Journal of Big Data

SURVEY PAPER Open Access

A survey on data-efficient algorithms in big data era

Amina Adadi*

*Correspondence:
Amina.adadi@gmail.com
ISIC Research Team, L2MI Laboratory, Moulay Ismail University, Meknes, Morocco

Abstract

The leading approaches in Machine Learning are notoriously data-hungry. Unfortunately, many application domains do not have access to big data because acquiring data involves a process that is expensive or time-consuming. This has triggered a serious debate in both the industrial and academic communities calling for more data-efficient models that harness the power of artificial learners while achieving good results with less training data and in particular less human supervision. In light of this debate, this work investigates the issue of algorithms' data hungriness. First, it surveys the issue from different perspectives. Then, it presents a comprehensive review of existing data-efficient methods and systematizes them into four categories. Specifically, the survey covers solution strategies that handle data-efficiency by (i) using non-supervised algorithms that are, by nature, more data-efficient, by (ii) creating artificially more data, by (iii) transferring knowledge from rich-data domains into poor-data domains, or by (iv) altering data-hungry algorithms to reduce their dependency upon the amount of samples, in a way they can perform well in small samples regime. Each strategy is extensively reviewed and discussed. In addition, the emphasis is put on how the four strategies interplay with each other in order to motivate exploration of more robust and data-efficient algorithms. Finally, the survey delineates the limitations, discusses research challenges, and suggests future opportunities to advance the research on data-efficiency in machine learning.

Keywords: Data hungry algorithms, Data-efficiency, Small sample learning, Transfer learning, Data augmentation

Introduction

Building machines that learn and think like humans is one of the core ambitions of Artificial Intelligence (AI) and Machine Learning (ML) in particular. On the quest for this goal, artificial learners have made groundbreaking accomplishments in many domains spanning object recognition, image processing, speech recognition, medical information processing, robotics and control, bioinformatics, natural language processing (NLP), cybersecurity, and many others. Their success has captured attention beyond academia. In industry, many companies such as Google and Facebook devoted active research instances to explore these technologies.

Ultimately, AI has succeeded to speed up its pace to be like humans and even defeats humans in some fields. AlphaGo [1] defeats human champions in the ancient game of

Springer Open

Go. The deep network ResNet [2] obtains better classification performance than humans on ImageNet. And, recently, Google launched Meena [3] a human-like AI chatbot that can conduct sensible and specific conversations like humans. However, there is another side to this coin, the impressive results achieved with modern ML (in particular by deep learning) are made possible largely by the use of huge datasets. For instance, DeepMinds's AlphaGo used more than 38 million positions to train their algorithm to play Go. The ImageNet database used by ResNet contains about 1.2 million labeled examples. And Meena has been trained on a massive 341 GB corpus, the equivalent of roughly 341,000 books, far more than most people read in a lifetime. This is obviously far from human-like learning. One thing that makes human learners so efficient is that we are active, strategic information-seekers. We learn continually and we make use of our previous experiences. So far, it is not clear how to replicate such abilities into artificial learners. In the big data era, algorithms continue to be more data-hungry, while real facts indicate that many application domains can often only use a few data points because acquiring them involves a process that is expensive or time-consuming.

Consequently, many researchers and engineers began to recognize that the progress of ML is highly dependent on the premise of the availability of large number of input samples (generally with annotations). Without massive data, ML success is uncertain. Hence, key ML researchers have sounded a cautionary note regarding the data hunger behavior of algorithms. In his controversial work "Deep Learning: A Critical Appraisal" [4], Marcus listed ten concerns about deep learning research, at the top of the list data hungriness. he noted that "in problems where data are limited, deep learning often is not an ideal solution". Data-hungriness was also included in the unsolved problems in AI research, described in the book "Architects of Intelligence" by Martin Ford [5]. Most of the experts interviewed in this book are calling for more data-efficient algorithms. For instance, Oren Etzioni quoted in the book that "stepping stone [towards AGI] is that it's very important that [AI] systems be a lot more data-efficient. So, how many examples do you need to learn from?" [5, p. 502].

As such, our work extends the recent call for more research on data-efficient algorithms. In fact, we view these concerns as an opportunity to examine in-depth what it means for a machine to learn efficiently like humans, what are the efforts deployed to alleviate data- hungriness, and what are the possible research avenues to explore. Studying data hungriness of ML algorithms is unfortunately a topic that has not yet received sufficient attention in the academic research community, nonetheless, it is of big importance and impact. Accordingly, the main aim of this survey is to stimulate research on this topic by providing interested researchers with a clear picture of the current research landscape.

Prior to this paper, we know of few works that attempted to investigate the issue of data-hungriness. Shu et al. [6] proposed a survey that covers learning methods for small samples regime, they focused on concept learning and experience learning. Wang et al. [7] surveyed few-shot learning by putting on light methods operating at the level of data, model, and algorithms. Qi et al. [8] discussed small data challenges from unsupervised and semi-supervised learning perspectives, they presented an up-to-date review of the progress in these two paradigms. As a matter of fact, existing surveys are limited in the way they approach the problem and in the scope they cover. In contrast, our work seeks comprehensiveness, we tackle the issue from an interdisciplinary perspective and discussed potential solutions from different backgrounds and horizons. In doing so, we brought different concepts under one roof that were never discussed together and tried to draw connections between them. Furthermore, as all AI players are concerned by the issue, while elaborating the survey we deliberately tried to make it accessible to the non-theoretician while still providing precise arguments for the specialist. In this respect, we make three main contributions:

- We propose a comprehensive background regarding the causes, manifestations, and implications of data-hungry algorithms. For a good understanding of the issue, particular attention is given to the nature and the evolution of the data/algorithm relationship.
- Based on an analysis of the literature, we provide an organized overview of the proposed techniques capable of alleviating data hunger. Through this overview, readers will understand how to expand limited datasets to take advantage of the capabilities of big data.
- In our discussions, we identify many directions to accompany previous and potential future research. We hope it inspires more researchers to engage in the various topics discussed in the paper.

Accordingly, the remainder of the survey is organized as follows. "Background" section presents a preliminary background. "Review" section surveys existing solutions and organizes surveyed approaches according to four research strategies. "Discussion" section discusses research directions and open problems that we gathered and distilled from the literature review. Finally, "Conclusion" section concludes this survey.

Background

Data and algorithms: a match made in heaven

Communicating, working, entertaining, traveling and other daily life activities perfectly exemplify the fruit of combining algorithms and data. Notwithstanding, in order to dig deep into the complex nature of the link between these two concepts, we must look below to understand its germination. In this section, we provide an exhaustive investigation of data/algorithms link. The link is developed in light of four perspectives: (i) by examining the historical trajectory followed by the two concepts, (ii) by drawing insights from biology, as the two concepts can be observed both in nature and in digital, (iii) by studying the related technical and theoretical background, (iv) and by identifying business motivations that feed this link.

Historical perspective

Because the historical evolution in form and content of "Data" and "Algorithm" concepts has a poignant bearing on the issue, it is worth exploring the intellectual history of these two concepts in order to illustrate how they are intertwined. A curious fact about our data-obsessed era is that little is known about the origin and the evolution of this vital concept. Indeed, it is common to think of "big data", "machine learning" and related technologies as relatively modern technologies. Yet the roots across these domains, that gave rise to the spectacular advances we are witnessing today, are often not well known and have never been assembled in a single work, to be studied or analysed.

Presumably, the word "data" began to be used around the 17th century [9], it is derived from the Latin meaning "facts given or granted". The concept of data as a given has been criticized by many social scientists [9, 10], they claimed that the concept symbolizes rather "the facts taken or observed from nature by the scientist" and should instead be characterized as "capta" which means taken and constructed. It was mainly the invention of Gutenberg's Printing Press upon the Renaissance that helped the rapid transmission of ideas and expanded access to knowledge. People started then to use "data" originally to refer to "facts given as the basis for calculation in mathematical problems". The age of Enlightenment ushered in many fields and disciplines like economics, biology, and political science which accentuated the proliferation of data. Centuries of data sharing and scientific methodologies have led to the emergence of a new field: Statistics, data became to refer, at the time, to "numerical facts collected for future reference".

Meanwhile, the word "algorithm" was derived directly from the 9th century mathematician al-Khwarizmi, the author of the oldest work of algebra "al-Mukhtasar fi Hisâb al-jabr wa l-Muqabala" [11]. Beyond the etymological roots of the term, the concept of "Algorithm" can be traced back to the third millennium BCE. The first examples of algorithms can be found in Babylonian tablets and in Egyptians scrolls [12]. In the 12th century, the book of al-Khwarizmi was translated from Arabic to Latin and so the denary numeration system began to spread throughout medieval Europe under the name of "algorismus". Around the 18th century the term "algorismus" became the modern "algorithm". The use of the word has also evolved to include arithmetic procedures for solving problems or performing tasks [13].

A major series of achievements came during the industrial revolution (IR) with intense investment in technological innovations. This was a turning point for "data" and "algorithms" that henceforth will be used synergistically to produce tools that indented to substitute human labour: Machines [14]. Basically, data were fed to machines and algorithms guided their actions. During this period, capturing and recording data practice was developed through Jacquard's loom and Hollerith's tabulating machines. The punched cards, used at the time, were the primary data entry medium. Furthermore, a myriad of algorithms was developed at the time to automate all sorts of human actions. The first algorithm meant to be executed on a machine was created by Ada Lovelace in 1843. In 1847, George Boole invented binary algebra, the basis for modern computer code. In 1888, Giuseppe Peano established the axiomatization of mathematics in a symbolic language.

The 20th century was the digital era par excellence. It marks the modern evolution of "data" and "algorithm" concepts. Driven by advances in electronics, the concept of "algorithm" was formalized in 1936 through Alan Turing's Turing machines and Alonzo Church's lambda calculus, which in turn formed the foundation of computer science [15]. Data took now a digital form and it refers to collections of binary elements transmittable and storable by which computer operations (implemented algorithms) are performed. This opened up the evolution of data modelling for databases [16], programming languages and paradigms [17], and as foreseen by Moore [18], computing power has exposed, resulting on sophisticated tools and methods to store and process data, and complex algorithms with higher level of computational prowess. The 1990s brought a real communication revolution with the arrival of the Web and the expansion of the Internet, which was as disruptive as Gutenberg's Printing Press 600 years ago but with a much larger impact and scale. Indeed, as more and more social networks start appearing and the Web 2.0 takes flight, the volume and speed of data creation were increased with a velocity that had never been experienced before, data are literally everywhere. Starting from the early 2000s we called these very large datasets: Big data [19].

Data by themselves are useless, to be useful, it must be processed [20]. While the developed infrastructures managed to store and retrieve massive data, traditional business intelligence tools started to show their limitations in terms of processing data with high velocity, volumes, and varieties. Thus, sophisticated algorithms with advanced capabilities of extracting knowledge from a large amount of data were needed. Artificial Intelligence offered a very interesting solution at this regard: Machine learning algorithms.

Right from the beginning, the ultimate goal of machines was to be as intelligent as humans. Back to the 1950s, Alan Turing's infamous question "Can machines think?" [21] was a landmark that set the foundations of AI. That was followed by heavy investments in AI research and projects. Experts predicted that it will take only a few years until we reach strong AI of an average human being. Yet, unfortunately, these predictions turned out to be wildly off the mark. In the 1970s, after the Lighthill report [22] stated that AI has failed to live up to its promises and that it was too brittle to be used in practice, AI research took the back seat, and remained an area of relative scientific obscurity and limited practical interest for over half a century. This period is known as AI winter. During this time, ML research, as a subfield of AI, struggled too with slow and modest realizations. It mainly focused on Artificial Neural Networks (ANN) which despite their potential theory, their practical feasibility was very limited due to the lack of available data and computational power. Indeed, in contrast to traditional algorithms, ML algorithms are not purpose-built programs for specific data. Instead, they are fed by observation data that enable them to gradually learn how to solve problems by induction. So, they need large data to make accurate predictions. In other words, they were waiting for the big data era to flourish again. This indeed was the case, with the rise of Big Data and improvements in computing power, ANN made a comeback in the form of Deep Learning [23]. At the present time, with a quite mutualistic relationship, ML and big data are making unprecedented and stunning success stories in diverse domains and more are still ahead of us. Figure 1 illustrates the remarkable recent interest in the two fields using google trends.

In hindsight, it seems clear by now that the concepts of "algorithms" and "data" are deeply rooted in history, not just a short-lived phenomenon. Figure 2 traces the growing interest in the two concepts over time, starting from the 18th century using Google Books Ngram Viewer. It seems also clear that these two concepts share a tangled causal chain of events. Surprisingly, few historical studies exist on the particular interaction between Algorithms and Data. This aspect is often neglected by contemporary scientists and researchers, more concerned with advances in the modern age. Here, we attempted to connect the history of the two concepts, believing that the historical background can help us to put facts in context and to understand the source

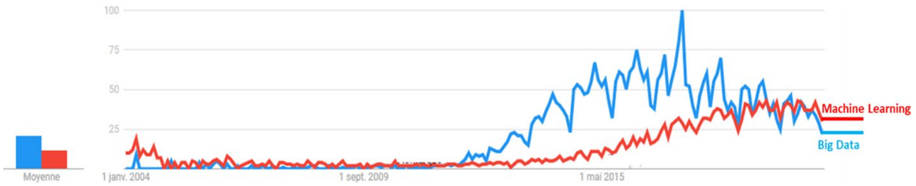

Fig. 1 "Big Data" and "Machine Learning" search trends, 2004–2020. Data source: Google Trends (https://www.google.com/trends)

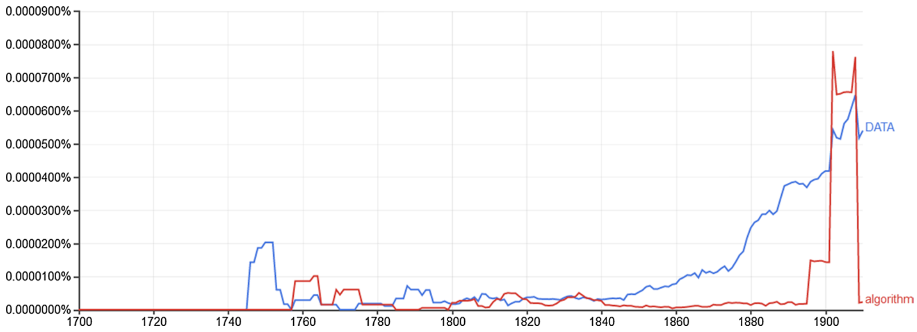

Fig. 2 Interest evolution of "Data" and "Algorithm" terms between 18th and 20th century. Data source: Google Books Ngram Viewer. (https://books.google.com/ngrams)

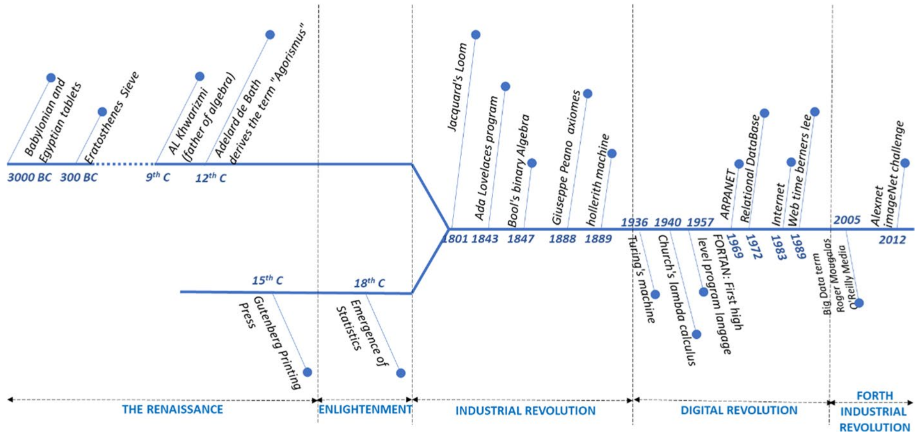

Fig. 3 "Data" and "Algorithms" evolution with key milestones.

of the voracious appetite of contemporary algorithmic practice for massive data. As summarized in Fig. 3, during the prehistoric space, foundations, and theories around the two concepts have been established. Starting from IR period, fields related to the two concepts started to converge with the emergence of machines. In the course of events, both data and algorithm concepts have evolved –Data in nature and size, and algorithms in complexity and intelligence. What brings them together is a common goal –Reproducing human intelligence.

Biological perspective

From a biological standpoint, data and algorithms have been around since the beginning of time and existed well before a special word has been coined to describe them. After all, data is a physical concept, as Landauer quoted in his paper [24] "Information is not a disembodied abstract entity; it is always tied to a physical representation. It is represented by engraving on a stone tablet, a spin, a charge, a hole in a punched card, a mark on paper, or some other equivalent". Following this line of thinking, many theorists support the idea of data being the essential unit of the physical universe [25–27]. This was famously encapsulated by physicist John Wheeler in his pithy slogan "It from Bit" [27], meaning that every aspect of a particle can be expressed as data, and put into binary code, which makes in Wheeler view "everything is data (information)" [27]. Indeed, many examples in nature depict the world as an entity capable of encoding data. The DNA molecule encodes biological data about all known organisms. The retina encodes visual data seeing through the eyes. And fingerprints encode biometric data that uniquely identify a natural person.

The same research line claimed the so-called "The computational theory of the universe". Lloyd [28] argued that "the computational paradigm for the universe supplements the ordinary mechanistic paradigm: the universe is not just a machine; it is a machine that processes information. The universe computes". Referring back to Wheel's 'it from bit' view [18], every process in the universe can be reduced to interactions between particles that produce binary answers: yes or no. That means nature, at its most fundamental level, is simply the flipping of binary digits or bits, just like algorithms do. The idea of the universe being a computer might seem to be only a metaphor. Metaphors usually reflect the most advanced thinking of the era that spawned it, and computers are the defining machines of our era, it seems thus natural to draw a parallelism between the Universe and Computer. Lloyd [28], however, argued in the defense of the theory that the computing universe stem from mathematics and physics facts: Maxwell, Boltzmann, and Gibbs showed that all atoms register and process information long before computers arrive. Aristotle has also discussed the physics of the computing universe and its implications thousands of years ago in his "beyond the physical" book [29]. Assuming that the universe is a computational entity, processing and interpretation of bits (data), give naturally rise to all sorts of complex order and structure seen in nature, which make the laws of physics essentially algorithms that calculate and handle data [24]. How planets move in the solar system is an algorithm, how a spider spins its webs is an algorithm, and how a baby recognizes his mother's face is also an algorithm.

In light of this, we can conclude that data, algorithms, and hence their interaction are shaping every biological organism and physical phenomena in the world. They are concepts giving by nature and not created by humans. In fact, what we are attempting to do is use these biologically inspired paradigms to create ever more intelligent technology. Nature teaches us, all tools and machines invented throughout human history are simply reverse-engineering the data processes that underlie biology, including that of our brain. Indeed, the most powerful information-processing system known has inspired many researches that try to mimic its functioning [30–33]. The most obvious example is ANN which as the name implies, try to learn tasks (to solve problems) by mimicking the networks of biological neurons. Other nature and biological organisms have also been a source of inspiration for many algorithms [34–36]. Taking animals as an example, a large variety of bio-inspired algorithms that simulate biological processes or mimics a collective behavior of animals has been reported in the literature [34]. Ant algorithms mimic the foraging behavior of social ants, Bees-inspired algorithms are inspired by the foraging behavior of honey bees in nature. Bat algorithm is inspired by the echolocation behavior of microbats. And genetic algorithm is inspired by biological evolution, to name a few. In spite of the popularity and success of nature- and bio-inspired computation, researchers in the field warn against the growing gap between the original biological models and the man-made models [34, 37]. Indeed, as reported by Molina et al. [34] a poor relationship is often found between the natural inspiration of an algorithm and its behavior. We are interested in one particular behavior "data consumption". By taking the previous example of ANN, this learning algorithm is known for being exceptionally data-hungry, it needs many examples and experiences to learn. However, it is not the case for its biological counterpart. For instance, a self-driving car powered by a deep learning algorithm may need to crash into a tree thousands of times in virtual simulations before it figures out how not to do that. While a person can learn to drive a car in only 15 h of training without crashing into anything. Visibly, nature is much less demanding in terms of data to learn. Therefore, the question here is why (learning) algorithms do not inherit the capacity of learning from a few experiences like their biological inspiration?

Technical perspective

In the introduction of his book "Machine Learning" [38], Tom Mitchell provides a short yet useful formalism for learning algorithms: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".*

This formalism is broad enough to include most tasks that we would conventionally call "learning" tasks. It puts emphasis on three core features that define a learning problem: (i) the learner's output (**T**), (ii) measures of success (**P**), and (iii) the training data (**E**). Although considered vital in the learning process, Mitchell's definition of experience or training data is not enough concrete to explore the requirement aspect in a precise formal setting. Fortunately, learnability has been extensively theorized in a dedicated field called Computational Learning Theory (CLT) [39]. In a broader sense, CLT formally studies fundamental principles of learning as a computational process, it seeks to formalize at a precise mathematical level the learning efficiency in terms of both data usage and processing time. CLT proposes the Probably Approximately Correct (PAC) model of learning [40] to formally define the efficiency of learning algorithms (referred to as learners). More precisely, the PAC allows analysing whether and under what conditions a learner L will *probably* output an *approximately* correct classifier.

