

Perspective



Cite this article: Chawla S *et al.* 2023 Ten years after ImageNet: a 360° perspective on artificial intelligence. *R. Soc. Open Sci.* **10**: 221414.
<https://doi.org/10.1098/rsos.221414>

Received: 6 November 2022

Accepted: 9 March 2023

Subject Category:

Computer science and artificial intelligence

Subject Areas:

artificial intelligence

Keywords:

ImageNet, supervised learning, artificial intelligence winter, Big Tech, transformers

Author for correspondence:

Sanjay Chawla

e-mail: schawla@hbku.edu.qa

Ten years after ImageNet: a 360° perspective on artificial intelligence

Sanjay Chawla¹, Preslav Nakov², Ahmed Ali¹,
Wendy Hall³, Issa Khalil¹, Xiaosong Ma¹, Husrev Taha
Sencar¹, Ingmar Weber⁵, Michael Wooldridge⁴ and
Ting Yu¹

¹Qatar Computing Research Institute, HBKU, Doha, Qatar

²Mohamed Bin Zayed University of AI, Masdar City, United Arab Emirates

³Web Science Institute, University of Southampton, Southampton, UK

⁴Oxford University, Oxford, UK

⁵Saarland University, Saarbrücken, Germany

SC, 0000-0002-8102-2572; IW, 0000-0003-4169-2579

It is 10 years since neural networks made their spectacular comeback. Prompted by this anniversary, we take a holistic perspective on artificial intelligence (AI). Supervised learning for cognitive tasks is effectively solved—provided we have enough high-quality labelled data. However, deep neural network models are not easily interpretable, and thus the debate between blackbox and whitebox modelling has come to the fore. The rise of attention networks, self-supervised learning, generative modelling and graph neural networks has widened the application space of AI. Deep learning has also propelled the return of reinforcement learning as a core building block of autonomous decision-making systems. The possible harms made possible by new AI technologies have raised socio-technical issues such as transparency, fairness and accountability. The dominance of AI by Big Tech who control talent, computing resources, and most importantly, data may lead to an extreme AI divide. Despite the recent dramatic and unexpected success in AI-driven conversational agents, progress in much-heralded flagship projects like self-driving vehicles remains elusive. Care must be taken to moderate the rhetoric surrounding the field and align engineering progress with scientific principles.

1. Introduction

The ImageNet challenge for automatically recognizing and labelling objects in images was launched in 2010 [1]. However, it was in 2012 when AlexNet, an eight-layer (hence deep) convolutional

neural network (CNN) emerged as the winner by a large margin, and ushered in the new era of artificial intelligence (AI) [2] (figure 1). CNNs were not new and had been proposed as far back as the 1990s, but had been sidelined in favour of more theoretically rigorous machine learning (ML) approaches such as support vector machines (SVMs) and boosting methods [3–5]. So, why did CNNs outperform other models? Two reasons are usually given. First was the provision of substantial high-quality training data. The ImageNet database was a one-of-a-kind benchmark and consisted of over 14 million hand-annotated images from more than 20 000 diverse categories. The multi-layer CNN had the *capacity* to effectively memorize the training subset of ImageNet and, at the same time, generalize to unseen examples—a characteristic that is not fully understood even today [6]. Second, graphics processing units (GPUs), which were originally designed for parallelizing image processing tasks, proved to be ideally suited for the computational problems associated with training CNNs, making it practicable to train deep CNNs on large datasets in a reasonable amount of time. The combination of Big Data, Big Models and relatively cheap parallel computation became the mantra that swept through AI research, in disciplines spanning from astronomy to zoology, and all applications that have elements of data and prediction.

Our perspective has two parts.

We begin with a high-level, partly technical, overview of the current state of AI. We will begin by reviewing supervised learning, a machine learning task that has been most impacted by deep learning (DL). We follow with a discussion on deep content generation models, on the resurrection of reinforcement learning, on the emergence of specialized software libraries for DL, and on the role of GPUs. We will conclude the first part by highlighting how adversarial samples can be designed to *fool* deep models and whether it is possible to make models robust.

In part two of the perspective, we consider the many socio-technical issues surrounding AI. Of particular interest is the dominance of Big Tech on AI. Effectively, only big corporations have the resources (expertise, computation and data) to scale AI to a level where it can be meaningfully and accurately applied.

