecise, confusing and at times seemingly incorrect—particularly for novices (Brown 1983). The professor in human-computing interaction Amy J. Ko from Washington University, has even referred to programming languages as the least usable, but most powerful human-computer interfaces ever invented (Ko 2014). Incomprehensible error messages break down the ability to see through layers of abstraction and are thereby often unhelpful in correcting the code. This is particularly problematic as these messages act as the primary source of information to help novices rectify their mistakes (Becker et al. 2019). For the average person to be able to use error messages to correct malfunctioning code requires another level of abstraction that allows the algorithms themselves to “reflect” about their processes and—if not rectifying itself—at least provide an understandable explanation of the problem that enables the user to rectify the problem. However, with today’s abstraction levels, understanding error messages is comparable to the difficulty of reading source code (Becker, et al. 2019). Consequently, external experts that may be hard to find are often needed to solve these issues.

5.10 Experts are needed but scarce

When abstractions fail or turn out to be inadequate, the challenge is to “*make the abstraction more concrete*”,¹ moving from abstraction to the underlying specific abstraction levels (Ganascia 2015). The average user has limited time, resources, competence and cognitive capacity to deal with the complexities of the DLSF other than on a superfluous level. As we have seen in this study, increased abstraction leads to more and more people using the higher levels of abstraction. Further, since non-experts often lack fundamental skills in standard processes for debugging as well as testing for quality control and reliability an increasing amount of errors will go unnoticed. In addition, when something goes wrong, comparatively few users will know how to correct these errors. Heylighen has pointed out that increasing system complexity and information overload can negate the advantages of ephemeralization since it puts an increasing amount of pressure on the reducing number of experts able to control the ephemeralized systems (Heylighen, Accelerating Socio-Technological Evolution: from ephemeralization and stigmergy to the global brain, 2007). With just 25 million people around the world fluent in standard programming languages, the proportionate number of people able to correct errors in higher abstraction levels decreases as abstraction increases (Economist 2020). The scarcity of expertise is worsened by the fact that the industry provides higher salary than academia and often a more satisfying work environment (Wolff et al. 2020; Woolston 2021). In addition, academia rarely has access to the most advanced deep learning models (Bommasani et al. 2021).

Experts are, therefore, more likely to choose an industry career rather than pursuing academic careers in statistics, computer science or mathematics. However, deep learning builds upon these traditional sciences and it is to these fundamental levels we need to revert when something goes wrong. Ultimately, we may end up in a situation where the experts able to correct malfunctioning abstraction levels are exceedingly hard to find. If there are any experts left, they become increasingly expensive to hire.

5.11 Abstraction is power

Influencing the progress of abstraction and having access to experts when developing and using these abstractions becomes a considerable source of power. The philosopher Bertrand Russell pointed to the connection between abstraction and practical power:

“A financier, whose dealings with the world are more abstract than those of any other 'practical' person, is also more powerful than any other practical person. Financiers can deal in wheat or cotton without needing ever to have seen either: all they need to know is whether the price will go up or down. This is abstract mathematical knowledge, at least as compared to the knowledge of the agriculturist. Similarly the physicist, who knows nothing of matter except certain laws of its movements, nevertheless knows enough to be able to manipulate it.” (Russell 2009).

Initially, academia often developed new levels of abstraction within deep learning, such as when the Montreal Institute for Learning Algorithms at University of Montreal developed the DLSF Theano (Brownlee 2016).² To some extent, this guaranteed a certain level of transparency on the considerations made when producing the abstraction levels. Today, large corporations have outcompeted academia in the development of new abstraction levels (Bommasani et al. 2021; Economist 2020). One such example is the discontinuation of development of DLSF Theano. According to the development team behind Theano, it was outcompeted because of “*strong industrial players*” (Lamblin 2017).