Formally, given a [40]:

- Input data $\boldsymbol{X}$.
- Output values $\boldsymbol{Y} = \{-1, +1\}$.
- Training data $\boldsymbol{Data} = \{\ \langle \mathbf{x}_i, \mathbf{c}(\mathbf{x}_i) = \mathbf{y}_i\ \rangle\ , \mathbf{x}_i \in \boldsymbol{X}, \mathbf{y}_i \in \boldsymbol{Y}\}_{i=1}^{m}$
- $\boldsymbol{C}$ set of training concepts $\mathbf{c} \in \boldsymbol{C}$: $\mathbf{c}$: $\boldsymbol{X} \rightarrow \{0, 1\}$.

- Instances are generated at random from ***X*** according to some probability distribution **D**. In general, **D** may be any distribution and it will be unknown to the learner. **D** must be stationary, i.e. it does not change over time.
- A set ***H*** of possible hypotheses.
- A learner ***L*** outputs some hypothesis ***h*** from ***H*** as a model of ***c***.

Definition 1 True error $\boldsymbol{error_D\ (h)}$ of the hypothesis ***h*** with respect to the target function ***c*** and the probabilistic distribution **D** is the probability that the hypothesis ***h*** wrongly classifies a randomly selected instance according to **D** $\boldsymbol{(error_D(h) \equiv Pr_{x \in \mathrm{D}}\ [c(x) \neq h(x)])}$

Definition 2 Consider a concept class ***C*** defined over a set of instances ***X*** of length ***n*** (***n*** is the size of instances, i.e. the size of their representation) and a learner ***L*** using hypothesis space ***H***. ***C*** is PAC-learnable by ***L*** using ***H*** if for all $\boldsymbol{c \in C}$, distribution **D** over ***X***, **ε** such that $\mathbf{0 < \boldsymbol{\varepsilon} <}\frac{1}{2}$, **δ** such that $\mathbf{0 < \boldsymbol{\delta} <}$, learner ***L*** will with probability at least $\mathbf{1 - \boldsymbol{\delta}}$ (confidence) output a hypothesis $\boldsymbol{h \in H}$ such that $\boldsymbol{error_D\ (h)} \leq \varepsilon$, in time that is polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}$, ***n***.

The definition of PAC learnability contains two approximation parameters. The accuracy parameter **ε** determines how far the output classifier can be from the optimal one (this corresponds to the "approximately correct" part of "PAC"), and a confidence parameter δ indicating how likely the classifier is to meet that accuracy requirement (corresponds to the "probably" part of "PAC"). In short, the goal of a PAC-learner is to build a hypothesis with high probability (1- **δ**) that is approximately correct (error rate less than **ε**). Knowing that a target concept C is PAC-learnable allows us to bound the sample size necessary to probably learn an approximately correct classifier. Indeed, one of the fundamental questions in CLT is sample complexity, that is how much training data is required to achieve arbitrary small error with high probability. Valiant [40] proposed the following theorem (Formula 1), for use with finite concept classes, which gives an upper bound on required data as a function of the accuracy (ε) and confidence (δ) parameters:

$$m \geq \frac{1}{\epsilon}\left(ln|H| + ln\frac{1}{\delta}\right) \tag{1}$$

m is the amount of data needed to assure that any consistent hypothesis will be probably (with probability (1-δ)) approximately (within error ε) correct. We note that *m* grows linearly in 1/ε and logarithmically in 1/δ and *H.* Which means as ε gets smaller (i.e., as we want a more accurate hypothesis), we need more and more data. As there are more hypotheses in our hypothesis space, we also need to see more data. Likewise, as the probability of an approximately correct learner grows. More plainly, as we consider more possible classifiers, or desire a lower error or higher probability of correctness, we absolutely need more data. However, it is worth noting that there's only a logarithmic dependency on 1/δ, which means we can learn within an exponentially small probability of error using only a polynomial number of training data. There's also a log dependence on the number of hypothesis H, which means that even if there's an exponential number of hypotheses in our hypothesis space, we can still do the learning with a polynomial amount of data.

The theoretical role and influence of data in the learning process emphasized before, have been supported by a large number of empirical studies [41–43] which based on practical observations and experiences have confirmed the premise of "using more training data is necessary to improve performance". Figure 4 is an example of a learning curve of some of these works.

As noted by Gibbons et al. [41] (Fig. 4c), although for most learning algorithms, performance grows as data expands, not all algorithms are equally hungry. Indeed, there are algorithms that are more hunger than others are, and there is often a direct correlation between complexity and hungriness; the most complexe algorithms usually are more demanding in terms of data (e.g. deep learning), and the less complexe algorithms usually do not require massive data to learn (e.g. decision tree) [44]. Another observation to point out is that there is a saturation point which, as depicted in Fig. 4a), marks certain stability in performance and beyond which more data does not improve the overall model.

Business perspective

If ML algorithms were some products we buy in the supermarket, packing would surely be labeled: "Just add data and mix". From a business perspective, rather than being inherent, this dependency seems to be naturally derived from the data-driven strategy empowering the modern economy. The declining costs of computing elements and the ever-rising amount of accessible data has brought us past a point of inflection that marks the transition to a new economic era, that of the data-driven economy (DDE) [45]. In DDE, data is considered a strategic asset, and the success or failure of a firm now depends on how much data it has. An ever-expanding body of

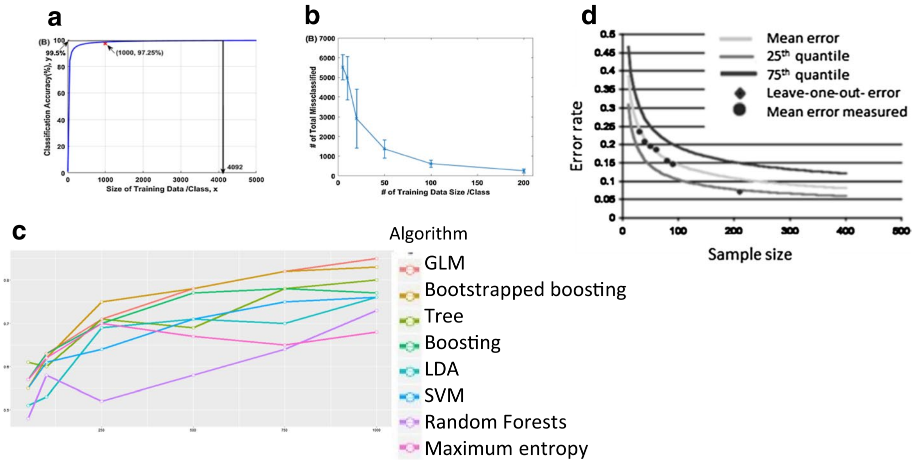

Fig. 4 Examples of learning curves. **a** Learning curve of medical image classification using differing sample size [42], **b** The number of misclassified medical images on the whole body in an increasing number of training data sets [42], **c** Algorithm performance with increasing training sample sizes and differing algorithms [41], **d** Learning curve in the tumor versus nontumor classification constructed using differing sample size [43]

evidence points to the crucial role of data in economic growth. According to Research and Markets report [46], the global big data market size will grow from USD 138.9 billion in 2020 to USD 229.4 billion by 2025, at a compound annual growth rate of 10.6 % during the forecast period. 98.8 % of Fortune 1000 represented in the last version of NewVantage Partners executive survey [47] confirm an active investment in Big Data underway. 64.8 % of these leading firms are investing greater than USD 50MM in Big Data and AI initiatives. The business benefits of data-driven strategies can be grouped into three broad axes:

- *Innovating better*: Data is regarded as a relevant determinant for the likelihood of a firm becoming a product innovator as well as for the market success of product innovations [48]. Insights obtained from data can be used to generate new innovative products, services, and processes leading to improved firms' innovation performance and thus the firm performance overall.
- *Understanding customer better*: Data is revolutionizing the matching function between firms and customers. Data is generated primarily by customers and can be used to getting close to customers, understand their behavior, and reflect their value co-creating actions. This benefits firms with regard to precision marketing, new product development, and realigning business strategy to maintain sustainable competitive advantage [49].
- *Managing better*: Another business benefit from data is realized through significant changes in management practices. More precisely speaking, the emergence of data driven decision making [50]. Data have led many managers to change how they make decisions, relying less on intuition and more on data. As Jim Barksdale, the former CEO of Netscape quipped, "If we have data, let's look at data. If all we have are opinions, let's go with mine" [50]. According to scholars [50, 51] embracing the data-driven decision-making practice creates opportunities to make better decisions that lead to commercial growth, evolution, and an increased bottom line.

All these cases of data use generate more business value, but also more data-hungry apps and practices. Digital native firms that, naturally, due to their business model, utilize data (e.g. Google and Amazon) are the most innovators in the area. However, not all firms or businesses can afford to generate or collect massive data, for such businesses it should be some viable alternatives to compete in the AI race.

In the end, the multi-perspective study has genuinely served to clearly articulate the issue. Each perspective has brought its own valuable insights to understand the complementary and closely intertwined nature of the "data/algorithms" relationship. The main conclusion that we can draw so far is that data and algorithms have shared a long history together, and a tight relationship exists between the two concepts in nature, however, in stark contrast to their artificial counterparts, the biological algorithms do not require massive data to learn. Formally, the theory of learning shows that the amount of data we need for a learning algorithm depends on the targeted performance. If we want more performance, we need more data. Last, seeking more business value from data has resulted in a data-driven economy that lacks alternative algorithms that can learn even if only small data is available.

Data hungry algorithms: an inconvenient truth

Against the presented backdrop, resolving the hungriness issue might appear evident. After all, we live in the data era. If algorithms need more data, we should feed them with more data. While it might be value in this approach, whether it is the right or the wrong one, is debatable. There are largely two main reasons that make such approach problematic: data scarcity and AI sustainable progress, which reflects respectively industrial need for cost-effective learning and academic ambition of Artificial General Intelligence (AGI).

The curse of scarcity

While it would be natural to think that almost every single business or market is snowed under with an avalanche of data, it is far from the truth. Data are available for only a subset of companies, in many cases data are considered a scarce resource. In fact, in a real-world setting, data are hard to acquire and if it openly exists, it often has questionable quality. A recent survey by Dimensional Research shows that 96% of enterprises encounter data quality and labeling challenges in ML projects [54]. Obtaining voluminous and accurately labeled data is challenging for many reasons: First, (i) the high cost of data collection or annotation, learning algorithms do not only crave massive samples, but the data have also to be manually annotated beforehand, involving tasks as complex as making human-like judgments about images or videos which implies a significant cost, time and effort. Crowdsourcing can be utilized to harness the crowd to directly annotate data and thus reducing human labor cost. The result, however, inevitably contain a large amount of low-quality annotations [55]. Second, (ii) limited domain expertise, to label some general images with trivial categories such as "cats" and "dogs" we need to understand the difference between these two animals, which might seem very "common sense", but to label medical images as "cancer" or "not cancer", we need deep medical expertise, and it is often hard to find such domain experts who can credibly identify and label a specific type of data such as tumors or chemical formulas [56]. Finally, (iii) access limitation is also a pain point for acquiring more data. Especially for domain involving sensitive data, the amount of data can be limited due to privacy, safety or ethical issues. For example, the collection and labeling of DICOM medical image scans is challenging for privacy reasons [47]. Recently, compliance and regulatory issues have become pressing concerns for enterprises dealing with data, especially after GDPR entered into force [58]. Even companies who used to have access to a large amount of data might face increasing difficulties.

Moreover, besides being laborious and expensive, in some cases having more data is simply impossible. In some domains and for rare events sufficient data might not be available, which may hinder ML adoption in such applications. The most striking scenarios include: (i) Studying rare phenomena such as earthquakes, epidemics and floods. (ii) Aggregate modeling of states, countries, rare animal's race, or any situation where the population itself is limited. And (iii) time series forecasting which often lacks historical or seasonality data for a target variable. These scenarios can be found in many domains such as production, marketing, government, military, and education. Perhaps the most obvious domains that are the most cursed by the scarcity of data are: (a) Robotics, robots are expected to act like humans or animals, they often have to operate in ever-changing, uncontrolled real-world environments, and will inevitably encounter instances of classes, scenarios, textures, or environmental conditions that were not covered by the training data [59]. This is why operating reliably in irregular scenarios such as strange weather occurrences or other vehicles' unpredictable driving patterns still problematic for autonomous vehicles. (b) Medicine is also deeply cursed; medical data require a ground truth normally provided by an expert physician ending up with only a small set of annotated data, microarray and RNA-Seq data are typical of this type of small sample problem [310, 311]. In addition, new diseases consistently occur with little historical data, and rare diseases frequently occur with few cases, which can only obtain scarce training data with accurate labels [56]. This is why it is still challenging for Computer-Aided-Diagnosis to detect rare tumors such as bladder cancer, for which there are only a few medical records [60].

AI's next frontier

Data superpowers to increase algorithm's performance has sparked the so-called "The Unreasonable Effectiveness of Data" mindset [61], which advances that even very complex problems may be solved by simple statistical models trained on massive datasets. Google's director of research, Peter Norvig, puts it this way: "We don't have better algorithms. We just have more data" [52]. This leads us to consider a fundamental question about the ML field future: will continually increase amounts of training data be sufficient to drive continued progress in ML absent the development of more advanced and sophisticated algorithms?

While many primary works answered affirmatively to the question [61, 62], other recent studies are attempting to prove the opposite. As stated by Domingos [53] "Data alone is not enough, no matter how much of it you have". Basing his statement on "no free lunch" theorem, Domingos argued that learners need to be improved in a way to embody knowledge beyond the data it is given in order to generalize beyond it. Zhu et al. [63] highlighted by the evidence a surprising observation that is at some point, off-the-shelf implementations often decrease in performance with additional data. At such point, improving the algorithm is required to uphold performance. Greco et al. [64] provided a qualified defense of the value of less data, they claimed that seen through the lens of cognitively inspired AI, the future of the field is about less data, not more.

Furthermore, the move toward data-efficient AI is also a necessity in order to make learners more human-like. In fact, in spite of their biological inspiration and performance achievements, in their current implementation learners differ from human intelligence in crucial ways [66]. As previously discussed, humans or even animals can quickly learn new skills or adapt to changing circumstances based on a few experiences. A child, for instance, can learn to recognize a new kind of object or animal from just a short exposure to a single example. Getting burned once will teach him to be careful with fire, he may recognize a face that he has seen only briefly, and he can recognize a lullaby that he has heard only a few times before [67]. Similarly, soon after birth and without the benefit of massive data sets, animal babies start to figure out how to solve problems of feeding, fighting, fleeing, and mating. A squirrel can jump from tree to tree within months of birth, a colt can walk within hours, and spiders are born ready to hunt [68]. The main reasons why young animals (including humans) learn faster, better, and with less data, is that they rely heavily on innate mechanisms [68] and make use of prior knowledge. Such innate learning process is yet to be reproduced in artificial learners, hence the need to rekindle the old "nature versus nurture" debate [69] in AI context, if the goal is to achieve AGI, a human-level machine intelligence that is capable of learning the way we do.

A final and a more general motivation for developing data-efficient algorithms stems from the aim of achieving a more robust AI. As described by Marcus [70], the next level of AI is not necessarily superhuman, but has to solve problems in a systematic and reliable way, it should not be a "pointillistic" intelligence that works in many cases but fails in many others, but rather it must implement a "solutionism intelligence" that solves any problem encountered, under all conditions. Thus, it is our contention that to reach the next level of AI, algorithms should work for both data and non-data-driven settings. Hence, a research agenda for robust AI should include solutions to improve learners' performance for problems with a small dataset.

As a result, at the frontiers of AI, efforts should be underway to develop improved forms of ML that are more robust and more human-like. Enhanced algorithms that will allow autonomous vehicles to drive us around both usual and unpredictable places, and that will work as well for rare diseases as for the common ones without being excessively demanding in their requirements for data. This new generation of learners will reshape our understanding of AI and disrupt the business landscape in profound ways.

Review

This part reports on the findings of our extensive literature review conducted by examining relevant works dealing with learners' data-efficiency issue. Different perceptions to approach the problem lead to different ways to solve it. Based on the study of the related body of research, we distilled four main strategies to alleviate algorithms data hungriness. Each one is spanning its own spectrum and together they shape the advanced in this research landscape. Figure 5 categorizes existing techniques into a unified taxonomy and organizes them under the umbrella of each strategy. We devote a section to each strategy. First, we point out research exploring learning algorithms that go beyond the realm of supervised learning. Second, we review relevant techniques to enlarge artificially the training dataset. Third, we overview the different forms that learning from previous experiences can take. Finally, we introduce a new research direction that aims to conceive innovative hybrid systems that combine both high-prediction, explainability, and data-efficiency.

Non-supervised learning paradigms

When talking about data hungriness in ML, we are mostly referring to supervised learning algorithms, it is this type of learning that had the most voracious appetite for data. Supervised methods need labelled data to build classification and regression models and the performance of these models relies heavily on the size of labelled training data available. One straightforward strategy to alleviate this data-dependency would be then to

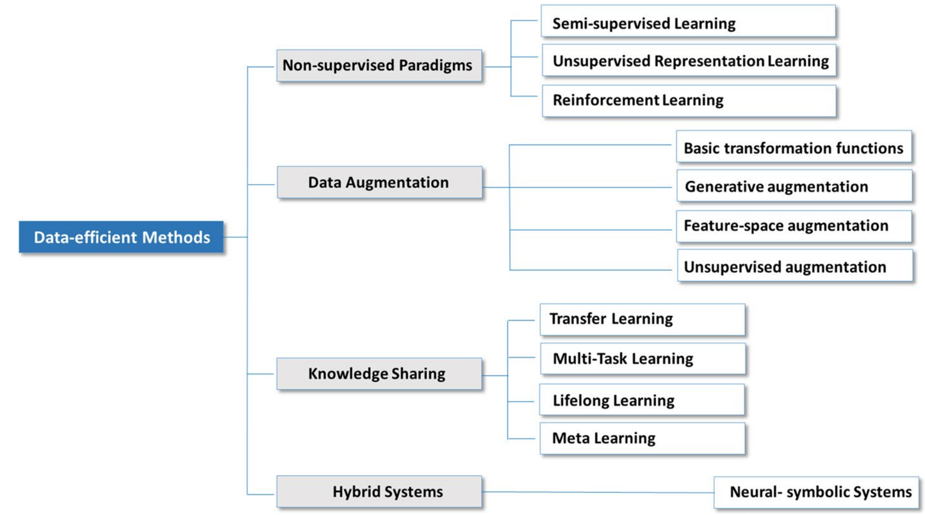

Fig. 5 Taxonomy of data-efficient methods, while many methods lie in the intersections of multiple categories they are grouped by where they are reviewed in this survey

use other learning paradigms. Paradigms that either do not require pre-existing data and could generate ones by interacting with their environment (i.e. reinforcement learning), or paradigms that need only small set of labelled data (i.e. semi-supervised learning), or paradigms that use for learning raw unlabelled data (i.e. unsupervised learning). In this section, we scan recent methods in the literature that involve these non-supervised learning paradigms.

Semi-supervised learning methods

The wide availability of unlabeled data in several real-world scenarios, and, at the same time, the lack of labeled data has naturally resulted in the development of semi-supervised learning (SSL) [71]. SSL is an extension of supervised learning that uses unlabeled data in conjunction with labeled data for better learning. SSL can also be viewed as unsupervised learning with some additional labeled data. Accordingly, SSL may refer to either semi-supervised classification [71] where unlabeled data are used for regularization purposes under particular distributional assumptions to enhance supervised classification. Or semi-supervised clustering [72], where labeled data are used to define some constraints to obtain better-defined clusters than the ones obtained from unlabeled data. In the literature, most attention has been paid to the methods of these two groups. Relatively less studies deal with other supervised/unsupervised problems such as semi-supervised regression [73] and semi-supervised dimensionality reduction [74]. Depending on the nature of the training function, SSL methods are commonly divided, in the literature, into two settings: inductive and transductive. Given a training dataset, inductive SSL attempts to predict the labels on unseen future data, while transductive SSL attempts to predict the labels on unlabeled instances taken from the training set [71]. Abroad variety of SSL methods have been proposed in the two settings. These methods differ in how they make use of unlabeled data, and in the way they relate to supervised algorithms. Next, we review the most three dominant families of methods namely: (i) self-labeled methods, (ii) graph-based methods, and (iii) extended supervised methods.

(i) Self-labeled methods

These techniques are used to solve classification tasks, they aim to obtain enlarged labeled data by assigning labels to unlabeled data using their own predictions [75]. As general pattern, one or more supervised base learners are iteratively trained with the original labeled data as well as previously unlabeled data that is augmented with predictions from earlier iterations of the learners. The latter is commonly referred to as pseudo-labeled data. The main advantage of this iterative SSL approach is that it can be "wrapped" around any supervised learner.

The basic iterative process schema for self-labeled techniques is self-training [75], it consists of a single supervised classifier that is iteratively trained on both labeled data and data that has been pseudo-labeled in previous iterations of the algorithm. Tanha et al. [76] discussed the choice of the base learner, they stated that the most important aspect of the learner is to correctly estimate the confidence of the predictions so as to be successful. They experimentally showed that ensemble learner as a base learner gives an extra improvement over the basic decision tree learners. Livieris et al. [77] proposed an algorithm that dynamically selects the most promising base learner from a pool of classifiers based on the number of the most confident predictions of unlabeled data. Li and Zhou [78] addressed the issue of erroneous initial predictions that can lead to the generation of incorrectly labeled data, they presented the SETRED method which incorporates data editing in the self-training framework in order to actively learn from the self-labeled examples.