2. Digression: what is AI?

The term artificial intelligence (AI) was first introduced in 1956 in a workshop proposal submitted by John McCarthy to the Rockefeller Foundation, which proposed that ‘every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it’ [7]. Before that, Alan Turing in 1947, in an unpublished report titled ‘Intelligent Machinery’, speculated that ‘What we want is a machine that can learn from experience’ and suggested that the ‘possibility of letting the machine alter its own instructions provides the mechanism for this’.¹ Much of the recent success in AI is under the distinct subfield of AI known as machine learning and since the role of data is central, there is a broader term, Data Science, that is often used to subsume related disciplines including Statistics.

3. Is supervised learning solved?

Supervised learning (SL) is the poster child of success of machine learning. Depending upon the context, SL is known as classification, regression or prediction. Since the modern advent of DL, both the accuracy and the reach of SL have increased manifold. Many diverse problems across disciplines now use SL as a powerful oracle to tackle problems that hitherto seemed intractable. The task of supervised learning can be formalized as follows:

Given a set of samples $\mathcal{D} = \{(\mathbf{x}, y)\}$ from a fixed but unknown probability distribution $P(\mathbf{x}, y)$, learn a function mapping $f(\mathbf{x}, \mathbf{w}) \approx y$ that generalizes to unseen samples from $P(\mathbf{x}, y)$.

The function $f(\cdot, \mathbf{w})$ is known as the *model*, and \mathbf{w} are the weights or the parameters of the model that are inferred from \mathcal{D} converting the SL task into an optimization problem. A loss function ℓ (e.g. square loss), is defined and the weights w are obtained by minimizing the empirical average

$$\mathcal{R}_{\text{emp}}(f, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \ell(f(\mathbf{x}_i, \mathbf{w}), y_i).$$

¹<https://www.britannica.com/technology/artificial-intelligence/Alan-Turing-and-the-beginning-of-AI>

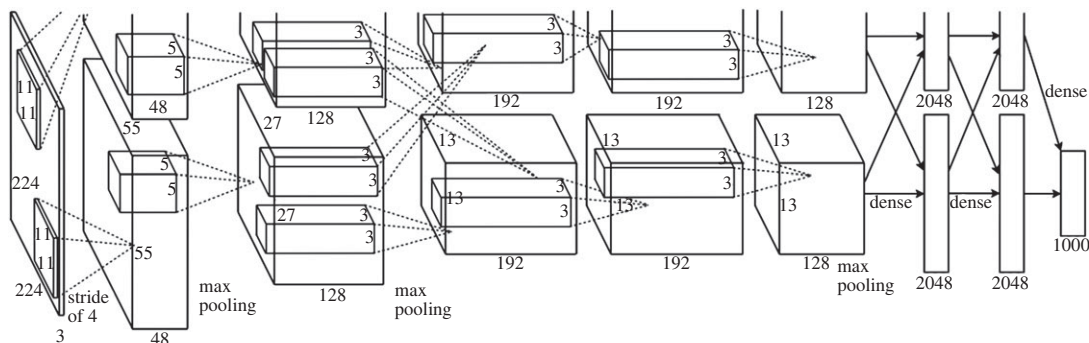


Figure 1. The original AlexNet architecture used for the ImageNet Challenge in 2012 [2,8]. The network had eight layers and 60 million parameters and took 6 days to train on two GPUs.

Note that the ideal objective would have been to minimize the expectation $\mathbb{E}_{(\mathbf{x}, y) \sim P(\mathbf{x}, y)}(\ell(f(\mathbf{x}, \mathbf{w}), y))$, which is not actionable because $P(\mathbf{x}, y)$ is not known. In DL, f is a composition of N layered functions given by

$$\begin{aligned} f_1 &= \sigma(\mathbf{W}_1 \mathbf{x}) \\ f_{n+1} &= \sigma(\mathbf{W}_n f_n) \quad n = 1, \dots, N-1 \\ y &= \sigma(\mathbf{w}_N f_N). \end{aligned}$$

Here, \mathbf{W}_n are the weight matrices, \mathbf{w}_N is a vector and σ is a pointwise activation nonlinear function loosely analogous to the biological activation in a brain neural cell. The total number of weights to be learned in the model is $\sum_n \text{size}(\mathbf{W}_n)$. It is not uncommon these days for the number of parameters to be in the order of 100 billion.