These large corporations influence and steer the democratization of deep learning by developing and releasing DLSFs to the public. As such, they also act as mimetic models, both to other corporate agents and academia, pointing to specific abstractions and the types of problems they can solve. In addition, the same companies control a large portion of the data that is shared on the internet (Norwegian Consumer Council 2018) and they employ an increasing fraction of experts (Bommasani et al. 2021). Abstraction, data and the experts to make use of them are thus sources of considerable mimetic and practical power.

6 Conclusions and outlook

In this article, we quantify the increase of abstraction in deep learning by examining publicly available code on Github and discuss its philosophical and societal implications.

A metasystem integrates a number of initially independent components and creates a qualitative different system by steering or controlling their interactions (Turchin 1977; Heylighen 2003). Our analysis highlights the importance of considering the broader societal and philosophical implications of the abstraction explosion in deep learning, as it may signify a larger technological transition that moves from

¹ Note that this is akin to the abstract, the negative and the concrete of Hegelian dialectics.

² See for example the groundbreaking research by Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. (Krizhevsky et al. 2012).

the physical into the digital metasystem with consequences for how we understand and interact with technology (Last 2017). Turchin and Heylighen have referred to such major changes as metasystem transitions (Turchin 1977; Heylighen 2003). Several transitions have followed the same process through key evolution in information storage and replication (Szathmáry et al. 1995; Maynard Smith and Szathmáry 1997; Szathmáry 2015; Turchin 1977). One of the drivers of the current metasystem transition is an increase in abstraction, but also what Kurzweil refers to as the law of accelerating returns (Kurzweil 2001). As abstraction increases, feedback loops will lead to more and more people using these new tools, driving the progress further. Reaping the benefits of abstraction, while avoiding associated potential problems such as mimetic deadlock, black box issues concerning privacy and fairness, and excess power concentration, will remain an ongoing challenge in the foreseeable future for academia, industry and governance.

Acknowledgements We acknowledge funding from the University of Copenhagen, Department of Computer Science, Denmark; GKN Aerospace Engines, Sweden and the Center Leo Apostel (CLEA), Vrije Universiteit Brussel, Belgium.