Co-training is a variant of self-training schema that uses multiple supervised classifiers [79]. Considered as a special case of the multiview learning [80], cotraining schema assumes that, by dividing the feature space into two separate categories, it is more effective to predict the unlabeled data each time. In the work of Didaci et al. [81], the relation between the performance of cotraining and the size of the labeled training set was examined, and their results showed that high performance was achieved even in cases where the algorithm was provided with very few instances per class. Jiang et al. [82] introduced a hybrid method which combines the predictions of two different types of a generative classifier (Naive Bayes) and a discriminative classifier (Support Vector Machine) to take advantage of both methods. The final prediction is controlled by a parameter that controls the weights between the two classifiers. Their Experimental results showed that their method performs much when the amount of labeled data is small. Qiao et al. [83] proposed a deep cotraining method that trains multiple deep neural networks (DNN) to be the different views and exploits adversarial examples to encourage view difference, in order to prevent the networks from collapsing into each other. As a result, the co-trained networks provide different and complementary information about the data.

(ii) Graph-based methods

Transductive methods typically define a graph over all data points, both labeled and unlabeled, the nodes of the graph are specified by unlabeled and labeled samples, whereas the edges specify the similarities among the labeled as well as unlabeled samples

[84]. The common graph-based SSL methods are based on a two-stage process that are: (i) constructing a graph from the samples and then (ii) propagating the partial labels to infer those unknown labels via the graph [71]. Initial research on graph-based methods was focused on the inference phase. Pang and Lee [85] approached the inference from a min-cut perspective. They used the min-cut approach for classification in the context of sentiment analysis. Other works approached graph-based inference phase from the perspective of Markov random fields [86] and Gaussian random fields [87]. On the other hand, the process of construction of the graph basically involves two stages: the initial phase involves graph adjacency matrix construction, and the second phase deals with graph weight calculation. Blum and Chawla [88] experimented graph construction using k-nearest neighbor and ε nearest neighbor. The approach simply connects each node to all nodes to which the distance is at most ε. The most used functions for calculation of graph weights are: the Gaussian similarity function and the inverse Euclidean distance [71]. We note that although graph-based methods are typically transductive, inductive graph-based methods do also exist in the literature, this line of work encompasses approaches that utilize the intrinsic relationship from both labeled and unlabeled samples to construct the graph to estimate a function [89]. However, it is generally acknowledged that transductive graphs usually perform better than inductive ones [84]. Another line of work, that has received recently much attention, is the scalable graph with SSL. A commonly used approach to cope with this issue is called anchor graph regularization [90]. This model builds a regularization framework by exploring the underlying structure of the whole dataset with both datapoints and anchors. Liu et al. [90] provided a complete overview of approaches for making graph-based methods more scalable.

(iii) Extended supervised methods

These methods are direct extensions of traditional supervised learning methods to the semi-supervised setting. The most prominent examples of this class of methods are: (i) semi-supervised support vector machine and (ii) semi-supervised neural networks.

Mainstream models of semi-supervised SVM include many variants such as S3VM [91], TSVM [92], LapSVM [93], meanSVM [94], and S3VM based on cluster kernel [95]. The related literature presents S3VM and TSVM as the two most popular variants. The optimal goal of S3VM is to build a classifier by using labeled data and unlabeled data. Similar to the idea of the standard SVM, S3VM requires the maximum margin to separate the labeled data and unlabeled data, and the new optimal classification boundary must satisfy that the classification on original unlabeled data has the smallest generalization error. TSVM exploits specific iterative algorithms which gradually search a reliable separating hyperplane (in the kernel space) with a transductive process that incorporates both labeled and unlabeled samples. Since their introduction, semi-supervised SVM models have evolved on different aspects and various approaches have proposed to improve existing variants or to create new ones [96].

Recently, numerous research efforts have been made to build an effective classification model using semi-supervised neural networks (SSNN) methods. The hierarchical nature of representations in DNN makes them a viable candidate for semi-supervised approaches. If deeper layers in the network express increasingly abstract representations of the input sample, one can argue that unlabeled data could be used to guide the network towards more informative abstract representations. A common strategy of this line of research is to train the DNN by simultaneously optimizing a standard supervised classification loss on labeled samples along with an additional unsupervised loss term imposed on either unlabeled data or both labeled and unlabeled data [97]. The typical structure for such strategy is Ladder Networks [98], an autoencoder structure with skip connections from the encoder to decode. proposed by Rasmus et al. [98], this model is trained to simultaneously minimize the sum of supervised and unsupervised cost functions by backpropagation, avoiding the need for layer-wise pre-training. Prémont-Schwarz et al. [99] extended the Ladder Network architecture to the recurrent setting by adding connections between the encoders and decoders of successive instances of the network. A related group of SSNN methods is known as teacher- student models [71] where a single or an ensemble of teacher models are trained to predict on unlabeled data and the predicted labels are used to supervise the training of a student model. Thus, the teacher guides the student to approximate its performance under perturbations in the form of noises applied to the input and hidden layers of models. The teacher in the Teacher-Student structure can be summarized as being generated by an exponential moving average (EMA) of the student model. Various ways of applying the EMA lead to a variety of methods of this category. In the VAT Model [100] and the Π Model [101], the teacher shares the same weights as the student, which is equivalent to setting the averaging coefficient to zero. The Temporal Model [101] is similar to Π Model except that it also applies an EMA to accumulate the historical predictions. The Mean Teacher [102] applies an EMA to the student to obtain an ensemble teacher. There are other types of SSNN methods that are based on generative models [130], the primary goal of these methods is to model the process that generated new data, this technique will be reviewed in the "Data Augmentation" section.

Unsupervised representation Learning methods

The limited performance of data-hungry models when only a limited amount of labeled data is available for training has led to an increasing interest in literature to learn feature representations in an unsupervised fashion to solve learning tasks with insufficient labeled data. Unsupervised representation learning [300] encompasses a group of methods that make use of unlabeled data to learn a representation function *f* such that replacing data point *x* by feature vector *f(x)* in new classification tasks reduces the requirement for labeled data. Such learners seek to learn representations that are sufficiently generalizable to adapt to various learning tasks in future. In this case, the representations learned from unsupervised methods are usually assessed based on the performances of downstream classification tasks on top of these representations. Thus, the focus here is not on clustering or dimensionality reduction, but rather on learning unsupervised representations. Accordingly, we review in this subsection the recent progress and the most representative efforts on unsupervised representation learning methods. Generally, three groups of research fall under the umbrella of methods for training unsupervised representations, namely: (i) Transformation-Equivariant Representations, (ii) Self-supervised methods, and (iii) Generative Models.

(i) Transformation-equivariant representations

The learning of Transformation-Equivariant Representations (TERs), was introduced by Hinton et al. [103] as the key idea of training capsule nets and has played a critical role in the success of Conventionnel Neural Networks (CNNs). It has been formalized afterward in various ways. Basically, TER learning seeks to model representations that equivary to various transformations on images by encoding their intrinsic visual structures. Then the successive problems for recognizing unseen visual concepts can be performed on top of the trained TER in an unsupervised fashion. Along this line of research, Group-Equivariant Convolutions (GEC) [104] have been proposed by directly training feature maps as a function of different transformation groups. The resultant feature maps are proved to equivary exactly with designated transformations. However, GEC have a restricted form of feature maps as a function of the considered transformation group, which limits the flexibility of its representation in many applications. Recently, Zhang et al. [105] proposed Auto-Encoding Transformations (AET), this form of TER guarantees more flexibility to enforcing transformation equivariance by maximizing the dependency between the resultant representations and the chosen transformations. Qi et al. [106] proposed later an alternative Auto-encoding Variational Transformation (AVT) model that reveals the connection between the transformations and representations by maximizing their mutual information.

(ii) Self-supervised methods

Self-supervision is a form of unsupervised learning where the data provides the supervision. Broadly speaking, self-supervised learning converts an unsupervised learning problem into a supervised one by creating surrogate labels from the unlabeled dataset, potentially greatly reducing the number of labeled examples required [107]. Currently, there are several techniques to achieve that, including Autoregressive models, such as PixelRNN [108], PixelCNN [109], and Transformer [110]. These methods are trained by predicting the context, missing, or future data, they can generate useful unsupervised representations since the contexts from which the unseen parts of data are predicted often depend on the same shared latent representations. Generative models can also be considered as self-supervised, but with different goals: Generative models focus on creating diverse and realistic data, while self-supervised representation learning care about producing good features generally helpful for many tasks.

(iii) Generative models

As for SSL, Auto-Encoders [141], Generative Adversarial Nets (GAN) [130] and many other generative models have been widely studied in unsupervised learning problems, from which compact representations can be learned to characterize the generative process for unlabeled data. By using an unsupervised fashion such models aim essentially at generating more data, this is why, as mentioned before, generative models are reviewed under the “Data Augmentation” strategy.

Reinforcement learning

Another learning paradigm that has driven impressive advances in recent years without the need for gobs of real-world data is Reinforcement Learning (RL) [111].

RL is one step more data-efficient than supervised learning. In supervised learning, the learner learns from a labeled dataset with guidance. Whereas RL agent interacts with its environment, performs actions, and learns by a self-guided trial-and-error method [301]. In other words, in the absence of a training dataset, RL agent is bound to learn from its experience. Seen from this perspective, RL algorithms can be viewed as an optimized-data alternative to supervised learning algorithms, since the sample complexity does not depend on preexisting data, but rather on the actions the agent takes and the dynamics of the environment [302].

One of the remarkable achievements of such learning paradigm is AlphaGo Zero [1], as given absolutely no prior data other than the game's rules. With no other input, simply by playing against itself, AlphaGo Zero learned the game of Go better than any human or machine ever had. Another example is PILCO (Probabilistic Inference for Learning Control) [303], a model-based policy search method that propagates uncertainty through time for long-term planning and learns parameters of a feedback policy by means of gradient-based policy search. It achieved an unprecedented data efficiency for learning control policies from scratch (it requires only about 20 trials, experience of about 30 s), and is directly applicable to physical systems, e.g., robots.

Following the taxonomy of Arulkumaran et al. [112] two main RL approaches can be distinguished: (i) methods based on value functions which are based on estimating the value (expected return) of being in a given state. This approach forms the foundation of the state-action-reward-state-action (SARSA) algorithm [113], and Q-learning [114] the most commonly used RL algorithms. And (ii) methods based on policy search that do not need to maintain a value function model, but directly search for an optimal policy. There is also a hybrid, actor-critic approach, which employs both value functions and policy search. Between the two approaches, policy-based methods are known to be significantly more sample-efficient because they reuse data more effectively [304]. For instance, Guided Policy Search [305] is very data-efficient as it uses trajectory optimization to direct policy learning and avoid poor local optima.

From the model perspective, RL algorithms can be categorized as (i) model based and (ii) model free depending on whether the agent has the access or learns a model of the environment [112]. Having a model in hands allows the agent to plan ahead to predict state transitions and future rewards. Thus, If the model is correct, then the learning would be greatly benefited in terms of sample efficiency compared to model-free methods. Hence, model-based algorithms are taking the lead in terms of data efficiency as they try to derive a model of the environment and use that model for training the policy instead of data from real interactions (e.g., PILCO) [304].

Contemporary deep reinforcement learning (DRL) has led to tremendous advancements [115], but has also inherited shortcomings from the current generation of deep learning techniques that turned the paradigm of trial-and error-learning to a data-hungry model [116]. Indeed, the combination requires humongous experience before becoming useful, it is even claimed that DRL hunger for data is even greater than supervised learning. This is why although DRL can potentially produce very complex and rich models, sometimes simpler, more data-efficient methods are preferable [112].

In fact, DRL excels at solving tasks where large amounts of data can be collected through virtually unlimited interaction with the environment such as game settings. However, training DRL model with limited interaction environment such as production-scale, healthcare or recommender systems is challenging because of the expensiveness of interaction and limitation of budget at deployment. The recent wave of DRL research tried to address this issue, for instance, Botvinick et al. [117] suggested in its recent work two key DRL methods to mitigate the sample efficiency problem: episodic deep RL and meta-RL. Buckman et al. [306] proposed a stochastic ensemble value expansion (STEVE) to combine deep model-free and deep model-based approaches in RL in order to achieve the high performance of model-free algorithms with low sample complexity of model-based algorithms. To reduce the number of system interactions while simultaneously handling constraints, Kamthe et al. [307] introduced a model-based DRL framework based on probabilistic Model Predictive Control (MPC) with learned transition models using Gaussian processes. The proposed approach requires on average only six trials (18 s). Popov et al. [308] introduced two extensions to the Deep Deterministic Policy Gradient algorithm (DDPG) for data-efficient DRL. They showed that by decoupling the frequency of network updates from the environment interaction, data-efficiency has substantially been improved. In a recent work, Schwarzer et al. [309] proposed Schwarzer Self-Predictive Representations (SPR), a method that makes use of self-supervised techniques along with data augmentation to train DLR in limited interaction environment. The model achieves a median human-normalized score of 0.415 on Atari in a setting limited to 100 k steps of environment interaction, which represents, according to the authors, 55 % relative improvement over the previous state-of-the-art.

Discussion

Ultimately, unlabeled data are expected to be a game-changer for AI to move forward beyond supervised, data-hungry models. While introducing his most recent research « SimCLR » [118] a framework for contrastive learning of visual representations that has achieved a tremendous performance leap in image recognition using unsupervised learning, AI pioneers Geoff Hinton quoted recently in AAAI 2020 Conference that « unsupervised learning is the right thing to do ». Appearing on the same AAAI stage,

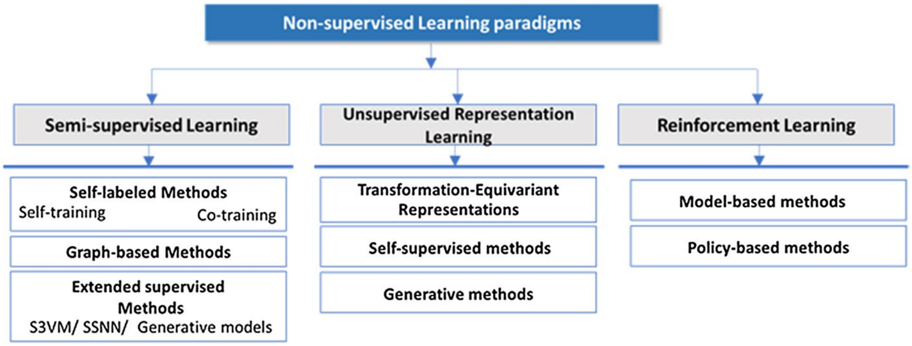

Fig. 6 Categorization of non-supervised learning methods

Table 1 Summary of non-supervised learning methods

Non-supervised Learning paradigms	Non-supervised Learning techniques	References
Semi-Supervised Learning	Self-labeled Methods	[76–78, 81–83]
	Graph-based Methods	[85, 88, 90]
	Extended supervised Methods	[90, 92, 93],94,95, 100,101,102]
Unsupervised Representation learning	Transformation-Equivariant Representations	[103–106]
	Self-supervised methods	[108– 110]
	Generative methods	[130, 141]
Reinforcement Learning	Model-based methods	[303, 306, 307]
	Policy-based methods	[305, 308]

Turing Award winner Yann LeCun agreed that unsupervised learning, semi-supervised learning, or any model training that does not require manual data labeling are vital tools for the progress of ML and its applications. The literature is flourishing with a broad variety of semi-supervised and unsupervised algorithms (Fig. 6; Table 1 summarizes the key discussed methods). As a matter of fact, recently, both lines of research have strongly focused on DNN, particularly deep generative models that have been extensively used for self-supervision and have also been extended to the semi-supervised setting. However, despite the success of these methods, a considerable amount of empirical studies reveals that exploiting unlabeled data might deteriorate learning performance [71]. The potential performance degradation caused by the introduction of unlabeled data is one of the most important issues to be resolved especially in SSL. Furthermore, we noted that the evaluation aspect has received relatively little attention in the literature. Pragmatic baselines to be used for empirically evaluating the performance of non-supervised learning methods in order to choose an approach that is well suited to a given situation are relatively rare. Recently, Oliver et al. [119] established a set of guidelines for the realistic evaluation of SSL algorithms. In turn, Palacio-Ninoe et al. [120] have proposed evaluation metrics for unsupervised learning algorithms. In recent works, there has been a notable shift towards automatic selection and configuration of learning algorithms for a given problem. However, while automating ML pipeline has been successfully applied to supervised learning [121], this technique is yet to be extended to the non-supervision settings.

Data augmentation

To fight the data scarcity problem and to increase generalization, the literature suggests the use of Data Augmentation (DA) techniques. DA entails a set of methods that apply mutation in the original training data and synthetically creating new samples [122]. It is routinely used in classification problems to reduce the “overfitting” caused by limited training data [123]. Indeed, when a model is trained with a small training set, the trained model tends to overly fit to the samples in the training set and results in poor generalization. DA acts as a regularizer to combat this. Considered more and more as a vital and ubiquitous instrumental data processing step in modern ML pipelines, DA has become a subject of big interest in both academic and industrial settings. Contributions in this field are actively growing; new DA techniques emerge in a regular basis. Being unable to cover all existing techniques, based on the studied literature, we rather propose a classification of existing augmentation strategies hinge on four aspects: (i) Whether the mutation/transformation is handcraft or smart (learning-based), accordingly we distinguish between basic and generative augmentations. (ii) Whether the augmentation is performed in the data or the feature space, accordingly we distinguish between data-space and feature-space augmentations. (iii) Whether the data to be augmented are acquired or come from another similar dataset, accordingly we distinguish between in situ augmentation and borrowed augmentations. (iv) Whether the data to be augmented are labeled or unlabeled, accordingly we distinguish between supervised and unsupervised augmentations. In the following, we briefly introduce the main methods and review works that made the biggest impact in each class of augmentation.

Basic vs generative augmentations

The most popular and basic augmentation schema is the traditional transformations, the aim of this class of methods is to preserve the label of the data through simple transformations which can happen in realistic data. For image augmentation, for example, this can be achieved by performing geometric transformations (such as random flipping, cropping, translation, rotation…), or by changing color, brightness, or contrast (Fig. 7). Intuitively, a human observer can still recognize the semantic information in the transformed image, while for the learner it is perceived as new data. The manipulations applied to ImageNet [123], remains the standard for this class of technique. The model has been used extensively for various purposes since its development. Vast amounts of research have used it to benchmark their models against or as a base model to test new transformations. On the other hand, the MNIST (handwritten digit) dataset [124] is commonly augmented using elastic distortions [125], another transformation technique that mimics the variations in pen stroke caused by uncontrollable hand muscle oscillations. Yaegeret al. [126] also used the same technique for balancing class frequencies, by producing augmentations for under-represented classes. Mixing paring samples [127] proposed by Inoue et al. is another basic augmentation technique for image classification task, which can create the new image from an original one by overlaying another image randomly picked from the training set. Zhong et al. [128] introduced random erasing as a means to make models more robust to occlusion, by randomly erasing rectangular regions of the input image. Generally, basic class of augmentations has been proven to be fast, reproducible, and reliable technique with an ease implementation [122, 129]. However, it relies on simple and basic transformation functions, in some specific cases, this could result in further overfitting. This has prompted further investigation for new more advanced and powerful DA techniques that include learning algorithms in the augmentation process.

Motivated by the recent advance of generative models especially adversarial learning, Generative Adversarial Networks (GAN) [130] have been increasingly used for generating synthetic data. In a nutshell, in GAN based augmentation, two networks are trained to compete with each other, the Generator and the Discriminator, the first creates new data instances (typically an image) while the second evaluates them for authenticity (real or fake), this co-optimized process results in generating realistic synthesized data (Fig. 7). The result obtained using generative models differs from the one obtained by

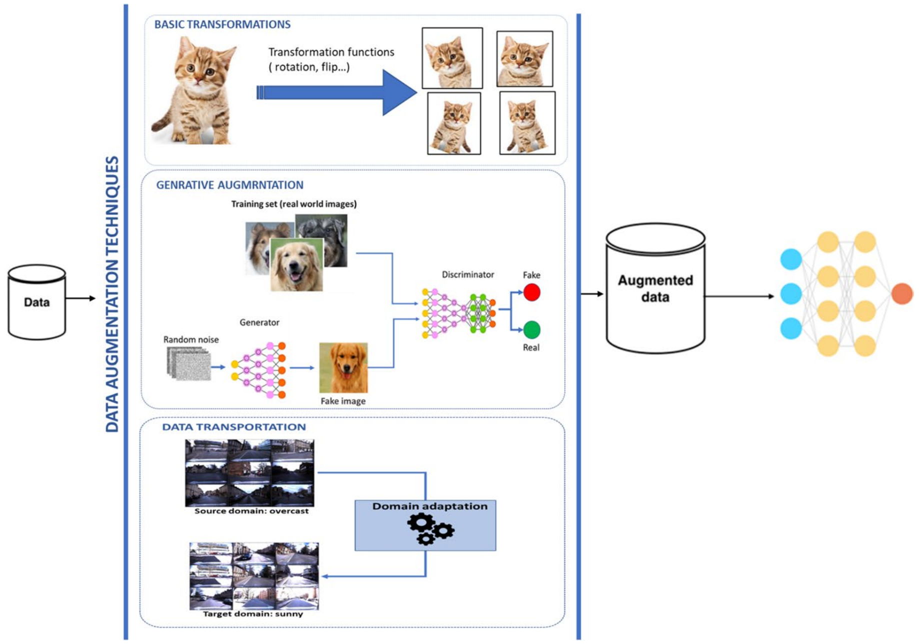

Fig. 7 Main augmentation techniques (images for domain adaptation are from [181])

basic transformations. The latter modifies real data with some sort of predefined transformation functions while the former creates new synthetic data. The synthetic data need to be different enough from the original ones so that these variations lead to a better generalization capacity. In contrast to basic augmentation techniques which are limited to minor changes on data to not damage the semantic content. This makes generative augmentation similar to imagination or dreaming, it has a creative effect that makes it known for its artistic applications, but this schema also serves as a great tool for DA.

Basic GAN architectures are unable to create high-quality new samples. This is why the main contributions in GAN based augmentation are new architectures that modify the standard GAN framework through different network architectures, loss functions, evolutionary methods, and others to produce higher quality additional data. One of these variants is conditional GAN introduced by Odena et al. [131] in 2016 to generate data by controlling the random noise generation. Many extensions of conditional GANs have been proposed afterward. ACGAN (Auxiliary classifier GAN) [131] changed the GAN energy function to add the discrimination class error of the generated sample and the real sample. This variant demonstrates that a complex latent coder could boost the generative sample's resolution. Antoniou et al. proposed DAGAN (Data Augmentation GAN) [132] that generates synthetic data using a lower-dimensional representation of a real image. The authors train a conditional GAN on unlabeled data to generate alternative versions of a given real image. Mariani et al. proposed BAGAN (balancing GAN) [133] as an augmentation tool to restore balance in imbalanced datasets. The use of non-conditional GANs to augment data directly has only very recently been explored. Karras et al. used PGGAN (Progressive Growing of GAN) [134] a stable architecture to training GAN models to generate large high-quality images that involves incrementally increasing the size of the model during training. This approach has proven effective at generating high-quality synthetic faces that are startlingly realistic. The DCGAN (deep conventional GAN) [135] is one of the successful network architectures for GANs. The main contribution of the DCGAN is the use of convolutional layers in the GAN framework which provides stable training in most cases and produces higher resolution images.

Rather than generating addition samples, another class of innovative variants of GAN attempts to translate data across domains, this consists of learning a mapping between data from a source domain (typically with large samples) and data from a similar target domain (with small samples), such as dogs to wolfs. This helps to compensate the domain with few samples by data from other related domains. Inpix2pix [136], a conditional GAN was used to learn a mapping from an input image to an output image; Inpix2pix learns a conditional generative model using paired images from source and target domains. CycleGAN (Cycle consistent adversarial networks) was proposed by Zhu et al. [137] for image-to-image translation tasks in the absence of paired examples through introducing the cycle consistency constraint. Similarly, Disco GAN [138] and Dual GAN [139] used an unsupervised learning approach for image-to-image translation based on unpaired data, but with different loss functions. CoGAN [140] is a model which also works on unpaired images, using two shared-weight generators to generate images of two domains with one random noise.

Another generative technique to synthesize data using neural networks is the so-called variational autoencoder (VAE). Originally proposed in [141], VAE can be seen as a generative model that learns a parametric latent space of the input domain from which new samples can be generated. This has been mostly exploited for image generation [142]. However recently, VAEs have also been recently used for speech enhancement [143] and also for music sounds synthesis [144].

As reported by many scholars [122, 145], the primary problem with generative augmentations is that it is hard to generate data other than images, and even within image data setting it is very difficult to produce high-resolution output images. Moreover, like any ANN, GAN and VAE require a large amount of data to train and its model can be unstable or it can overfit. Thus, depending on how limited the initial dataset is, generative may not be a practical solution [145].

Data-space vs feature-space augmentation

Basic augmentations discussed above are applied to data in the input space, they are called "data warping" methods [146] as they generate additional samples through transformations applied in the data-space. The main challenge with such augmentation schemes is that they are often tuned manually by human experts. Hence, they are "application-dependent" (transformations are domain-specific) and they require domain expertise to validate the label integrity and to ensure that the newly generated data respects valid transformations (that would occur naturally in that domain).

On the other end of the spectrum, we have "synthetic over-sampling" methods, which create additional samples in feature-space. This class of techniques presents thus the advantage of being domain-agnostic, requiring no specialized knowledge, and can, therefore, be applied to many different types of problems [146, 147]. Synthetic

Minority Over-sampling Technique (SMOTE) [148] is a well-known feature augmentation method which handles imbalanced dataset by joining the k nearest neighbors to form new instances. Adaptive Synthetic (ADASYN) [149] is similar to SMOTE, they function in the same way. By contrast, ADASYN adds a random small bias to the points after creating the samples to make them not linearly correlated with their parents, which increases the variance in the synthetic data. The fact that image datasets are often imbalanced poses an intense challenge for DA. Like SMOTE and ADASYN, a lot of work has emerged focusing on restoring the balance in imbalanced images while creating new samples. Milidiu et al. [312] proposed the Seismo Flow, a flow-based generative model to create synthetic samples, aiming to address the class imbalance. Shamsolmoali et al. [313] introduced a GAN variation called CapsAN that handles the class imbalance problem by coalescing two concurrent methods, GANs and capsule network. Lee et al. [314] showed that pre-training DNNs with semi-balanced data generated through augmentation-based over-sampling improves minority group performance.

Furthermore, by manipulating the vector representation of data within a learned feature space, a dataset can be augmented in a number of ways, DeVries and Taylor [147] discussed adding noise, interpolating, and extrapolating as useful forms of feature space augmentation, while Kumar et al. [150] studied six feature space DA methods to improve classification, including Upsampling, Random Perturbation, Conditional Variational Autoencoder, Linear Delta, Extrapolation and Delta-Encoder.

In situ augmentations vs borrowed augmentations

Common augmentation techniques described so far are self-sufficient, that is they make use of the available small data to generate larger dataset without the need for any external data. For this, we can consider them "In situ augmentations". However, they only work under the assumption that some initial data are available in the first place. In scenarios where no primary data are available, previously discussed techniques are not applicable. A very human-like way to tackle this issue is to ask someone to lend you what you are missing (such as borrowing salt or pepper from a neighbor or asking a dress from a friend). Similarly, instead of being limited only to the available training data, a "Borrowed augmentations" schema -if we may call it- augments data by aggregating and adapting input-output pairs from similar but larger data sets. A typical application of this method is autonomous vehicle where training data can be transferred into a night-to-day scale, winter-to-summer, or rainy-to-sunny scale (Fig. 7). Basically, transforming samples from a dataset to another aims at learning the joint distribution of the two domains and finding transformations between them. This line of research addresses the problem of domain shift [151] known as the dataset bias problem, i.e. mismatch of the joint distribution of inputs between source and target domains. An early work [152] that addressed the problem, proposed to learn a regularized transformation using information-theoretic metric learning that maps data in the source domain to the target domain. This is considered one of the first studies of domain adaptation [153] in the context of object recognition. However, this approach requires labeled data from the target domain as the input consists of paired similar and dissimilar points between the source and the target domain. In contrast, Gopalan et al. [154] proposed a domain adaptation technique for an unsupervised setting, where data from the target domain is unlabeled. The domain shift, in this case, is obtained by generating intermediate subspaces between the source and target domain, and projecting both the source and target domain data onto the subspaces for recognition. Unsupervised domain adaptation has been largely investigated afterward [155–158]. Recently, it was shown that a GAN objective function can be used to learn target features indistinguishable from the source ones. Hence, most recent works regarding data transportation cross-domains are based on generative models. For instance, the aforementioned technique of image-to-image translation based on GANs is a successful example of such schema, other similar techniques include neural style transfer (translate images from one style to another) [159], Text-to-Image Translation [160], Audio-to-Image Generation [161], Text-to-Speech synthesis [162] … etc. By relying on GAN, other recent works made use of this model to boost performance. Wang et al. [163] proposed Transferring GANs (TGANs) which incorporate a fine-tuning technique into GAN, to train this latter with low-volume target data. Yamaguchi et al. [164] import data contained in an outer dataset to a target model by using a multi-domain learning GAN. Huang et al. [165] proposed AugGAN, a cross-domain adaptation network, which allows to directly benefit object detection by translating existing detection RGB data from its original domain other scenarios. As one may note, while most works address transferring data cross domain for image generation, the challenge is still modestly explored in other domains [166].

Supervised vs unsupervised data augmentation

Augmentations schemas are class-preserving transformations, they rely on labeled data (supervised augmentation). However, if getting more data is hard, getting more labeled data is harder. Whilst collecting unlabeled data is easier and cheaper as human effort is not needed for labeling, a major issue is how to augment data without labels. Typically, SSL and unsupervised methods discussed previously are the best candidates to address the issue. Remarkably, tackling the challenge of using unlabeled data has been the subject of relatively few works in the literature in comparison with supervised augmentation methods. In recent work, Xie et al. [167] showed that data augmentation can be performed on unlabeled data to significantly improve semi-supervised learning. Their model relies on a small amount of labeled examples to make correct predictions for some unlabeled data, from which the label information is propagated to augmented counterparts through the consistency loss. Aside cycle consistency regularization, the commonly used approach for augmenting smaller labeled datasets using larger unlabeled datasets is self-training or more generally co-training [75], as discussed in the previous strategy, this type of training relies on an iterative process that use pseudo-labels on unsupervised data to augment supervised training. Always with the goal of leveraging a large amount of unlabeled data and a much smaller amount of labeled data for training, others methods have been proposed in the literature using methods such as Temporal Ensembling [101], Mean Teacher [102], self-paced learning [168], and data programming [169].

Discussion

To sum up, there no best augmentation schema, the choice of the technique to use depends on the application scenario. When no data is available, borrowed augmentations

Table 2 Summary of augmentation methods

Augmentation techniques	References
Basic transformation	[125, 126, 128]
Generative augmentation	[131–140 142–144]
Feature-space augmentation	[147–150]
Domain adaptation	[153–158, 165]
Unsupervised augmentation	[167, 75, 101, 102, 168, 169]

should be considered. When a large amount of unlabeled samples exists, unsupervised augmentations are the best choice. Fig. 7; Table 2 depicts the main reviewed DA techniques. However, it is noteworthy that there are very few studies in the literature that compare empirically the performance of the different augmentations. Wong et al. [146] compared data-space or feature-space and found that it was better to perform data augmentation in data-space, as long as label preserving transforms are known. Shijie et al. [170] compared generative methods with some basic transformations. They found the combinations of the two types of augmentation drive better performance. Indeed, the choice of combining augmentation techniques can result in massively inflated dataset sizes. However, this is not guaranteed to be advantageous, especially in very limited data setting, this could result in further overfitting [122]. Furthermore, the classes of techniques described in this section are neither mutually exclusive nor exhaustive. That means depending on the complexity, the space, the domain, and the data annotability on which the augmentation occurs, techniques can belong to different classes. For example, generative augmentations like cycleGAN are used to implement image to image translation, which is a type of borrowed augmentations. GANs have been also exploited in the context of unsupervised augmentation. For instance, Wang et al. [171] proposed a variant of CycleGAN (DicycleGAN) that performs an unsupervised borrowed augmentation based on a generative model.

Regardless their numbers and capacities, current DA implementations remain manually designed. A key research question is then to find automatically the effective DA schema for a given dataset by searching in a large space of candidate transformations. State-of-the-art approaches to address this problem include TANDA a framework proposed by Ratner et al. [172] to learn augmentations based on GAN architecture. And, AutoAugment [173] demonstrated state-of-the-art performance using a reinforcement learning algorithm to search for an optimal augmentation technique amongst a constrained set of transformations with miscellaneous levels of distortions. Several subsequent works including RandAugment [174] and Adversarial AutoAugment [175] have been proposed to reduce the computational cost of AutoAugment, establishing new state-of-the-art performance on image classification benchmarks.

As noted several times before, DA has essentially been used to achieve nearly all state-of-the-art results for image data, particularly for medical imaging analysis. In this domain where high-quality supervised samples are generally scarce and fraught with legal concerns regarding patient privacy, image augmentation is considered a de facto technique [176–178]. Medical data suffer also from the so-called “p large, n small” problem (where p is the number of features and n is the number of samples), hence, some works [310] attempted to fight the curse of data dimensionality along with the curse of data scarcity by proposing a dimensionality reduction-based method that can be used for data augmentation. Unfortunately, dataset augmentation is not as straightforward to apply in other domains as it is for images. Current effort of exploring DA in others non-image domains includes mainly sound, speech, and text augmentation. In this vein, Schluter and Grill [179] investigated a variety of DA techniques for application to singing voice detection. Wei et al. [180] proposed a text augmentation technique for improving NLP application performance.

Knowledge sharing

A common assumption in most ML algorithms states that the training and future (unknown) data must be drawn from the same data space and have to follow the same distribution [182] (as stressed before, following the PAC-learnability criteria, the distribution **D** must be stationary-see the "Background" section). This implies that when the task to be learned or its domain change, the model needs to be rebuilt from scratch using newly collected training data. This paradigm is called single task learning or isolated learning. The fundamental problem with this way of learning is that it does not consider any other related information or the previously learned knowledge to alleviate the need for training data for a giving task. This is in sharp contrast of how we humans learn. As discussed in the "Background" section, human learning is very knowledge-driven: we accumulate and maintain the knowledge learned from previous tasks and use it seamlessly in learning new tasks and solving new problems with little data and effort. Towards the ultimate goal of building machines that learn like humans, some research areas attempted to break the training data exclusive dependency by exploring the idea of using prior knowledge as additional inputs for ML models apart from standard training data. We characterize this family of approaches as knowledge sharing strategy. Depending on how, when and what extent of knowledge is shared, the research is conducted under different guises, however all approaches share the same spirit: reusing knowledge instead of relying solely on the tasks' training data. Next, we investigate the four main ways of sharing knowledge found in the literature, namely (A) Transfer Learning, (B) Multi-Task-Learning, (C) Lifelong Learning, and (D) Meta-Learning.

Transfer Learning

Inspired by human beings' capabilities to transfer knowledge across tasks, Transfer Learning (TL) aims to improve learning and minimize the amount of labeled samples required in a target task by leveraging knowledge from the source task. Following the Pan et al. [182] definition: *given a source domain D_S and a learning task T_S, a target domain D_T and a learning task T_T, TL aims to help improve the learning of the target predictive function fT(.) in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$*. Accordingly, TL allows the tasks and distributions used in training (source) and testing (target) to be different. When the target and source domains are the same, i.e., $\mathrm{D_S} = \mathrm{D_T}$, and their learning tasks are the same, i.e., $\mathrm{T_S} = \mathrm{T_T}$, the learning problem becomes a traditional ML problem.

Surveys [182] and [183] proposed and discussed a taxonomy of TL which has been widely accepted and used. Depending on the availability of labeled data in source and/or target data, they distinguished between [182]: (i) inductive TL, (ii) transductive TL and (iii) unsupervised TL, which correspond respectively to the case of having available labeled target domain data, the case of having labeled source and no labeled target domain data, and the case of having no labeled source and no labeled target domain data. Domain adaptation, the DA technique discussed before is a type of transductive TL in which the source task and the target task are the same but their domains are different. Furthermore, regardless of the availability of labeled and unlabeled data, TL problems can generally be categorized into two main classes [183]: homogeneous transfer learning and heterogeneous transfer learning, the former category focused on generalization performance across the same domain representations, meaning that the samples in a source domain and those in a target domain share the same representation structure but follow different probability distributions, the majority of TL approaches belong to this category. In the latter category, the feature spaces between the source and target are nonequivalent and are generally non-overlapping, this case is more challenging as knowledge is available from source data but it is represented in a different way than that of the target. This method thus requires feature and/or label space transformations to bridge the gap for knowledge transfer, as well as handling the cross-domain data distribution differences.

The effectiveness of any transfer method depends on the source task and how it is related to the target task. A transfer method would produce positive transfer between appropriately related tasks, while negative transfer occurs when the source task is not sufficiently related to the target task or if the relationship is not well leveraged by the transfer method [184]. Increasing positive transfer, and avoiding negative transfer is one of the major challenges in developing transfer methods.

TL methods in the literature share the same function: leveraging the knowledge in the source domain. Three classes of TL methods can be defined based on the type of the shared knowledge: instance, feature, or model (parameter), accordingly we can distinguish between: (i) instance-based TL approaches that reuse labeled data from the source domain by re-weighting or resampling instances to help to train a more precise model for a target learning task [185]. (ii) feature-based TL approaches, the transfer in this type of approaches is operated in an abstracted "feature space" instead of the raw input space. The aim is to minimize domain divergence and reduce error rates by identifying good feature representations that can be utilized from the source to target domains [186]. And Model-based TL, also known as parameter-based TL, here the transferred knowledge is encoded into model parameters, priors or model architectures. Therefore, the goal of this class of approaches is to discover what part of the model learned in the source domain can help the learning of the model for the target domain [187]. Model-based TL is arguably the most frequently used method. Additionally, we also identified relational based TL where data are non-independent and identically distributed. The three main TL approaches implicitly assume that data instances are independent and identically distributed. However, in real-world scenarios often contain some structures among the data instances, leading to relational structures in these domains, like for example social network domain. A family of approaches called relational-based TL attempts to handle this issue by building a mapping of the relational knowledge between the source relational domain and the target relational domain [188].

In the studied literature, TL methods are used in the classic learning tasks including classification, regression, and clustering tasks, relatively fewer but impactful works have also handled TL for reinforcement learning [189]. Success applications of TL include computer vision [190], NLP [191], and urban computing [192]. Emerging and promising research lines in TL include (i) Hybrid-based approaches, TL solutions that focus on transferring knowledge through the combination of different TL methods, for instance by using both instances and shared parameters. This is relatively a new approach and a lot of interesting research is emerging [193]. (ii) Deep transfer learning, as deep learning becomes a ubiquitous technique, researchers have begun to endow deep models with TL capabilities. The powerful expressive ability of deep learning has also been leveraged to extract and transfer knowledge such as the relationships among categories. Fine-tuning [194] is a glaring example of popular and effective technique for knowledge transfer in terms of model parameters based on pre-trained models. The knowledge distillation technique [195], which involves a teacher network and a student network, is also a good example of this line of work. (iii) Transitive TL [196], a new type of TL problem where the source and target domains have very few common factors, making most TL solutions invalid. Always by following the human learning model which can conduct transitive inference and learning, novel TL solutions have proposed to connect the source and target domains by one or more intermediate domains through some shared factors. (iv) AutoTL, addresses the issue of learning to transfer automatically [197]. Wei et al. [198] proposed a transfer learning framework L2T that automatically explores the space of TL method candidates to discover and apply the optimal TL method that maximally improves the learning performance in the target domain.

Multi-task learning

If a TL method aims to improve the performance of the source task and target task simultaneously, we are dealing with a Multi-task learning (MTL) problem [199]. MTL shares the general goal of leveraging knowledge across different tasks. However, unlike TL there is no distinction between source and target tasks, multiple related tasks each of which has insufficient labeled data to train a model independently, are learned jointly using a shared representation. The training data from the extra tasks serve then as inductive bias, acting in effect as constraints for the others, improving general accuracy, and the speed of learning. As a result, the performance of all tasks is improved at the same time with no task prioritized. MTL is clearly close to TL, in some literature it is even considered as a type of inductive TL [182], this is why it is generally acknowledged that MTL problem could be approached with TL methods, however the reverse is not possible [200]. Some works investigated hybrid scenarios where new task is arrived when multiple tasks have been already learned jointly by some MTL method. This could be seen as MTL problem for old tasks and TL problem to leverage knowledge from the old tasks to the new task. Such setting is called asymmetric multi-task learning [201].

A variety of different methods has been used for MTL, basically to each nature of the learning task corresponds a different setting in MTL [202]. Accordingly, (i) the multi-task supervised learning is based on training labeled data for each task. As for TL, researches in this area have been conducted on three categories, that are, (a) feature-based multi-task supervised learning, specifically the problem of feature-selection [203] and feature transformation [204]. (b) Model-based multi-task supervised learning, notably, the low-Rank approach [205], the task clustering approach [206], and task relation learning approach [207]. Finally, very modest contributions have been done on the third category, (c) instance-based multi-task supervised learning [208]. (ii) In multi-task unsupervised learning, each task deals with discovering useful patterns in data. (iii) In multi-task semi-supervised learning, tasks based their predictions on labeled data as well as unlabeled data. (iv) In multi-task active learning, each task selects representative unlabeled data to query an oracle with the hope to reduce the labeling cost as much as possible. (v) In multi-task reinforcement learning, each task aims to maximize the cumulative reward by choosing actions. (vi) In multi-task multi-view learning, each task exploits multi-view data. Recent years witness extensive studies on streaming data, known as online multi-task learning [209], this class of methods is used when training data in multiple tasks arrive sequentially, hence (vii) in multi-task online learning, each task is to process sequential data.

In settings where MTL consists of tasks with different types including supervised learning, unsupervised learning, reinforcement learning…etc., the MTL is characterized as heterogeneous. In contrast to the homogeneous MTL which consists of tasks of the same type. Unless it is explicitly underlined, the default MTL setting is the homogeneous one [202].