Funding Open access funding provided by Royal Danish Library.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest On behalf of all the authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Becker B, Denny P, Pettit R, Bouchard D, Bouvier D, Harrington B, Prather J (2019) Compiler error messages considered unhelpful: the landscape of text-based programming error message research. IN: Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education, pp 177–210
- Bingham E, Chen J, Jankowiak M, Obermeyer F, Pradhan N, Karaletsos T, Goodman N (2019) Pyro: deep universal probabilistic programming. *J Mach Learn Res* 20:973–978
- Bommasani R, Liang P, Hudson D, Adeli E, Altman R, Arora SV, Etchemendy (2021) On the opportunities and risks of foundational models. arXiv preprint, <http://arXiv.org/2108.07258>
- Brown P (1983) Error messages: the neglected area of the man/machine interface. *Commun ACM* 26(4):246–249
- Brownlee J (2016) Deep learning with python: develop deep learning models on Theano and TensorFlow using Keras. In: Machine learning mastery
- Buolamwini JG (2018) Gender shades: intersectional accuracy disparities in commercial gender classification. In: Proceedings of machine learning research, 77–91
- Castelvecchi D (2016) Can we open the black box of AI? *Nat News* 538(7623):20–23. <https://doi.org/10.1038/538020a>
- Demirović D, Skejić E, Šerifović-Trbalić A (2018) Performance of some image processing algorithms in Tensorflow. In: 25th International Conference on Systems, Signals and Image Processing (IWSSIP), 1–4. <https://doi.org/10.1109/IWSSIP.2018.8439714>
- Economist (2020) Businesses are finding AI hard to adopt. Retrieved March 15, 2023, from <https://www.economist.com/technology-quarterly/2020/06/11/businesses-are-finding-ai-hard-to-adopt>
- Economist (2022) What if all workers wrote software, not just the geek elite? *Economist*. Retrieved March 15, 2023, from <https://www.economist.com/business/2022/01/29/what-if-all-workers-wrote-software-not-just-the-geek-elite?giftId=e827aa1a-7c4e-4ca0-85c3-f9d2953ab719>
- Fuller B (1938) Nine chains to the moon. Anchor Books, 1938, 1973. Anchor Books
- Ganascia J-G (2015) Abstraction of levels of abstraction. *J Exp Theor Artif Intell* 27(1):23–35
- Girard R (1965) Deceit desire and the novel. Johns Hopkins Press
- Girard R (1988) Generative Scapegoating. Violent origins: Walter Burkert, René Girard, and Jonathan Z. Smith on Ritual Killing and cultural formation. Stanford University Press, pp 73–105
- Girard R (2009) Battling to the end: conversations with Benoît Chantre. MSU Press
- Girard R, Oughourlian J-M, Lefort G (2002) Things hidden since the foundation of the world. Stanford University Press
- GitHub (2022a) *About GitHub*. Retrieved Feb 14, 2022a, from <https://github.com/about>
- GitHub (2022b) *PyTorch*. Retrieved Feb 02, 2022b, from GitHub: <https://github.com/pytorch/pytorch/releases/tag/v0.1.1>
- Google AI (2015) Google AI Blog. Retrieved 02 Apr 2022, from <https://ai.googleblog.com/2015/11/tensorflow-googles-latest-machine.html>
- Havlíček V, Córcoles A, Temme K, Harrow A, Kandala A, Chow J, Gambetta J (2019) Supervised learning with quantum-enhanced feature spaces. *Nature* 567(7747):209–212. <https://doi.org/10.1038/s41586-019-0980-2>
- Heylighen F (2003) Introduction to principia cybernetica. Retrieved February 09, 2022, from Principia Cybernetica Web: <http://pespmc1.vub.ac.be/INTRO.html>
- Heylighen F (2007) Accelerating socio-technological evolution: from ephemeralization and stigmergy to the global brain. *Globalization as evolutionary process*. Routledge, pp 304–329
- Holweg M, Younger R, Wen Y (2022). The reputational risks of AI. *Calif Manag Rev Insights*, 64(2)
- Kaye K (2022) Should just anyone be given keys to the AI machine? Why low-code AI tools pose new risks. (protocol.com) Retrieved Mar 15, 2023, from <https://www.protocol.com/enterprise/no-low-code-ai-risks>
- Keras (2016) Keras Blog. Retrieved Feb 14, 2022, from <https://blog.keras.io/introducing-keras-10.html>
- Kiritchenko S, Mohammad S (2018) Examining gender and race bias in two hundred sentiment analysis systems. In: Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM), New Orleans)