Given the nature of its process, MTL has been studied under the decentralized settings where each machine learns a separate, but related, task. In this vein, multiple parallel and distributed MTL models have been introduced in the recent literature [209–211]. Recently, research in MTL using DNN has produced a wide spectrum of approaches that have yielded impressive results on some tasks and application such as image processing [212], NLP [213] and biomedicine [214]. Conversely, there have been exciting results using MLT methods in DNN. Generally, there are two commonly used approaches to carrying out MTL in deep learning: hard and soft [215]. Hard parameter sharing implies the sharing of hidden layers between all tasks, and the output layers are different. Soft parameter sharing gives each task its own model with its own parameters, where these model parameters have a regularized distance to facilitate the sharing of learning.

Lifelong learning

One of the long-standing challenges for both biological systems and computational models (especially ANN) is the stability-plasticity dilemma [216]. The basic idea is that a learner requires plasticity for the integration of new knowledge, but also stability in order to prevent the forgetting of previous knowledge. The dilemma is that while both are desirable properties, the requirements of stability and plasticity are in conflict. Stability depends on preserving the structure of representations, plasticity depends on altering it. An appropriate balance is difficult to achieve. Generally, ANN models tend often to have excessive plasticity, a problem that is dramatically referred to as "catastrophic forgetting" (or "catastrophic interference") [216] which basically means the loss or disruption of previously learned knowledge when a new task is learned. Recently, a number of approaches have been proposed to mitigate catastrophic forgetting. They aim to design models that are sensitive to, but not disrupted by, new data. These approaches are categorized as lifelong/continual learning (LL) approaches. LL embodies a knowledge sharing process as it makes use of prior knowledge from the past observed tasks to help continuously learning new/future tasks. Hence, LL studies scenarios where a large number of tasks come over time. Thus, to deal with the continuous stream of information, LL approaches include essentially two elements: (a) a retention strategy to sequentially retain previously learned knowledge and (b) a transfer mechanism to selectively transfer that knowledge when learning a new task. Most of the research effort in LL has focused primarily on how to retain knowledge, in doing so, the focus has been shifted to counter catastrophic forgetting. various approaches have been proposed in this sense including (i) architectural methods, (ii) regularization methods, and (iii) rehearsal methods [217]. A high-level analysis of LL literature shows that since its introduction 25 years ago in [218], LL concept has mainly evolved in respect of the four-learning paradigms:

(i) *Lifelong supervised learning*: Early contributions in this area were based on memory systems and neural networks. Thrun [219] proposed two memory-based learning methods: k-nearest neighbors and Shepard's method. Although they are still used today, memory-based systems suffer from the drawback of large working memory requirements as they require explicit storage of old information [216]. On neural networks level, initially, Thrun and Mitchell worked [220] on a LL approach called explanation-based neural networks EBNN. Since, Silver et al. have extensively work on the extension and the improvement of the neural network approaches through many works [221–223]. Furthermore, a lifelong naive bayesian classification technique was proposed by Chen et al. [224], which is applied to a sentiment analysis task. Ruvolo and Eaton [225] proposed an efficient LML algorithm (ELLA) to improve an MTL method to make it a LL method. Clingerman and Eaton [226] proposed GP-ELLA to support Gaussian processes in ELLA.

(ii) *Lifelong unsupervised learning*: Works in this area are mainly about lifelong topic modeling and lifelong information extraction. Lifelong Topic Modeling approaches extract knowledge from topic modeling results of many previous tasks and utilizes the knowledge to generate coherent topics in the new task related works in this vein include [227, 228]. As the process of information extraction is by nature continuous and cumulative, information extraction represents an evident area for applying LL. Significant works of this line of research include [229, 230].

(iii) *Lifelong semi-supervised learning*: The most well-known and impactful work in this area is NELL, which stands for Never-Ending Language Learner [231 − 230]. NELL is a lifelong semi-supervised learning system that has been reading the Web continuously for information extraction since January 2010, and it has accumulated millions of entities and relations.

(iv) *Lifelong reinforcement learning*: Thrun and Mitchell [218] first studied lifelong reinforcement learning for robot learning. Recently, many works have been proposed in this area due to the recent surge in research in RL after being successfully used in the computer program. Bou Ammar et al. [232] presented a policy gradient efficient lifelong reinforcement learning algorithm. Tessler et al. [233] proposed a lifelong learning system that transfers reusable skills to solve tasks in a video game.

> Rolnick et al. [234] introduced CLEAR, a replay-based method that greatly reduces catastrophic forgetting in multi-task reinforcement learning.

By analyzing the LL literature, we note that despite the first pioneering attempts and early speculations, research in this field has never been carried out extensively until the recent years. In their book, Chen et al. [235] emphasized some reasons behind the slow advancement. The main reason according to them is that ML research in the past 20 years focused only on statistical and algorithmic approaches. Moreover, much of the past ML research and applications focused on supervised learning using structured data, which are not easy for LL because there is little to be shared across tasks or domains. They also underlined the fact that many effective ML methods such as SVM and deep learning cannot easily use prior knowledge even if such knowledge exists. However recently as most of the limits caused by these factors have been exceeded, LL is becoming increasingly a rich area of scientific contributions and new approaches have emerged. Notably, continual learning in DNN [216] and lifelong interactive knowledge learning for chatbots [236]. Still, we believe that existing LL literature does not sufficiently cover the evaluation aspect, that is what makes a LL system successful, how to compare existing LL algorithms, and what metrics are most useful to report. Hence, much more efforts are expected in the research area for years to come.

Meta-learning

Meta-learning, or learning to learn (LTL), improves the learning of a new task by using meta-knowledge extracted across tasks [237]. In a nutshell, LTL treats learning tasks as learning examples. It aims to improve the learning algorithm itself, given the experience of multiple learning episodes. Basically, in a meta-learning system, we distinguish the meta-learner, which is the model that learns across episodes, and the inner-learner, which is instantiated and trained inside an episode by the meta-learner. More specifically, the inner-learner model, typically a CNN classifier, is initialized, and then trained on the support set (e.g., the base training set). The algorithm used to train the inner-learner is defined by the meta-learner model. This latter, updates the inner-learner to be able to improve while solving a task in the classic way (base learning) with only a very small set of training examples. At the end of the episode, the meta-learner's parameters are trained from the loss resulting from the task learning error [238]. Thus, meta-learning is tightly linked to the process of collecting and exploiting meta-knowledge. Meta-knowledge collecting is performed by extracting algorithm configurations such as hyperparameter settings, pipeline compositions and/or network architectures, the resulting model evaluations, the learned model parameters, as well as measurable properties of the task itself, also known as meta-features. Then the meta-knowledge is transferred to guide the search for optimal models for new tasks [239].

From our perspective, we consider LTL a tool for knowledge sharing more than an approach of reusing knowledge per se. Indeed, in the scanned literature, LTL is usually introduced as a method to solve other knowledge-sharing scenarios. Particularly, LTL is commonly described as the de facto method to solve few-shot learning (FSL) problems [7], a regime where only few experiences are available. Therefore, we propose in the following to review LTL methods in respect of the three discussed approaches, namely: TL, MTL, and LL, while shedding light on FSL, the most popular instantiation of LTL in the field of supervised learning.

A. Meta-learning-based methods for FSL

As the name implies, FSL refers to the problem of learning a new concept or task with only a few training examples or no pre-labeled learning example [7]. FSL is not a knowledge sharing approach itself, but it is an umbrella term encompassing techniques that make use of prior knowledge methods to deal with data scarcity scenarios. There are three main variants of FSL, (i) zero-shot learning [240], which deals with learning a task that has no associated labeled training samples, (ii) one-shot learning [241] where tasks are learned from a single example, and (iii) low shot learning, assumes that a handful (typically 2–5) labeled examples exist for target/novel classes. Recently, FSL has sparkled with several successful applications in literature including few-shot classification [242], few-shot object detection [243], semantic segmentation [244], and landmark prediction [245]. Generally, existing FSL models fall into two main groups, (i) Hallucination-based methods (practically data augmentation) deal directly with the data scarcity by "learning to augment", however DA could alleviate the issue, but does not solve it. In this section, we focus on the second group (ii) Meta-learning-based methods that tackle the FSL problem by "learning to learn". The majority of this class of methods can be labeled as either a metric learning algorithm or as a gradient-based meta-learner.

(i) *Metric learning algorithm*: These methods address the FSL problem by "learning to compare". The basic idea of metric learning is to learn a distance function between data points (like images). It has proven to be very useful for solving FSL problem for classification tasks: instead of having to fine-tune on the support set (the few labeled images), metric learning algorithms classify query images by comparing them to the labeled images. Koch et al. [246] proposed the Siamese Neural Networks to solve few-shot image classification. Their model learns a siamese network by metric-learning losses from a source data, and reuses the network's features for the target one-shot learning task. Vinyals et al. [247] proposed Matching Networks that use an episodic training mechanism. Snell et al. [248] introduced prototypical Networks that learn a metric space in which classification can be performed by computing distances to prototype representations of each class. Sung et al. [249] proposed Relation Network that uses CNN-based relation module as a distance metric. Li et al. [250] designed a model named Covariance Metric Networks (CovaMNet) to exploit both the covariance representation and covariance metric based on the distribution consistency for the few-shot classification tasks. Wertheimer et al. [251] localized objects using a bounding box. Garcia et al. [242] used Graph Neural Network based model. Despite the rich contributions in this line of research, relation measure, that is how to robustly measure the relationship between a concept and a query image remains a key issue in this class of FSL methods.

(ii) *Gradient-based Meta-Learning*: These methods address the FSL problem by "learning to optimize". They embed gradient-based optimization into the meta learner.

More specifically, in such models, there is an inner- loop optimization process that is partially or fully parameterized with fully differentiable modules. The methods of this class differ according their choice of the meta-model algorithm. The most famous meta-learners in the literature are perhaps (i) Meta-LSTM introduced by Ravi & Larochelle [252], a meta-learner that uses a Long-Short-Term-Memory network to replace the stochastic gradient decent optimizer and the weight-update mechanism. And (ii) Model-Agnostic Meta-Learning (MAML) [253] is currently one of the most elegant and promising LTL algorithms. MAML provides a good initialization of a model's parameters to achieve optimal fast learning on a new task with only a small number of gradient steps. This method is compatible with any model trained with gradient descent (model-agnostic), and has been shown to be effective in many classification and reinforcement learning applications. Following this line of work, many recent studies [315–318] focused on learning better initialization by adaptively learning task-dependent modifications. In these works, the inner-loop optimization is generally based on first-order optimizer algorithms such as SGD and Adam. A few recent studies propose optimizer-centric approaches [319–321], they are models that focus not only on adjusting the optimizer algorithm but on learning the inner optimizer itself.

B. Meta-learning in TL setting

There are many works in the literature that combined TL with LTL. Aiolli [254] proposed an approach to transfer learning based on meta kernel learning. Eshratifar et al. [255] propose a joint training approach that combines both TL and meta-learning loss functions into a learning algorithm. Sun et al. [256] proposed a novel FSL method called meta-transfer learning which learns to adapt a DNN for FSL tasks. Later, the authors introduced the hard task meta-batch scheme as a learning curriculum that further boosts the learning efficiency of the proposed meta transfer learning [257]. Li et al. [258] bring forward a novel meta-transfer feature method (MetaTrans) for measuring the transferability among domains. Some of the recent applications of meta-transfer learning include meta-transfer learning for learning to disentangle causal mechanisms [259], meta-transfer learning for zero-shot super-resolution [260], meta-transfer learning for code-switched speech recognition [261], and meta transfer learning for adaptive vehicle tracking in UAV Videos [262].

3 Meta-learning in MTL setting

LTL has recently emerged as an important direction for developing algorithms for MTL. Indeed, meta-learning can be brought in to benefit MTL, notably by learning the relatedness between tasks or how to prioritize among multiple tasks. In this vein, Lin et al. [263] proposed an adaptive auxiliary task weighting to speed up training for reinforcement learning. Franceschi et al. [264] proposed a forward and reverse gradient-based hyperparameter optimization for learning task interactions. Epstein et al. [265] proposed a meta-learning framework for extracting sharing features among multiple tasks that are learned simultaneously. Chen et al. [266] used a shared meta-network to

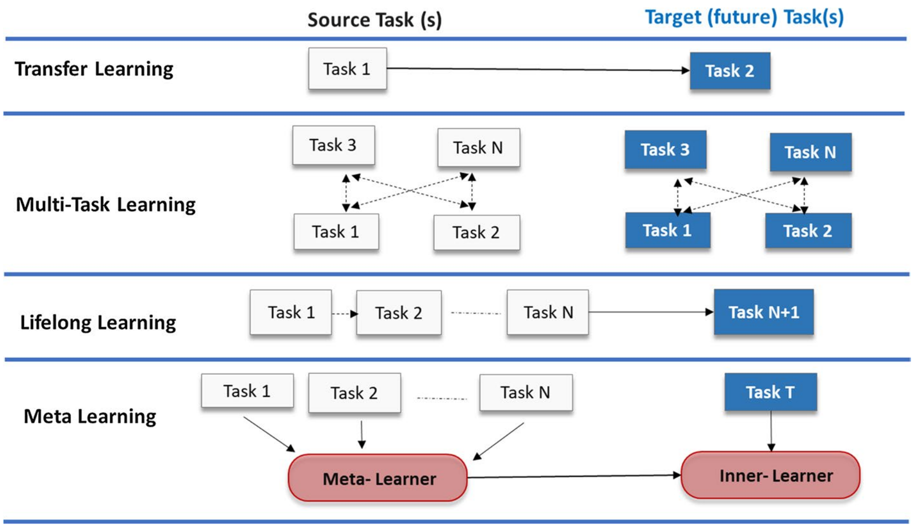

Fig. 8 Knowledge sharing methods

Table 3 Summary of knowledge sharing methods

Knowledge sharing methods	References
Transfer learning	[185–189, 193–196, 198]
Multi-Task learning	[201, 203–209]
Lifelong learning	[219, 220, 224–230] [232–234]
Meta learning	[258, 264],[266, 268, 271– 273]
Few-shot learning	[246–253]

capture the meta-knowledge of semantic composition and generate the parameters of the task-specific semantic composition models in MLT setting. Amit et al. [267] proposed a PAC-Bayes meta-learning method designed for multi-task learning.

4 Meta-learning in LL setting

LL can also be realized through LTL. Riemer et al. [268] proposed a framework called Meta-Experience Replay (MER) that integrates meta-learning and experience replay for continual learning. Javed et al. [269] proposed OML, a meta-learning objective that directly minimizes catastrophic interference by learning representations that accelerate future learning and are robust to forgetting under online updates in continual learning. He et al. [270] proposed task agnostic continual learning framework based on meta-learning that is implemented by differentiating task specific-parameters from task agnostic parameters, where the latter are optimized in a continual meta-learning fashion, without access to multiple tasks at the same time. Munkhdalai et al. [271] introduced a meta-learning model called MetaNet that supports meta-level LL by allowing ANN to learn and to generalize a new task or concept from a single example on the fly. Vuiro et al. [272] proposed a meta-training scheme to optimize an algorithm for mitigating catastrophic forgetting. Xu et al. [273] described an LTL method to improve word embeddings for a lifelong domain without a large corpus.

Discussion

In this section, by knowledge sharing we referred to all types of learning based on prior experiences with other tasks. When, how, and what knowledge is shared determinate the class of methods (Table 3 summarizes the reviewed class of methods). Nevertheless, throughout the literature, we noted a number of terminology inconsistencies. Phrases such as "transfer learning" and "multi-task learning" or "few-shot learning" and "meta-learning" are sometimes used interchangeably. This is often a source of confusion as the studied concepts are closely related and boundaries between them aren't always clear. Certainly, the reviewed approaches are similar in their common goal of knowledge reuse, however, they are different in their specific ways to handle knowledge transfer (Fig. 8 highlight the transfer mechanism of each approach). TL improves the learning of a target task through the transfer of knowledge from a related source task that has already been learned. MTL considers how to learn multiple tasks in parallel, at the same time, and exploit their intrinsic relationship, such that they help each other to be learned better. LL is sequential learning that continually learns overtime by accommodating new knowledge while retaining previously learned experiences. Meta-learning transfers meta-knowledge across tasks, it can thus be considered a meta-solution to transfer knowledge in TL, MTL, and LL. FSL is a problem and not a solution, that studies learning tasks with a few experiences. Hence, reviewed knowledge sharing solutions can be used to solve this problem, particularly meta-learning approaches. Among the five concepts, TL is probably the largest one, as all reviewed approaches involve, at some level, transfer related operations. However, it is important to note that TL is unidirectional, its goal is to improve the learning of only the target task, learning of the source task(s) is irrelevant and not considered. Similarly, LL (in its vanilla version) only transfer knowledge forward to help future learning and do not go back to improve the model of previous tasks. While in MTL all tasks and data are provided together, allowing the model to be trained on and then to improve all tasks at the same time, but at a potentially high computational cost. Recently, backward or reverse knowledge transfer is increasingly studied in the context of LL [274]. Furthermore, TL and MLT typically need only few similar tasks and do not require the retention of explicit knowledge. LL, on the other hand, needs significantly more previous tasks in order to learn and to accumulate a large amount of explicit knowledge so that the new learning task can select the suitable knowledge to be used to help the new learning. Hence, the growth of the number of tasks and knowledge retention are key characteristics of LL, this why many optimization efforts have been observed in the presented literature regarding these two aspects. On another note, meta-learning trains a meta-model from a large number of tasks to quickly adapt to a new task with only a few examples. It can be useful for better knowledge retention through metric learning or for measuring relatedness between tasks or to select the useful knowledge to be transferred. However, one key assumption made by most meta-learning techniques is that the training tasks and test/new tasks are from the same distribution, while other approaches do not make this assumption. This is a major weakness that can limit the scope of LTL application and which has to be seriously addressed in the future LTL research.

Despite the underlined differences, clearly, knowledge sharing approaches are closely related, they share many challenging issues that are expected to preoccupy the future literature in this field as well as key characteristics that allow them to work collaboratively and synergistically. For example, if we continuously apply TL in a learning system, we can obtain a lifelong machine learning system, inversely we can view TL as LL system in the particular case where the number of the tasks is two. On the other hand, LL could also be considered as online MTL where we deal with multiple tasks, and data points arrive in sequential order. Another special case of LL that is worth to be mentioned that at level, is curriculum learning [275]. Similarly to MLT, in this case, all tasks and data are made available, but the problem is to identify the optimal order in which to train on data for the most efficient and effective learning. An intuitive type of curriculum is to learn tasks from "easy" to "hard" (similar to the way humans often learn new concepts). Another common characteristic is the regularization effect, knowledge sharing approaches, especially those dealing with multiple tasks, benefit from the effect regularization due to parameter sharing and of the diversity of the resulting shared representation. They also somehow implicitly augment data (e.g., domain adaptation).

On the other end of the spectrum, knowledge sharing approaches share also the same concerns. Notably, the effectiveness of all reviewed approaches depends on the task relatedness, defining task similarity is a key overarching challenge. As mentioned before, considerably less attention has been given to the rigorous evaluation to compare between methods of the same approach or between approaches of different nature. Also, dealing with knowledge implies to answer some important questions such as what forms of knowledge are important, how to represent them, and what kinds of reasoning capabilities are useful, since reasoning allows the system to infer new knowledge from existing knowledge, which can be used in the new task learning. However, so far, little research has been done to address these questions in knowledge sharing literature. Hence, we believe that research in knowledgeable systems needs more engagement and wider attention of academic researchers, more efforts are expected in order to bring this fields to maturity and make it able to compete classical paradigms of learning.

2.4 Hybrid learners

Data hungriness is mainly related to DNN when they are used in a supervised fashion, these models represent a branch of learning called connectionism. Another potential strategy to cure hungriness would be then to go out of the box and to look for other branches of learning that are more data-efficient. In his recent book, Domingos [276] has drawn borders between five schools of thoughts in ML, namely symbolists, connectionists, evolutionaries, bayesians, and analogizers. Driven by the same goal of building learning machines, each type of learner makes different assumptions about data. Evolutionaries take roots in evolutionary biology, they use genetic algorithms to deal with structure discovery problem. By being basically research and optimization algorithms, learners of this family require relatively less data. They are mainly used to optimize other hungry learners [277, 278] but they are known to be costly. Bayesians find their origins in statistics, they use probabilistic inference to cope with uncertainty. Algorithms of this family are mostly supervised such as SVM, accordingly they require a large amount of data. Similarly, analogizers also need data about the solution of a known situation to transfer it to a new situation faced using mainly Kernel machines, recommender systems are the most famous application of analogy-based learning. Generally, all three families obey the rule of "more data, better learning". However, connectionists represented by ANN are without a doubt the most data-driven tribe, inspired by neuroscience this branch produces learning algorithms to find the connection weights that make it possible for a neural network to accomplish some intelligent task. Connectionism is generally associated with an empiricist position that considers all of mind as the result of learning and experience during life. According to connectionists experiences/data are the only sources of learning, the more data we have the more we can learn [276]. On the other end of the spectrum, symbolists are arguably the most data–efficient tribe. Symbolists view learning as the inverse of deduction and take ideas from philosophy, psychology, and logic. They presume that the world can be understood in the terms of structured representations and assume that intelligence can be achieved by the manipulation of symbols, through rules and logic operating on those symbols to encode knowledge [279]. "Symbolic" AI is considered as the classic AI, it is sometimes referred to as GOFAI (Good Old-Fashioned AI). It was largely developed in an era with vastly less data and computational power than we have now. Symbolic AI bases its intelligent conclusions and decisions on the memorized facts and rules rather than raw massive data. However, it suffers from several drawbacks regarding generalization and change adaptation that, interestingly, are the strengths of connectionists models. The right move would be then to integrate connectionists models, which excels at perceptual classification, with symbolic systems, which excel at inference and abstraction. This movement is known in the literature as Neural-Symbolic Computing (NSC) [280].

NSC aims at integrating robust connectionist learning and sound symbolic reasoning. The idea is to build a strong hybrid AI model that can combine the reasoning power of rule-based software and the learning capabilities of neural networks. In a typical neural-symbolic system, knowledge is represented in symbolic form, whereas learning and reasoning are computed by a neural network. Hence, the symbolic component takes advantage of the neural network's ability to process and analyze unstructured data. Meanwhile, the neural network also benefits from the reasoning power of the rule-based AI system, which enables it to learn new things with much less data. It is believed that this fusing would help to build a new class of hybrid AI systems with a non-zero-sum game conception that are much more powerful than the sum of their parts [280]. It is also claimed that this way of perceiving intelligence is much more analogical to the brain that uses mechanisms operating in the two fashions [4]. In that NSC is expected to bring scientists closer to achieving true artificial human intelligence.

The integration of the symbolic and connectionist paradigms has been pursued by a relatively small research community over the last two decades. Recently, with the strong penetration of DNN and the rise of complaints regarding explainability and data hungriness of these models. NSC has yielded several significant results that have shown to offer powerful alternatives for opaque data-hungry DNN. Yi et al. [281] proposed NS-VQA, neural-symbolic visual question answering approach that disentangles reasoning from visual perception and language understanding. The model uses DNN for inverse graphics and inverse language modeling, and a symbolic program executor to reason and answer questions. According to the authors, incorporating symbolic structure as prior knowledge offers three advantages: (i) robustness, (ii) interpretability, and (iii) data efficiency. They verified that the system performs well after learning on only a small number of training data. In the same vein, Vedantam et al. [282] also demonstrated that their neural-symbolic VQA model performs effectively in low data regime. Evans et al. [283] proposed a differentiable inductive logic framework which is a reimplementation of traditional Inductive Logic Programming (ILP) in an end-to-end differentiable architecture. The framework attempts to combine the advantages of ILP with the advantages of the neural network-based systems; a data-efficient induction system that is robust to noisy and ambiguous data, and that does not deteriorate when applied to small data.

Furthermore, the idea of neural-symbolic integration has also tempted knowledge transfer community. The idea is to extract symbolic knowledge from a related domain and transfer it to improve the learning in another domain, starting from a network that does not necessarily have to be instilled with background knowledge [284]. In this vein, Silver [285] discussed the link between NSC and LL, he exposed an integrated framework for neural-symbolic integration and lifelong machine learning where the symbolic component helps to retain and/or consolidate existing knowledge. Hu et al. [286] proposed a self-transfer approach with symbolic-knowledge distillation. They developed an iterative distillation method that transfers the structured information of logic rules into the weights of neural networks. The transferring is done via a teacher training network constructed using the posterior regularization principle. The proposed framework is applicable to various types of neural architectures, including CNN for sentiment analysis, and RNN for named entity recognition.

Discussion

All in all, there is no general standard solution regarding how to cure data hungriness, many perceptions exist but none of them can be asserted to be an absolute solution. Beyond research laboratories, results produced in real-world conditions indicate that existing techniques are yet to be industrialized. And more importantly, with the absence of rational metrics to evaluate and compare techniques, we cannot objectively justify the choice of a technique over another. That being said, we believe that research on this issue is just in its infancy. Without a doubt, considering the facts from industrial and academic worlds, it is a golden time for data-efficient algorithms to rise. However, considering what has been done in the literature so far, improvements are expected from the community working on the issue in order to advance research in this area. In this section, we discuss some research directions and open challenges distilled from the surveyed works, we propose to group them in four themes, namely: (i) Hybridization, (ii) Evaluation, (iii) Automation, and (iv) Humanization.

- **Hybridization.** The last strategy discussed in the review advocates the use of hybrid systems in order to benefit from the strength of each component and achieve more powerful systems. This perception is an interesting avenue for future research, in the sense that further value-added combinations can be investigated.

In the literature, we have seen how some techniques from the same strategy can be used in complementary to each other, like generative augmentations and basic transformations in DA, and meta-learning and TL in knowledgeable systems. However, works that study this kind of composition are still limited in both variety and depth. Furthermore, hybridization of techniques from different strategies is restrictively steered, in some way, towards almost one direction; combining DA with TL for DNN as an effective method for reducing overfitting, improving model performance, and quickly learning new tasks with limited dataset. Much research has been devoted into this vein [287–289], the aim is to develop practical software tools for systematical integration of DA and TL into deep learning workflows and helping engineers utilize the performance power of these techniques much faster and more easily. It is indeed the best we can hope to empower DNN and mitigate its limitations. However, we believe it is also healthy to explore the potential of other innovative combinations similar to neural-symbolic systems, that not only integrate the reviewed techniques but also call upon other domains such as evolutionary approaches, statistical models, and cognitive reasoning. In this sense, multi-disciplinary studies like this paper are needed to build links between backgrounds and domains that are studied separately and to bring closer their bodies of research that are moving in different directions. Here, we intuitively and seamlessly bridged between the different strategies by considering, for instance, that FSL problem can be viewed as a semi-supervised learning problem with few available labeled data. Its aim is to transfer the knowledge of learning (e.g., meta-learning) from the source tasks to the target ones. Domain adaptation, which is a particular way of transfer learning is also a useful technique of data augmentation. We believe that making connections and enabling hybridization is a rich, under-explored area for future research that could help to converge to one unified solution. A general, adaptable, data-resistant system that will perform well in domains where ample data is available but also in data-scarce domains. It's far from obvious how to combine all the pieces and to explore others to conceive such custom systems that work on both settings, but researchers have to shift their attention towards this goal in order to fill the gap in thinking of how to build robust AI.

- **Evaluation.** There are very few studies in the literature that compare empirically the performance of techniques of the same strategy, and even fewer techniques of different strategies.

Semi-supervised and unsupervised methods are often evaluated based on their performances on downstream tasks by using datasets such as CIFAR-10, ImageNet, Places, and Pascal VOC. CIFAR-10 and SVHN are popular choices for evaluating the performances of semi-supervised models by training them with all unlabeled data and various amount of labeled examples. To provide a realistic evaluation, it is important to establish more high-quality baselines to allow for proper assessment of the added value of the unlabeled data. Researchers should thus evaluate their algorithms on a diverse suite of data sets with different quantities of unlabeled data and report how performance varies with the amount of unlabeled data. Oliver et al. [119] compared several SSNN on two image classification problems. They reported substantial performance improvements for most of the algorithms, and observed that the error rates declined as more unlabeled data points were added. These results are interesting in the sense that they indicate that, in image classification tasks, unlabeled data used by ANN can drive consistent improvement in performance. Likewise, it would be interesting to explore more empirical evaluations to draw more promising results that will guide research for better unlabeled data-based learners.

As for DA and knowledgeable systems, more theory and formalisms are needed in order to accurately compare and fairly evaluate techniques from these strategies. Indeed, despite the rapid progress of practical DA techniques, precisely understanding their benefits remains ambiguous. There is no common theoretical understanding regarding how training on augmented data affects the learning process, the parameters, and the overall performance. This is exacerbated by the fact that DA is performed in diverse ways in modern ML pipelines, for different tasks and domains, thus precluding a general theoretical framework. Hence, more theoretical insights are expected to theoretically characterize and understand the effect of various data augmentations used in practice in order to be able to evaluate their benefits. On the other hand, knowledgeable systems research community still does not have a good understanding of what the knowledge is in general, how to represent knowledge, and how to use knowledge in learning effectively. A unified theory of knowledge and the related issues is urgently needed in order to compare between knowledgeable systems and to measure how they optimize data requirement.

Certainly, enriching the evaluation baselines of each strategy is an important research avenue to pursue. However, the ultimate goal would be to develop approaches to evaluate in an abstract level, that is to be able to evaluate an altered data-hungry system by measuring how the alteration techniques, abstracting from their nature, have optimized the need for data, and by verifying the performance resistance against the change in the availability of data.

Automation. A common research question discussed in the reviewed strategies is automated design. Automatic generation of a DA schema for a given dataset or automatic learning of a transfer algorithm for a given domain or tasks, are examples of the projection of the general concept of Automated Machine Learning (AutoML) [290]. AutoML has recently emerged as a novel idea of automating the entire pipeline of learners' design by using ML to generate better ML. AutoML is advertised as a mean to democratize ML by allowing firms with limited data science expertise to easily build production-ready models in an automatic way, which will accelerate processes, reduce errors and costs, and provide more accurate results, as it enables businesses to select the best-performing algorithm. Practically, AutoML automates some or all steps of a standard ML pipeline that includes data preparation, feature engineering, model generation, and model evaluation [290]. Hence, one of the missions of autoML is to automatically manage data quality and quantity in the first step of the pipeline. Currently, autoML services rely only on data searching [291] and data simulator [292] to deal with effective data acquiring, we expect however that advances in autoML will deeply revolutionize the way we deal with data needs in ML pipeline.

Furthermore, it is worth to highlight the strong interaction between autoML and the reviewed techniques. As discussed before autoML as a general concept can also be instantiated for DA [172–175] and TL [197, 198] solutions that can also be packaged in an end-to-end automatic process. On the other way around, DA, TL, and other techniques are very useful for autoML tools. In the data preparation step, DA can be regarded as a tool for data collection and as a regularizer to avoid overfitting. In model generation step, as auoML become most popular for the design of deep learning architectures, neural architecture search (NAS) techniques [293] which target at searching for good deep network architectures that suit the learning problem are mostly used in this step. However, this method has a high computational cost, to address this, TL can use knowledge from prior tasks to speed up network design. In this vein, Wong et al. [294] proposed an approach that reduces the computational cost of Neural AutoML by using transfer learning. They showed a large reduction in convergence time across many datasets. Existing AutoML algorithms focus only on solving a specific task on some fixed datasets. However, a targeted high-quality AutoML system should have the capability of lifelong learning. Pasunuru et al. [295], introduced a continual architecture search (CAS) approach enabling lifelong learning. In addition, as the core idea of auoML is to learn to learn, it is natural to find a growing body of research that combines meta-learning and autoML, particularly for NAS improvement [296, 297]. AutoML has also been studied in few-shot learning scenarios, for instance, Elsken et al. [297] applied NAS to few-shot learning to overcome the data scarcity, while they only search for the most promising architecture and optimize it to work on multiple few-shot learning tasks. Recently, the idea of unsupervised autoML has begun to be explored, Liu et al. [298] proposed a general problem setup, namely unsupervised neural architecture search (UnNAS), to explore whether labels are necessary for NAS. They experimentally demonstrated that the architectures searched without labels are competitive with those searched with labels.

- **Humanization.** At the root of every intelligent system is the dream of building machines that learn and think like people. Naturally, all attempts to cure data hunger behavior of ML models stem from mimicking the mechanism of human. Currently, humans still retain a clear advantage in terms of sample efficiency of learning. Hence, an obvious research path to keep pursuing is to explore more human-inspired theories and human-like techniques.

We contend that the quest for non-data hungry learning may profit from the rich heritage of problem descriptions, theories, and experimental tools developed by cognitive psychologists. Cognitive psychologists promote a picture of learning that highlights the importance of early inductive biases, including core concepts such as number, space, and objects, as well as powerful learning algorithms that rely on prior knowledge to extract knowledge from small amounts of training data. Studies and insights drawn from cognitive and psychology can then potentially help to examine and understand mechanisms underlying human learning strengths. After all, FSL has been modeled after children's remarkable cognitive processes to generalize a new concept from a small number of examples. According to developmental psychologists, humans fast learning is hugely reliant on cognitive biases, Shinohara et al. [299] suggested symmetric bias and mutually exclusive bias as the two most promising cognitive biases that can be effectively employed in ML tasks. Following this line of thought, many advances might come from exploring other cognitive abilities, an interesting avenue might be the study of the commonsense knowledge, how it develops, how it is represented, how it is cumulated, and how it is used in learning. A related study would be to explore intuitive learning theories of physical and social domains. Children at early age have primitive knowledge of physics and social rules, whether learned or innate, it is an intriguing area of research to investigate the prospects for embedding or acquiring this kind of intuitive knowledge in machines, and to study how this could help capturing more human-like learning-to-learn dynamics that enable much stronger transfer to new tasks and new problems, and thus, accelerate the learning of new tasks from very limited amounts of experience and data.

Conclusions

This paper provided a comprehensive survey on the current progress regarding data-efficiency in ML, a promising area in AI that has been attracting prominent research attention in recent years. Understanding the data-efficiency issue from different perspectives helped to categorize typical methods along four lines of research, according to how they solve the issue, namely, by using non-supervised algorithms, data augmentation, shared knowledge, or hybrid systems. In each category, advances and challenges were thoroughly discussed and some summaries and insights were presented.

The key findings motivate the need for more value-added synergy between existing data-efficient methods in order to build more robust systems. The automated ML design was also identified as an important avenue for optimizing the way AI is using massive data in the ML pipeline. Furthermore, the results also suggest the necessity of drawing more insight from cognitive science and behavioral studies to achieve data-efficient human-like learning.

Acknowledgements
Not applicable.

Authors' contributions
The author was in charge of conducting the study and elaborating the manuscript. The author read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The author declare that she has no competing interests.

Received: 12 November 2020 Accepted: 16 January 2021
Published online: 26 January 2021

References

1. Silver D, Huang A, Maddison C, Guez AJ, et al. Mastering the game of Go with deep neural networks and tree search. Nature. 2016;529(7587):484. .
2. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. p. 1063–6919.

3. Adiwardana D, Luong M, David R, et al. Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977(2020). 2020.
4. Marcus G. Deep learning: a critical appraisal. arXiv preprint arXiv:1801.00631 , 2018.
5. Ford M. Architects of Intelligence: the Truth About AI From the People Building It. Kindle. Birmingham: Packt Publishing; 2018.
6. Shu J, Xu Z, Meng D. Small sample learning in big data era. arXiv preprint arXiv:1808.04572, 2018.
7. Wang Y, Yao Q, Kwok JT, Ni LM. Generalizing from a few examples: A survey on few-shot learning. ACM Comput Surv. 2020;53(3):1–34.
8. Qi G, Luo J. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. arXiv preprint arXiv:1903.11260. 2019.
9. Kitchin R. The data revolution: big data, open data, data infrastructures and their consequences. Thousand Oaks: SAGE Publications Ltd; 2014.
10. Drucker J. Humanities approaches to graphical display. Digital Humanities Quarterly. http://www.digitalhumanities.org/dhq/vol/5/1/000091/000091.html. 2011.
11. Thomas W. Algorithms. From Al-Khwarizmi to Turing and Beyond. In Turing's Revolution. Birkhäuser, Cham. https://doi.org/10.1007/978-3-319-22156-4_2,2015.
12. Knuth DE. Ancient Babylonian algorithms. Commun ACM. 1972;15(7):671–7.
13. Chabert J. A History of Algorithms: From the Pebble to the Microchip. Berlin: Springer; 1999.
14. Paz EB, Ceccarelli M, Otero JE, Sanz JLM. Machinery during the industrial revolution. Dordrecht: Springer; 2009.
15. Asperti A, Ricciotti W. Formalizing Turing Machines. Logic, Language, Information and Computation. WoLLIC 2012. Lecture Notes in Computer Science, Vol. 7456. Springer, Berlin. 2012.
16. Navathe SB. Evolution of data modeling for databases. Commun ACM. 1992;35(9):112–23.
17. Mitchell JC. Concepts in programming languages. Cambridge: Cambridge Cambridge University Press; 2002.
18. Waldrop MM. The chips are down for Moore's law. Nature. 2016;530:7589. p. 144–7.
19. Gandomi A, Haider M. Beyond the hype: Big data concepts, methods, and analytics. Int J Inf Manage. 2015;35(2):137–44.
20. Batra S. Big data analytics and its reflections on DIKW hierarchy. Rev Manag. 2014;4(1):5–17.
21. Turing AM. Computing machinery and intelligence. Mind. 1950;59(236):433–60.
22. Lighthill J. Artificial intelligence: A general survey. Artificial intelligence: A Paper Symposium. Science Research Council. 1973.
23. Krizhevsky A, Sutskever I, Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25 (NIPS 2012). 2012.p. 1097–1105.
24. Landauer R. The physical nature of information. Phys Lett A. 1996;217:188–93.
25. Glattfelder JB. A Universe Built of Information. The Frontiers Collection. Springer. 2019. p. 473–514.
26. Davies P. Universe from Bit. In Information and the Nature of Reality: From Physics to Metaphysics. Cambridge: Cambridge University Press; 2014. p. 83–117.
27. Wheeler J, Information, Physics, Quantum: The Search for Links. In Proceedings III International Symposium on Foundations of Quantum Mechanics. 1990. p. 354–358.
28. Lloyd S. The computational universe. In Information and the Nature of Reality: from Physics to Metaphysics. Cambridge: Cambridge University Press; 2014. p. 118–33.
29. Cohen S. M. Aristotle's metaphysics. Stanford Encyclopedia of Philosophy. 2000.
30. Tang H. Weiwei Huang. Brain Inspired Cognitive System for Learning and Memory. Neural Information Processing. ICONIP 2011. Lecture Notes in Computer Science, vol 7062. Springer. 2011, 477-484.
31. Kurzweil R. How to Create a Mind: The Secret of Human Thought. Viking. ISBN 978-067002529-9. 2012.
32. Wang Y, Lu J, Gavrilova M, Fiorini R, Kacprzyk J. 2018. Brain-Inspired Systems (BIS): Cognitive Foundations and Applications. IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2018. p. 995–1000.
33. Chen J, Chen J. Zhang R, Hu X. Towards Brain-inspired System: Deep Recurrent Reinforcement Learning for Simulated Self-driving Agent. arXiv preprint arXiv:1903.12517. 2019.
34. Molina D, Poyatos J, Del Ser J, Garc S, Hussain A, Herrera F. Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. Cognitive Computation. 2020. p. 1–43.
35. Del Ser J, Osaba E, et al. Bio-inspired computation: Where we stand and what's next. Swarm Evolutionary Computation. 2019;48:220–50.
36. Zang H, Zhang S, Hapeshi K. A Review of Nature-Inspired Algorithms. J Bionic Eng. 2010;7:232–7.
37. Sorensen K. Metaheuristics - the Metaphor Exposed.International Transactions in Operational Research. 2013;22:3.p. 3–18.
38. Mitchell TM. Machine Learning. McGraw-Hill. ISBN978-007115467-3$4 1997.
39. Kearns MJ, Vazirani U. An introduction to computational learning theory. MIT Press. ISBN 978-026211193-5. 1994.
40. Valiant LG. A theory of the learnable. Commun ACM. 1984;27(11):1134–42.
41. Gibbons C, Richards S, Valderas JM, Campbell J. Supervised Machine Learning Algorithms Can Classify Open-Text Feedback of Doctor Performance With Human-Level Accuracy. J Med Internet Res. 2017;19:3. e65.
42. Cho J, Lee K, Shin E, Choy G, Do S. 2017. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?. arXiv preprint ararXiv:1511.06348. 2016.
43. Mukherjee S, Tamayo P, Rogers S. Estimating Dataset Size Requirements for Classifying DNA Microarray Data. J Comput Biol. 2003;10:2. p. 119–142.
44. Forman G, Cohen I. Learning from little: Comparison of classifiers given little training. Knowledge Discovery in Databases PKDD 2004. Lecture Notes in Computer Science Springer. 2004; 3202. p. 161–172.
45. Ciuriak D. The Economics of Data: Implications for the Data-Driven Economy. Chapter 2 in "Data Governance in the Digital Age". Centre for International Governance Innovation. 2018.
46. Research and Markets. Big Data Market by Component, Deployment Mode, Organization Size, Business Function (Operations, Finance, and Marketing and Sales), Industry Vertical (BFSI, Manufacturing, and Healthcare and Life

Sciences), and Region - Global Forecast to 2025. https://www.marketsandmarkets.com/Market-Reports/big-data-market-1068.html.2020.