- Kleinberg J, Mullainathan S, Raghavan M (2017) Inherent trade-offs in the fair determination of risk scores. In: 8th Innovations in Theoretical Computer Science Conference, pp. 43:1–43:23. <https://doi.org/10.4230/LIPIcs.ITCS.2017.43>
- Ko AJ (2014) Programming languages are the least usable, but most powerful human-computer interfaces ever invented. Retrieved Feb 14, 2022, from <https://medium.com/bits-and-behavior/programming-languages-are-the-least-usable-but-most-powerful-human-computer-interfaces-ever-invented-7509348dedc>
- Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 25:1097–1105
- Kurzweil R (2001) The law of accelerating returns. Retrieved Mar 15, 2023, from <https://www.kurzweilai.net/the-law-of-accelerating-returns>
- Lamblin P (2017) Theano Users Google group. Retrieved Feb 14, 2022, from <https://groups.google.com/g/theano-users/c/7Poq8BZutbY>
- Langer S (1953) *Feeling and form: a theory of art developed from philosophy in a new key*. Charles Scribner's Sons, New York
- Last C (2017) Global commons in the global brain. *Technol Forecast Soc Chang* 114:48–64. <https://doi.org/10.1016/j.techfore.2016.06.013>
- Marcus G (2018) Deep learning: a critical appraisal. arXiv preprint, <https://arxiv.org/1801.00631>. Retrieved Mar 15, 2023, from <https://arxiv.org/ftp/arxiv/papers/1801/1801.00631.pdf>
- Maynard Smith J, Szathmáry E (1997) *The major transitions in evolution*. Oxford University Press
- Naizat G, Zhitnikov A, Lim L-H (2020) Topology of deep neural networks. *J Mach Learn Res* 21(1):7503–7532
- Norwegian Consumer Council (2018) Deceived by design, how tech companies use dark patterns to discourage us from exercising our rights to privacy. Retrieved Mar 15, 2023, from <https://fil.forbrukerradet.no/wp-content/uploads/2018/06/2018-06-27-deceived-by-design-final.pdf>
- Oneto N, Anguita L, Navarin A, Sperduti D (2019) Recent advances in big data and deep learning. In: *Proceedings of the INNS Big Data and Deep Learning Conference INNSBDDL2019*, held at Sestri Levante, Genova, Italy 16–18 April 2019
- OpenAI (2023) Introducing ChatGPT. (openAI.com) Retrieved Mar 15, 2023, from <https://openai.com/blog/chatgpt>
- Pearl J (2000) *Causality: models, reasoning and inference*, vol 19.2. Cambridge University Press, Cambridge
- Powell A (2012) Democratizing production through open source knowledge: from open software to open hardware. *Media Cult Soc* 34(6):691–708
- Press G (2020) 54 predictions about the state of data In: 2021. Retrieved Mar 15, 2023, from [forbes.com: https://www.forbes.com/sites/gilpress/2021/12/30/54-predictions-about-the-state-of-data-in-2021/?sh=6af0ecb0397d](https://www.forbes.com/sites/gilpress/2021/12/30/54-predictions-about-the-state-of-data-in-2021/?sh=6af0ecb0397d)
- PyTorch (2018) PyTorch blog. Retrieved Feb 14, 2022, from <https://pytorch.org/blog/a-year-in/>
- Russell B (2009) *The ABC of relativity*. Routledge
- Sady, W. (2001). Ludwik Fleck—Thought Collectives and Thought Styles. In *Polish philosophers of science and nature in the 20th century* (pp. 197–205). Brill.
- Shi S, Wang Q, Xu P, Chu X (2016) Benchmarking state-of-the-art deep learning software tools. In: 7th International Conference on Cloud Computing and Big Data (CCBD), 99–104. <https://doi.org/10.1109/CCBD.2016.029>
- Szathmáry E (2015) Toward major evolutionary transitions theory 2.0. *Proc Natl Acad Sci* 112(33):10104–10111. <https://doi.org/10.1073/pnas.1421398112>
- Szathmáry E, Maynard Smith J (1995) The major evolutionary transitions. *Nature* 374(6519):227–232. <https://doi.org/10.1038/374227a0>
- Turchin V (1977) *The phenomenon of science: a cybernetic approach to human evolution*. Columbia University Press, New York
- Wang T, Zhao J, Yatskar M, Chang K, Ordonez V (2019) Balanced datasets are not enough: estimating and mitigating gender bias in deep image representations. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*
- Wang X, Zhao Y, Pourpanah F (2020) Recent advances in deep learning. *Int J Mach Learn Cybern* 11(4):747–750
- Wolff G, Anderson J, Viry P (2020) Europe has an artificial-intelligence skills shortage. Retrieved Mar 15, 2023, from <https://www.bruegel.org/2020/08/europe-has-an-artificial-intelligence-skills-shortage/>
- Woolston C (2021) Stagnating salaries present hurdles to career satisfaction. *Nature* 599:519–521. <https://doi.org/10.1038/d41586-021-03041-0>
- Yu E (2022) Singapore releases software toolkit to guide financial sector on AI ethics. Retrieved Mar 15, 2023, from <https://www.zdnet.com/article/singapore-releases-software-toolkit-to-guide-financial-sector-on-ai-ethics/>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.