47. NewVantage Partners. Data-Driven Business Transformation Connecting Data/AI Investment to Business Outcomes. http://newvantage.com/wp-content/uploads/2020/01/NewVantage-Partners-Big-Data-and-AI-Executive-Survey-2020-1.pdf. 2020.
48. Niebel T, Rasel F, Viete S. BIG data – BIG gains? Understanding the link between big data analytics and innovation. Econ Innov New Technol. 2019;28(3):296–316.
49. Xie K, Wu Y, Xiao J, Hu Q. Value co-creation between firms and customers: The role of big data-based cooperative assets. Inf Manag. 2016;53(8):1038–48.
50. Brynjolfsson E, McElheran K. The Rapid Adoption of Data-Driven Decision-Making. American Economic Review. 2016;106(9):39–133.
51. Brynjolfsson E, Hitt LM, Kim HH. Strength in numbers: how does data-driven decision-making affect firm performance. MIT Sloan Working Paper, Cambridge. Available at SSRN: https://ssrn.com/abstract=1819486.
52. Andrew M, Brynjolfsson E. Big data: the management revolution. Harvard Bus Rev. 2012;90(10):60–8.
53. Domingos P. A few useful things to know about machine learning. Commun ACM. 2012;55(10):77–87.
54. Dimensional Research. Artificial Intelligence and Machine Learning Projects Are Obstructed by Data Issues. https://cdn2.hubspot.net/hubfs/3971219/Survey%20Assets%201905/Dimensional%20Research%20Machine%20Learning%20PPT%20Report%20FINAL.pdf. 2019.
55. Zhou ZH. &nbsp;A brief introduction to weakly supervised learning. Natl Sci Rev. 2018;5:1.
56. Willemink MJ, Koszek WA, Hardell C, et al. Preparing medical imaging data for machine learning. Radiology. 2020;295(1):4–15.
57. Grünberg K, Jakab A, Langs G, et al. Ethical and Privacy Aspects of Using Medical Image Data. In Cloud-Based Benchmarking of Medical Image Analysis. Springer.2017. p. 33–43.
58. Zarsky T. Incompatible. The GDPR in the Age of Big Data. Seton Hall Law Review. 2017;47:4.
59. Mouret JB, Micro-Data, Learning. The Other End of the Spectrum. arXiv preprint arXiv: 1610.00946. 2016.
60. Ruparel NH, Shahane NM, Bhamare DP Learning from Small Data Set to Build Classification Model: A Survey. Proceedings on International Conference on Recent Trends in Engineering and Technology. 2013.
61. Halevy A, Norvig P, Pereira F. The Unreasonable Effectiveness of Data. IEEE Intell Syst. 2009;24(2):8–12.
62. Banko M, Brill E. Scaling to very very large corpora for natural language disambiguation. Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. 2001. p. 26–33.
63. Zhu X, Vondrick C, Ramanan D, Fowlkes C. Do We Need More Training Data or Better Models for Object Detection? In the British Machine Vision Conference, BMVC 2016.
64. Greco C, Polonioli A, Tagliabue J. Less (Data) Is More: Why Small Data Holds the Key to the Future of Artificial Intelligence. arXiv preprint arXiv:1907.10424.2019.
65. Liebert W, Schmidt JC. Collingridge's dilemma and technoscience. Poiesis Prax. 2010;7:55–71.
66. Mastorakis G. Human-like machine learning: limitations and Suggestions. arXivpreprint arXiv:1811.06052 . 2018.
67. Wolff JG. The SP Theory of Intelligence: Distinctive Features. IEEE Access. 2015; 4. p. 216–246.
68. Zador AM. A critique of pure learning and what artificial neural networks can learn from animal brains. Nat Commun. 2019;10(3770):1–7.
69. Marcus G. Innateness, AlphaZero, and Artificial Intelligence. arXiv preprint arXiv:1801.05667. 2018.
70. Marcus G. The next decade in AI: four steps towards robust artificial intelligence. arXiv arXiv:2002.06177. 2020.
71. van Engelen JE. Hoos. H. A survey on semi-supervised learning. Mach Learn. 2020;109(2):373–440.
72. Qin Y, Ding S, Wang L, Wang Y. 2019. Cognitive Computation. 2020; 11:5. p. 599–612.
73. Kostopoulos G, Karlos S, Kotsiantis S, Ragos O. Semi-supervised regression: A recent review. J Intell Fuzzy Syst. 2018;35:2. p. 1483–1500.
74. Kim K. An improved semi-supervised dimensionality reduction using feature weighting: Application to sentiment analysis. Expert Systems with Applications. 2018;109:49–65.
75. Triguero I, Garcia S, Herrera F. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowledge Information systems. 2015;42:2. p. 245–284.
76. Tanha J, van Someren M, Afsarmanesh H. Semi-supervised self-training for decision tree classifiers. Int J Mach Learn Cybern. 2017;8:1. p. 355–370.
77. Livieris IE, Kanavos A, Tampakas V, Pintelas P. An auto-adjustable semi-supervised self-training algorithm. Algorithm. 2018;11:9.
78. Li M, Zhou ZH Self-training with editing. In Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2005. 611–621.
79. Zhu X, Goldberg A. Introduction to semi-supervised learning. Synth Lect Artif Intell Mach Learn. 2009;3(1):1–30.
80. Xu C, Tao D, Xu C. A survey on multi-view learning. arXiv preprint arXiv:1304.5634 (2013).
81. Didaci L, Fumera G, Roli F. Analysis of co-training algorithm with very small training sets. In Structural, Syntactic, and Statistical Pattern Recognition, vol. 7626 of Lecture Notes in Computer Science. 2012. P.719–726.
82. Jiang Z, Zhang S, Zeng J. A hybrid generative/discriminative method for semi-supervised classification. Knowl Based Syst. 2013;37:137–45.
83. Qiao S, Shen W, Zhang Z, Wang B, Yuille A. Deep co-training for semi-supervised image recognition. In Computer Vision – ECCV 2018. Lecture Notes in Computer Science. 2018; 11219.
84. Chonga Y, Dinga Y, Yanb Q, Pana S. Graph-based semi-supervised learning: A review. Neurocomputing. 2020;408:216–30.
85. Pang B, Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In the 42nd annual meeting on association for computational linguistics, association for computational linguistics. 2004.
86. Getz G, Shental N, Domany E. Semi-supervised learning—A statistical physics approach. In Proceedings of the 22nd ICML workshop on learning with partially classified training data. 2005.

87. Wu X, Li Z, So AM, Wright J, Chang S. Learning with partially absorbing randomwalks. In Advances in neural information processing systems. 2012. p. 3077–3085.
88. Blum A, Chawla S. Learning from labeled and unlabeled data using graph mincuts. In Proceedings of the 18th international conference on machine learning. 2001.p. 19–26.
89. Dornaika F, Dahbi R, Bosaghzadeh A, Ruichek Y. Efficient dynamic graph construction for inductive semi-supervised learning. Neural Netw. 2017;94:192–203.
90. Liu W, Wang J, Chang SF Robust and scalable graph-based semi-supervised learning. Proceedings of the IEEE, 2012; 100:9, p. 2624–2638.
91. Bennett KP, Demiriz A. Semi-supervised support vector machines. In: Advances in neural information processing systems. 1999; 11.p. 368–374.
92. Joachims T. Transductive inference for text classification using support vector machines. In: Proceedings of the sixteenth international conference. 1999; 99, p. 200–209.
93. Melacci S, Belkin M. Laplacian support vector machines trained in the primal. J Mach Learn Res. 2011;12:1149–84.
94. Li Y, Kwok JT, Zhou Z. Semi-supervised learning using label mean. In the 26th international conference on machine learning (ICML 2009). 2009.p. 633–640.
95. Li T, Wang XL. Semi-supervised SVM classification method based on cluster kernel. Appl Res Comput. 2013;30:1.p. 42–45.
96. Ding S, Zhu Z, Zhang X. An overview on semi-supervised support vector machine. Neural Comput Appl. 2015;28:5. p. 969–978.
97. Ouali Y, Hudelot C, Tami M. An Overview of Deep Semi-Supervised Learning. arXiv preprint arXiv:2006.05278 (2020).
98. Rasmus A, Berglund M, Honkala M, Valpola H, Raiko T. Semi-supervised learning with ladder networks. In Advances in neural information processing systems. 2015.p. 3546–3554.
99. Prémont-Schwarz I, Ilin A, Ha TH, Rasmus A, Boney R, Valpola H. Recurrent ladder networks. In Advances in neural information processing systems. 2017.p. 6009–6019.
100. Miyato T, Maeda S, Ishii S, Koyama M. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans Pattern Anal Mach Intell. 2018;41:8. p. 1979–1993.
101. Laine S, Aila T. Temporal ensembling for semi-supervised learning. In 5th International Conference on Learning Representations (ICLR 2017). 2017.
102. Tarvainen A, Valpola H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Advances in Neural Information Processing Systems (NIPS 2017). 2017.
103. Hinton GE, Krizhevsky A, Wang SD Transforming autoencoders. in International Conference on Artificial Neural Networks. Springer. 2011.p. 44–51.
104. Cohen T. Welling M. Group equivariant convolutional networks. In International conference on machine learning. 2016. p. 2990–2999.
105. Zhang L, Qi JG, Wang L, Luo J. AET vs. AED: Unsupervised representation learning by auto-encoding transformations rather than data. arXiv preprint arXiv:1901.04596(2019).
106. Qi GJ, Zhang L, Chen CW, Tian Q. AVT: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. arXiv preprint arXiv: 1903.10863(2019).
107. Jing L, Tian Y. 2019. Self-supervised visual featurelearning with deep neural networks: A survey. arXivpreprint arXiv:1902.06162(2019).
108. Oord Avd, Kalchbrenner N, Kavukcuoglu K. Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759(2016).
109. Oord Avd, Kalchbrenner N, Espeholt L, Vinyals O, Graves A, et al. Conditional image generation with pixelCNN Decoders. in Advances in Neural Information Processing Systems (NIPS 2016).2016. p. 4790–4798.
110. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A. N, Kaiser Ł, Polosukhin I. Attention is all you need. In Advances in Neural Information Processing Systems (NIPS 2017). 2017.p. 5998–6008.
111. Sutton RS. Barto AG reinforcement learning: an introduction. Cambridge: MIT Press; 2018.
112. Arulkumaran K, Deisenroth MP, Brundage M, Bharath A. A brief survey of deep reinforcement learning. arXiv preprint arXiv:1708.05866(2017).
113. Rummery GA, Niranjan M. On-line Q-learning using Connectionist Systems. Cambridge: University of Cambridge, Department of Engineering; 1994.
114. Watkins CJCH, Dayan P. Q-Learning. Machine Learning. 1992;8(3):279–92.
115. Henderson P, Islam R, Bachman P, Pineau J, et al. 2018. Deep reinforcement learning that matters. In the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).
116. Li Y. Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274 (2017).
117. Botvinick M, Ritter S, Wang JX, Kurth-Nelson Z. Reinforcement learning, fast and slow. Trends Cogn Sci. 2017;23:5.
118. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020).
119. Oliver A, Odena A, Raffel C, Cubuk ED, Goodfellow I. J. Realistic evaluation of deep semi-supervised learning algorithms. arXiv preprint arXiv:1804.09170 (2018).
120. Palacio-Nino JO, Berzal F. Evaluation metrics for unsupervised learning algorithms. arXiv preprint arXiv:1905.05667 (2019).
121. Thornton C, Hutter F, Hoos HH, Leyton-Brown K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In the 19th ACM international conference on knowledge discovery and data mining (KDD '13). 2013.p. 847–855.
122. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data. 2019;6:48–60.
123. Krizhevsky A, Sutskever I, Hinton G. E. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS 2012).
124. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 86:11. p. 2278–2324.

125. Ciresan DC, Meier U, Gambardella LM, Schmid-huber J. Deep big simple neural nets excel on digit recognition. Neural Comput. 2010;22(12):3207–20.
126. Yaeger LS, Lyon RF, Webb BJ. Effective training of a neural network character classifier for word recognition. In Advances in Neural Information Processing Systems (NIPS 1997). 1997.p. 807–816.
127. Inoue H. Data augmentation by pairing samples for images classification. arXiv preprint arXiv:1801.02929(2018).
128. Zhong Z, Zheng L, Kang G, Li S, Yang Y. Random Erasing Data Augmentation. arXiv preprint arXiv:1708.04896 (2017).
129. Mikołajczyk A, Grochowski M. Data augmentation for improving deep learning in image classification problem. International Interdisciplinary PhD Workshop (IIPhDW2018). 2012.p. 117–122.
130. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In International Conference on Learning Representations (ICLR 2015).
131. Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier GANs. arXiv preprint arXiv:1610.09585 (2016).
132. Antoniou A, Storkey A, Edwards H. Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017).
133. Mariani G, Scheidegger F, Istrate R, Bekas C, Malossi C. Bagan: Data augmentation with balancing GAN - arXiv preprint arXiv:1803.09655 (2018).
134. Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196(2017).
135. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. In the 4th International Conference on Learning Representations (ICLR 2016).
136. Isola P, Zhu J-Y, Zhou T, Efros AA Image-to-image translation with conditional adversarial networks. In the IEEE conference on computer vision and pattern recognition. 2017. p. 1125–1134.
137. Zhu J-Y, Park T, Isola P, Efros A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In the IEEE International Conference on Computer Vision. 2017.p. 2223–2232.
138. Kim T, Cha M, Kim H, Lee JK, Kim J. Learning to discover cross-domain relations with generative adversarial networks. In the 34th International Conference on Machine Learning (JMLR 2017). 2017.p. 1857–1865.
139. Yi Z, Zhang H, Tan P, Gong M. DUALGAN: Unsupervised dual learning for image-to-image translation. In the IEEE International Conference on Computer Vision. 2017. p. 2849–2857.
140. Liu MY,Tuzel O.Coupled generative adversarial networks. In Advances in Neural Information Processing Systems (NIPS2016). 2016.
141. Kingma D. P,Welling M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2014).
142. Cai L, Hongyang G, Ji S. Multi-stage variational auto-encoders for coarse-to-fine image generation. In the 2019 SIAM International Conference on Data Mining. 2019.
143. Leglaive S, Girin L, Horaud R. A variance modeling framework based on variational autoencoders for speech enhancement. In IEEE International Workshop on Machine Learning for Signal Process. 2018.
144. Esling P, Chemla-Romeu-Santos A, Bitton A. Generative timbre spaces with variational audio synthesis. In the Int. Conf. on Digital Audio Effects. 2018.
145. Salimans T, Goodfellow I, et al. Improved techniques for training GANs. arXiv preprint arXiv:1606.03498 (2016).
146. Wong SC, Gatt A, Stamatescu V, McDonnell M. D. Understanding data augmentation for classification: when to warp? In International Conference on Digital Image Computing: Techniques and Applications (DICTA). 2016.
147. DeVries T, Taylor G. W. Dataset augmentation in feature space. In the international conference on machine learning (ICML 2017).
148. Chawla NV, Bowyer KW, Hall L, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intellig Res. 2002;16:321–57.
149. He H, Bai Y, Garcia EA, Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. IEEE International Joint Conference on Neural Networks. 2008.p. 1322–1328.
150. Kumar V, Glaude H, de Lichy C, Campbell W. A Closer Look At Feature Space Data Augmentation For Few-Shot Intent Classification. arXiv preprint arXiv:1910.04176(2019).
151. Quinonero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND. Covariate shift by kernel mean matching. Dataset Shift in Machine Learning. Cambridge: MIT Press; 2009. p. 131–60.
152. Saenko K, Kulis B, Fritz M, Darrell T. Adapting visual category models to new domains. In the european conference on Computer Vision (ECCV2010).
153. Csurka G. Domain adaptation for visual applications: A comprehensive survey. arXiv preprint arXiv:1702.05374 (2017).
154. Gopalan R, Li R, Chellappa R. Domain adaptation for object recognition: An unsupervised approach. In International Conference on Computer Vision (ICCV 2011). 2011.p. 999–1006.
155. Ganin Y, Lempitsky V. Unsupervised domain adaptation by back propagation. arXivpreprint arXiv:1409.7495(2014).
156. Ghifary M, Kleijn WB, Zhang M, Balduzzi D, Li W. Dee preconstruction classification networks for unsupervised domain adaptation. In European Conference on Computer Vision. 2016.p. 597–613.
157. Bousmalis K, Silberman N, Dohan D, Erhan D, Krishnan D. Unsupervised pixel-level domain adaptation with generative adversarial networks. In the IEEE conference on computer vision and pattern recognition. 2017.p. 3722–3731.
158. Sun Y, Tzeng E, Darrell T, Efros AA. Unsupervised Domain Adaptation through Self-Supervision. arXiv preprint arXiv:1909.11825 (2019).
159. Gatys LA, Ecker AS, Bethge M. Image style transfer using convolutional neural networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016). 2016.p. 2414–2423.
160. Hao W, Zhang Z, Guan H. CMCGAN: A uniform framework for cross-modal visual-audio mutual generation. In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
161. Xu T, Zhang P, Huang Q, Zhang H, Gan Z, Huang X, He X. 2018. ATTNGAN: Fine-grained text to image generation with attentional generative adversarial networks. In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018).

162. Gibiansky A, Arik S, Diamos G, et al. Deep voice 2: Multi-speaker neural text-to-speech. In Conference on Neural Information Processing Systems (NIPS 2017). 2017.p. 2966–2974.
163. Wang Y, Wu C, Herranz L, et al. Transferring GANs: generating images from limited data. In the European Conference on Computer Vision (ECCV 2018). 2018.p. 220–236.
164. Yamaguchi S, Kanai S, Eda T. Effective Data Augmentation with Multi-Domain Learning GANs. arXiv preprint arXiv:1912.11597 (2019).
165. Huang S, Lin A, Chen SP, et al. Aug-GAN: Cross domain adaptation with GAN-based data. In the European Conference on Computer Vision (ECCV 2018). 2018.p. 731–744.
166. Raille G, Djambazovska S, Musat C. Fast Cross-domain Data Augmentation through Neural Sentence Editing. arXiv preprint arXiv: 2003.10254 (2020).
167. Xie Q, Dai Z, Hovy E, Luong M, Le Q. V. Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848(2019).
168. Lin L, Wang K, Meng D, Zuo W, Zhang L. Active self-paced learning for cost-effective and progressive face identification. IEEE Trans Pattern Anal Mach Intell. 2018;40(1):7–19.
169. Ratner A, Bach SH, Ehrenberg H, et al. Snorkel: Rapid training data creation with weak supervision. VLDB J. 2017;11(3):709–30.
170. Shijie J, Ping W, Peiyi J, Siping H. Research on data augmentation for image classification based on convolution neural networks. In 2017 Chinese automation congress (CAC). 2017.p. 4165–70.
171. Wang C, Macnaught G, Papanastasiou G, et al. Unsupervised Learning for Cross-Domain Medical Image Synthesis Using Deformation Invariant Cycle Consistency Networks. In international Workshop on Simulation and Synthesis in Medical Imaging (SASHIMI 2018). 2018.p. 52–60.
172. Ratner AJ, Ehrenberg HR, Hussain Z, et al. Learning to Compose Domain-Specific Transformations for Data Augmentation. arXiv preprint arXiv:1709.01643(2017).
173. Cubuk ED, Zoph B, Mane D, et al. AutoAugment: Learning Augmentation Policies from Data. arXiv preprint arXiv:1805.09501(2019).
174. Cubuk ED, Zoph B, Shlens J, Le QV, Randaugment: Practical automated data augmentation with a reduced search space. IEEE F Conference on Computer Vision and Pattern Recognition. 2020.p. 2160–7516.
175. Zhang X, Wang Q, Zhang J, Zhong Z. Adversarial AutoAugment. arXiv preprint arXiv:1912.11188(2019).
176. Eaton-Rosen Z, Bragman F, Ourselin S, Cardoso M. J. Improving data augmentation for medical image segmentation. In International Conference on Medical Imaging with Deep Learning. 2018.
177. Frid-Adar M, Diamant I, Klang E, et al. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing. 2018;321:321–31.
178. Armanious K, Jiang C, Fischer M. MedGAN: Medical image translation using GANs. Comput Med Imaging Graph. 2020;79:101684.
179. Schluter J, Grill T. Exploring data augmentation for improved singing voice detection with neural networks. In International Society for Music Information Retrieval Conference (ISMIR). 2015.
180. Wei JW, Zou K. Eda. Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196(2019).
181. Wulfmeier M, Bewley A, Posner I. Addressing Appearance Change in Outdoor Robotics with Adversarial Domain Adaptation. In IEEE International Conference on Intelligent Robots and Systems. 2017.
182. Jialin Pan S, Yang Q. A survey on transfer learning. IEEE Transactions on knowledge data engineering. 2010;22:10. p. 1345–1359.
183. Weiss K, Khoshgoftaar T, M,Wang DD. A survey of transfer learning. J Big Data. 2016;3(1):1–40.
184. Rosenstein M, Marx Z, Kaelbling L. To transfer or not to transfer. In NIPS'05 Workshop, Inductive Transfer: 10 Years Later. 2005.
185. Liu B, Xiao Y, Hao Z. A Selective Multiple Instance Transfer Learning Method for Text Categorization Problems. Knowl-Based Syst. 2018;141:178–87.
186. Chen YS, Hsu CS, Lo CL. An Entire-and-Partial Feature Transfer Learning Approach for Detecting the Frequency of Pest Occurrence. IEEE Access. 2020; 8.p. 92490–92502.
187. Furfaro R, Linares R, Reddy V. Space objects classification via light-curve measurements: deep convolutional neural networks and model-based transfer learning. Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS). 2018.
188. Yang Z, Zhao J, Dhingra B, et al. Glomo: Unsupervisedly learned relational graphs as transferable representations. arXiv preprint arXiv:1806.05662(2018).
189. Yang Q, Zhang Y, Dai W, Pan S. Transfer Learning in Reinforcement Learning. In Transfer Learning (pp. 105–125). Cambridge: Cambridge University Press. doi:https://doi.org/10.1017/9781139061773.0102020.
190. Lia X, Grandvalet Y, Davoine F, et al. 2020. Transfer learning in computer vision tasks: Remember where you come from. Image Vision Comput. 2020; 93.
191. Malte A, Ratadiya P. Evolution of transfer learning in natural language processing. arXiv preprint arXiv:1910.07370 arXiv:(2019).
192. Wang L, Guo B, Yang Q. Smart City Development With Urban Transfer Learning. Computer. 2018;51(12):32–41.
193. Asgarian A, Sobhani P, Zhang JC. A hybrid instance-based transfer learning method. arXiv preprint arXiv:1812.01063 (2018).
194. Li H, Chaudhari P, Yang H. Rethinking the Hyperparameters for Fine-tuning. arXiv preprint arXiv:2002.11770(2020).
195. Yim J, Joo D, Bae J, Kim J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). 2017.p. 1063–6919.
196. Liang H, Fu W, Yi F. A Survey of Recent Advances in Transfer Learning. In IEEE 19th International Conference on Communication Technology (ICCT). 2576–7828. 2019.
197. Yang Q, Zhang Y, Dai W, Pan S. AutoTL: Learning to Transfer Automatically. In Transfer Learning pp (168–176). Cambridge: Cambridge University Press. doi:https://doi.org/10.1017/9781139061773.0142020.
198. Wei Y, Zhang Y, Yang Q. Learning to Transfer. arXiv preprint ivarX:1708.05629 (2017).

199. Caruana R. Multitask learning. Mach Learn. 1997;28:1. p. 41–75.
200. Olivas ES, Guerrero JDM, Martinez-Sober M, et al. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques. IGI Global. ISBN 9781605667669. 2009.
201. Lee HB, Yang E, Hwang S. J. Deep asymmetric multi-task feature learning. arXiv preprint arXiv:1708.00260(2017).
202. Zhang Y, Yang Q. A survey on multi-task learning. arXiv preprint arXiv:1707.08114(2017).
203. Zhang J. Multi-task feature selection with sparse regularization to extract common and task-specific features. Neurocomputing. 2019;340:76–89.
204. Liu P, Qiu X, Huang X. Adversarial multi-task learning for text classification. In the 55th Annual Meeting of the Association for Computational Linguistics (ACL). 2017.
205. Su Y, Li J, Qi H, Gamba P, Plaza A, Plaza J. Multi-Task Learning with Low-Rank Matrix Factorization for Hyperspectral Nonlinear Unmixing. In IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019).
206. Barzilai A, Crammer K. Convex multi-task learning by clustering. In the 18th International Conference on Artificial Intelligence and Statistics (AISTATS). 2015.
207. Long M, Cao Z, Wang J, Yu P. S. Learning multiple tasks with multilinear relationship networks. In Conference on Neural Information Processing Systems (NIPS 2017).
208. Bickel S, Bogojeska J, Lengauer T, Scheffer T. Multi-task learning for HIV therapy screening. In the 25th international conference on Machine learning. 2008, p. 56–63.
209. Yang P, Li P. Distributed Primal-Dual Optimization for Online Multi-Task Learning. In the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20). 2020.
210. Zhou Q, Chen Y, Pan SJ. Communication-efficient distributed multi-task learning with matrix sparsity regularization. Mach Learn. 2020;109:569–601.
211. Zhang C, Zhao P, Hao S, et al. Distributed multi-task classification: a decentralized online learning approach. Mach Learn. 2018;107:727–47.
212. Zhao Y, Tang F, Dong W, Huang F, Zhang X. Joint face alignment and segmentation via deep multi-task learning. Multimedia Tools Appl. 2019;78:13131–48.
213. Akhtar MS, Chauhan DS, Ekbal A. A Deep Multi-task Contextual Attention Framework for Multi-modal Affect Analysis. ACM Trans Knowl Discovery Data. 2020;14:3.p. 1–27.
214. Benton A, Mitchell M, Hovy D. Multitask learning for mental health conditions with limited social media data. In the 15th Conference of the European Chapter of the Association for Computational Linguistics. 2017.
215. Ruder S. An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017).
216. Parisi GI, Kemker R, Part JL, et al. Continual lifelong learning with neural networks: a review. Neural Netw. 2019;113:54–71.
217. Maltoni D, Lomonaco V. Continuous learning in single-incremental-task scenarios. Neural Netw. 2019;116:56–73.
218. Thrun S, Mitchell TM. Lifelong Robot Learning. In the Biology and Technology of Intelligent Autonomous Agents. 1995;144.
219. Thrun S. Is learning the n-th thing any easier than learning the first? In Conference on Neural Information Processing Systems (NIPS1996). 1996.p. 640–646.
220. Thrun S. Explanation-based Neural Network Learning: A Lifelong Learning Approach. The Kluwer International Series in Engineering and Computer Science book series (SECS). 1996; 357.
221. Silver DL, Mercer RE. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. Connect Sci. 1996;8:277–94.
222. Silver DL, Poirier R, Currie D. Inductive transfer with context-sensitive neural networks. Mach Learn. 2008;73(3):313–36.
223. Silver DL, Mason G, Eljabu L. Consolidation using sweep task rehearsal: Overcoming the stability-plasticity problem. In Advances in Artificial Intelligence, 2015; 9091. p. 307–322.
224. Chen Z, Ma N, Liu B. Lifelong learning for sentiment classification. In the 53rd Annual Meeting of the Association for Computational Linguistics (ACL). 2015.p. 750–756.
225. Ruvolo P, Eaton E. ELLA: an efficient lifelong learning algorithm. In the International Conference on Machine Learning. 2013. P.507–515.
226. Clingerman C, Eaton E. Lifelong learning with Gaussian processes. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. 2017.p. 690–704.
227. Chen Z, Liu B. Topic modeling using topics from many domains, lifelong learning and big data. In the 31st International Conference on Machine Learning (ICML 2014). 2014.p. 703–711.
228. Wang S, Chen Z, Liu B. Mining aspect-specific opinion using a holistic lifelong topic model. In the 25th International Conference on World Wide Web (WWW'16). 2016.p. 167–176.
229. Liu Q, Liu B, Zhang YL, Kim D, Gao Z. Improving opinion aspect extraction using semantic similarity and aspect associations. In the 30th AAAI Conference on Artificial Intelligence. 2016.
230. Mitchell T, Cohen W, Hruschka E, et al. Never-ending learning. Commun ACM. 2018;61(5):103–15.
231. Carlson A, Betteridge J, Wang RC, et al. Coupled semi-supervised learning for information extraction. In the third ACM international conference on Web search and data mining (WSDM'10). 2010.p. 101–110.
232. Bou Ammar H, Eaton E, Ruvolo P, Taylor M. Online multi-task learning for policy gradient methods. In: the 31st International Conference on Machine Learning. 2014.p. 1206–1214.
233. Tessler C, Givony S, Zahavy T, et al. A deep hierarchical approach to lifelong learning in minecraft. In the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17). 2017; 3. p. 1553–1561.
234. Rolnick D, Ahuja A, Schwarz J, Lillicrap T. Experience replay for continual learning. In advances in Neural Information Processing Systems (NIPS 2019). 2019.
235. Chen Z, Liu B. Lifelong Machine Learning. Morgan & Claypool publishers. ISBN 978-1627055017.2018.
236. Mazumder S, Ma N, Liu B. Towards a continuous knowledge learning engine for chatbots. arXiv preprint arXiv: 1802.06024 (2018).

237. Hospedales T, Antoniou A, Micaelli P. Meta-learning in neural networks: A survey. arXiv preprint arXiv: 2004.05439 (2020).
238. Mohammadi FG, Amini MH, Arabnia HR. An Introduction to Advanced Machine Learning: Meta-Learning Algorithms, Applications, and Promises. Optimization, Learning, and Control for Interdependent Complex Networks. 129–144. 2020.
239. Vanschoren J. Meta-learning: A survey. arXiv preprint arXiv:1810.03548(2018).
240. Xian Y, Lampert CH, Schiele B, Akata Z. Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly. IEEE Trans Pattern Anal Mach Intell. 2018;41(9):2251–65.
241. Bertinetto L, Henriques JF, Valmadre J, Torr P. A. Vedaldi. Learning feed-forward one-shot learners. In Advances in Neural Information Processing Systems (NIPS 2016). 2016. P.523–531.
242. Garcia V, Bruna J. Few-Shot Learning With Graph Neural Networks. arXiv preprint arXiv:1711.04043 (2018).
243. Kang B, Liu Z, Wang X, Yu F, Feng J, Darrell T. Few-shot Object Detection Via Feature Reweighting. In IEEE/CVF International Conference on Computer Vision (ICCV). 2019.
244. Dong N, Xing EP Few-Shot Semantic Segmentation with Prototype Learning. In the 29th British Machine Vision Conference (BMVC 2018).
245. Gui LY, Wang YX, Ramanan D, Moura J. Few-Shot Human Motion Prediction Via Meta-learning. In 15th European Conference Computer Vision (ECCV 2018). Lecture Notes in Computer Science. Springer International Publishing. ISBN 978-3-030-01236-6. 2018.
246. Kosh G, Zemel R, Salakhutdinov R. Siamese Neural Net-works For One-shot Image Recognition. In the 32nd International Conference on Machine Learning (ICML 2015). 2015; 37.
247. Vinyals O, Blundell C, Lillicrap T, Wierstra D, et al. Matching Networks For One Shot Learning. In Conference on Neural Information Processing Systems (NIPS 2016).
248. Snell J, Swersky K, Zemel R. 2017. Prototypical networks for few-shot learning. in Advances in Neural Information Processing Systems (NIPS 2017).
249. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, S TM Hospedales. Learning To Compare: Relation Network For Few-Shot Learning. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018.
250. Li W, Xu J, Huo J, Wang L, Gao Y, Luo J. Distribution consistency-based covariance metric networks for few-shot learning. In the 33th AAAI Conference on Artificial Intelligence (AAAI-19).
251. Wertheimer D, Hariharan B. Few-shot learning with localization in realistic settings. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.
252. Ravi S, Larochelle H. Optimization as a model for few-shot learning. In Proceedings of 5th International Conference on Learning Representations (ICLR 2017).
253. Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In the 34nd International Conference on Machine Learning (ICML 2017). 2017. p. 1126–1135.
254. Aiolli F. Transfer learning by kernel meta-learning. Workshop on Unsupervised and Transfer Learning. JMLR: Workshop and Conference Proceedings. 2012; 27. p. 81–95.
255. Eshratifar AE, Abrishami MS, et al. A meta-learning approach for custom model training. In the 33th AAAI Conference on Artificial Intelligence (AAAI-19). 2019.
256. Sun Q, Liu Y, Chua TS, Schiele B. Meta-transfer learning for few-shot learning. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.
257. Sun Q, Liu Y, Chen Z, et al. 2019. Meta-Transfer Learning through Hard Tasks. arXiv preprint arXiv:1910.03648 (2019).
258. Li XC, Zhan DC, Yang JQ, Shi Y, et al. Towards Understanding Transfer Learning Algorithms Using Meta Transfer Features. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2020). 2020.p. 855–866.
259. Bengio Y, Deleu T, Rahaman N. A meta-transfer objective for learning to disentangle causal mechanisms. arXiv preprint arXiv:1905.05667(2019).
260. Woong Soh J, Cho S, Ik Cho N. Meta-Transfer Learning for Zero-Shot Super-Resolution. arXiv preprint arXiv:2002.12213 (2020).
261. Indra Winata G, Cahyawijaya S, Lin Z. Meta-transfer learning for code-switched speech recognition. arXiv preprint arXiv:2004.14228 (2020).
262. Song W, Li S, Guo Y, et al. Meta Transfer Learning for Adaptive Vehicle Tracking in UAV Videos. In the international Conference on Multimedia Modeling. 2020.
263. Lin X, Baweja H, Kantor G, Held D. Adaptive Auxiliary Task Weighting For Reinforcement Learning. in Advances in Neural Information Processing Systems (NIPS 2019).
264. Franceschi L, Donini M, Frasconi P, Pontil M. Forward And Reverse Gradient-Based Hyperparameter Optimization, In the 34nd International Conference on Machine Learning (ICML 2014).
265. Epstein B, Meir R, Michaeli T. Joint autoencoders: a flexible meta-learning framework. In the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (PKDD 2018).
266. Chen J, Qiu X, Liu P, Huang X. Meta multi-task learning for sequence modeling. In 32nd AAAI Conference on Artificial Intelligence (AAAI-18). 2018.
267. Amit R, Meir R. Meta-learning by adjusting priors based on extended PAC-Bayes theory. In the international Conference on Machine Learning. 2018.p. 205–214.
268. Riemer M, Cases I, Ajemian R. Learning to learn without forgetting by maximizing transfer and minimizing interference. arXiv preprint arXiv:1810.11910arXiv:1810.11910 (2018).
269. Javed K, White M. Meta-learning representations for continual learning. In Advances in Neural Information Processing Systems (NIPS 2019).
270. He X, Sygnowski J, Galashov A, et al. 2019. Task agnostic continual learning via meta learning. arXiv preprint arXiv:1906.05201 (2019).
271. Munkhdalai T. Yu H. Meta Networks. arXiv preprint arXiv:1703.00837 (2017).
272. Vuorio R, Cho DY, Kim D, Kim J. Meta continual learning R Vuorio, Cho DY, Kim D, Kim J. arXiv preprint arXiv:1806.06928 (2018).

273. Xu H, Liu B, Shu L, Yu PS. Lifelong domain word embedding via meta-learning. arXiv preprint arXiv:1805.09991 (2018).
274. Wang H, Liu B, Wang S. Forward and Backward Knowledge Transfer for Sentiment Classification. arXiv preprint arXiv: 1906.03506 (2019).
275. Portelas R, Colas C, Weng L, et al. 2020. Automatic Curriculum Learning For Deep RL: A Short Survey. arXiv preprint arXiv: 2003.04664 (2020).
276. Domingos P. The master algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books. ISBN 978-046506570-7. 2015.
277. Han J, Choi D, Park S, et al. Hyperparameter optimization using a genetic algorithm considering verification time in a convolutional neural network. J Electr Eng Technol. 2020;15:721–6.
278. Choudhury SD, Pandey S, Mehrotra K. Deep Genetic Network. arXiv preprint arXiv:1811.01845(2018).
279. Garnelo M, Shanahan M. Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. Curr Opin Behav Sci. 2019;29:17–23.
280. Garcez AA, Gori M, Lamb LC. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. arXiv preprint arXiv:1905.06088 (2019).
281. Yi K, Wu J, Gan C. Neural-symbolic VQA. Disentangling reasoning from vision and language understanding. arXiv preprint arXiv: 1810.02338(2018).
282. Vedantam R, Desai K, Lee S. Probabilistic neural-symbolic models for interpretable visual question answering. arXiv preprint arXiv: 1902.07864 (2019).
283. Evans R, Grefenstette E. Learning explanatory rules from noisy data. J Artif Intell Res. 2018;61:1–64.
284. Tran S, Garcez A. Deep logic networks: Inserting and extracting knowledge from deep belief networks. IEEE T. Neur. Net. Learning Syst. 2018; 29.p. 246–258.
285. Silver DL. On Common Ground: Neural-Symbolic Integration and Lifelong Machine Learning. In the 9th Workshop on Neural-Symbolic Learning and Reasoning. 2013.
286. Hu Z, Ma X, Liu Z, Hovy E, Xing E. Harnessing deep neural networks with logic rules. In the 54th Annual Meeting of the Association for Computational Linguistics. 2018. p. 2410–2420.
287. Wolfe CR, Lundgaard K. T. Data Augmentation for Deep Transfer Learning, arXiv preprint arXiv:1912.00772 (2019).
288. Han D, Liu Q, Fan W. A new image classification method using CNN transfer learning and web data augmentation. Expert Syst Appl. 2018;95:43–56.
289. Milicevic M, Obradovic I, Zubrinic K. Data augmentation and transfer learning for limited dataset ship classification. WSEAS Trans Syst Control. 2018;13:460–5.
290. He X, Zhao K, Chu X. AutoML. A Survey of the State-of-the-Art. arXiv preprint arXiv:1908.00709(2019).
291. Yang J, Sun X, Lai YK, Zheng L, Cheng MM. Recognition from web data: a progressive Filtering approach. IEEE Trans Image Process. 2018;27(11):5303–15.
292. Ruiz N, Schulter S, Chandraker M.Learning to simulate. arXiv preprint arXiv:1810.02513 (2019).
293. Pham H, Guan MY, Zoph B.Efficient neural architecture search via parameter sharing. arXiv preprint arXiv :1802.03268 (2018).
294. Wong C, Houlsby N, Lu Y, Gesmundo A. Transfer learning with neural autoML. In Advances in neural information processing systems.
295. Pasunuru R, Bansal M. Continual and multi-task architecture search, arXiv preprint arXiv:1906.05226 (2019).
296. Kim J, Lee S, Kim S. Automated gradient based meta learner search, arXiv preprint arXiv:1806.06927(2018).
297. Elsken T, Staffer B, Metzen JH, Hutter F. Meta-learning of neural architectures for few-shot learning. In EEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2020). 2020.
298. Liu C, Dollár P, He K. Are labels necessary for neural architecture search?. arXiv preprint arXiv:2003.12056 (2020).
299. Shinohara S, Taguchi R, Katsurada K, Nitta T. A model of belief formation based on causality and application to n-armed bandit problem. T Jpn Soc AI. 2007;22:58–68.
300. Saunshi N, Plevrakis O, Arora S, Khodak M, Khandeparkar H, A Theoretical Analysis of Contrastive Unsupervised Representation Learning, Proceedings of the 36th International Conference on Machine Learning. 2019. p. 5628–5637.
301. Si J, Barto AG, Powell WB, Wunsch D. Reinforcement Learning and Its Relationship to Supervised Learning. in Handbook of Learning and Approximate Dynamic Programming, IEEE, 2004, p. 45–63, doi: https://doi.org/10.1109/9780470544785.ch2.
302. Kakade S, On the Sample Complexity of Reinforcement Learning, Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London. 2003.
303. Deisenroth MP, Rasmussen CE PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In Proceedings of the International Conference on Machine Learning, 2011.
304. Nguyen H, La H, Review of Deep Reinforcement Learning for Robot Manipulation. 2019 Third IEEE International Conference on Robotic Computing. (IRC), Naples, Italy, 2019, p. 590–595, doi: https://doi.org/10.1109/IRC.2019.00120.
305. Levine S, Koltun V. Guided policy search, in Intern. Conf. on Machine Learning, 2013, p. 1–9.
306. Buckman J, Hafner D, Tucker G, Brevdo E, Lee H.Sample-Efficient Reinforcement Learning with Stochastic Ensemble Value Expansion, Advances in Neural Information Processing Systems (NeurIPS 2018). 2018;31. p. 8224–8234.
307. Kamthe S, Deisenroth M. Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control. In Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, PMLR. 2018; 84. p. 1701–1710.
308. Popov I, Heess N, Lillicrap T, et al, Data-efficient Deep Reinforcement Learning for Dexterous Manipulation. arXiv preprint arXiv:1704.03073 (2017).
309. Schwarzer M, Anand A, Goel R. Data-Efficient Reinforcement Learning with Self-Predictive Representations. arXiv preprint arXiv:2007.05929 (2020).
310. Arowolo MO, Adebiyi MO, Adebiyi AA, et al. A hybrid heuristic dimensionality reduction methods for classifying malaria vector gene expression data. IEEE Access. 2020;8:182422–30.

311. Arowolo MO, Isiaka RM, Abdulsalam SO, et al. A comparative analysis of feature extraction methods for classifying colon cancer microarray data. EAI endorsed transactions on scalable information systems. 2017;4:14.
312. Milidiú RL, Müller LF. SeismoFlow -- Data augmentation for the class imbalance problem, arXiv:2007.12229 (2020).
313. Shamsolmoali P, Zareapoor M, Shen L, et al., Imbalanced data learning by minority class augmentation using capsule adversarial networks, Neurocomputing, 2020.
314. Lee H, Park M, Kim J. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In: 2016 IEEE international conference on image processing (ICIP). 2016. p. 3713–7. https://doi.org/10.1109/ICIP.2016.7533053.
315. Finn C, Xu K, Levine S. Probabilistic model-agnostic meta-learning. Advances in Neural Information Processing Systems (NeurIPS 2018). 2018;31. p. 9516–9527.
316. Grant E, Finn C, Levine S, Darrell T, Griffiths T. Recasting gradient-based meta-learning as hierarchical bayes. InICLR, 2018.
317. Rusu AA, Rao D, Sygnowski J, Vinyals O, Pascanu R, Osindero S, Hadsell R. Meta-learning with latent embedding optimization. InICLR, 2019.
318. Vuorio R, Sun SH, Hu H, Lim JJ. Multimodal model-agnostic meta-learning via task-aware modulation. Advances in Neural Information Processing Systems (NeurIPS 2019). 2019;32. p. 1–12.
319. Andrychowicz M, Denil M, Colmenarejo SG, et al. Learning To Learn By Gradient Descent By Gradient Descent. Advances in Neural Information Processing Systems (NeurIPS 2016). 2016;29.
320. Ravi S, Larochelle H. Optimization As A Model For Few-Shot Learning. inICLR, 2016.
321. Wichrowska O, Maheswaranathan N, Hoffman M. W, et al. Learned Optimizers That Scale And Generalize. inICML, 2017.

Publisher's note

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