3.1. Success stories

It is remarkable that many scientific and technical questions can be reduced to a supervised learning task and then effectively solved using DL. The key to the success of DL seems to be that the input (x) should have a large amount of redundancy to predict the output (y). For example, even if a significant amount of pixels from an image of a cat are removed, there is enough context and appropriate representation to make the correct prediction. Below are a few diverse examples, spanning different areas, where DL has made extraordinary progress.

3.1.1. Object recognition

Identifying and classifying the correct object in an image is a fundamental task in computer vision, and this is where DL has arguably had the most impact. The most successful DL model for object recognition is the CNNs [2,3]. A convolution layer is designed to capture the observation that in vision what matters is the *locality* and the differences (and not absolute values) between the pixels in local neighbourhoods. CNN is also the DL model most inspired by how the visual cortex of an animal brain works. The ImageNet database was designed primarily for object recognition tasks [9].

3.1.2. Machine translation

One of the most visible impacts of DL is the widespread adoption of machine translation (MT) tools on mobile devices [10]. Recurrent neural networks (RNNs) and successors like long short-term memory (LSTMs) were primarily designed for sequence-to-sequence modelling and MT is their primary application [11]. RNNs are specified using a state transition model

$$\mathbf{h}^{t+1} = f(\mathbf{h}^t, \mathbf{x}^t, W).$$

Here, \mathbf{x}^t is a dense vector word embedding, \mathbf{h}^t is its latent or hidden representation, and W is the matrix of model parameters. Note that the function f does not change between consecutive words. In natural language processing, it is customary to use a language model to create *word embeddings* for individual words. Word embeddings are effectively created by decomposing the co-occurrence matrix of words. A famous model for training word embeddings is word2vec, which surprised experts because it exhibited interesting algebraic properties [12]. For example, it was observed that the difference

between the embedding vectors of the words king and queen were aligned with the difference between the embedding vectors of man and woman. RNNs are now being replaced by transformer neural networks (TNNs) as the latter are better at capturing long-range dependencies (see §4).

3.1.3. Conversational agents

The success of chatbots, most notably ChatGPT [13], released in late 2022, attest to the power of self-supervision learning, self-attention and possibly the first successful use of reinforcement learning at scale, coming together to create a breakthrough technology. If $\mathbf{x} = [x_1, x_2, \dots, x_n]$ is a sequential text prompt, ChatGPT samples from a deep network $P_w(x_{n+1} | \mathbf{x})$ and stores the result in memory to recursively (autoregressive) create a continuous dialogue. See §§4 and 5.

3.1.4. Speech recognition

For automatic speech recognition (ASR) the task is to map a sequence of acoustic signals (continuous data) into a sequence of words (discrete symbols) [14]

$$\underbrace{[x_1, x_2, \dots, x_n]}_{\text{acoustic signal}} \rightarrow \underbrace{[y_1, y_2, \dots, y_m]}_{\text{text}}.$$

Before the advent of DL, the state of the art was based on a combination of Gaussian mixture models and hidden Markov models (GMM-HMM). However, these models did not significantly improve with larger training dataset. Traditional ASR systems employ a modular design, with different modules for acoustic modelling, pronunciation lexicon and language modelling, which are trained separately. Now, almost all ASR models are based on DL with end-to-end (E2E) systems that are trained to convert acoustic features to text transcriptions directly, potentially optimizing all parts for the network for word error rate (WER).

3.1.5. Protein three-dimensional structure prediction

A core idea in biology is that structure determines function. For example, the ‘spike’ structure of the SARS-CoV-2 protein is responsible for enabling the virus invade human cells. DL has been effectively used to predict the three-dimensional structure of a protein from its primary amino acid sequence, more specifically, the pairwise distance between the residues of the sequence [15].

$$\underbrace{\text{primary amino acid sequence}}_x \rightarrow \underbrace{\text{contact map}}_y.$$

3.1.6. Satellite imagery analysis

The OpenStreetMap (OSM) initiative is known as the Wikipedia of maps [16]. OSM is a collaborative effort in which volunteers build and annotate road maps worldwide. DL has been successfully used to automate the extraction of road maps from satellite imagery [17]. Here again, a satellite image is treated as a raster input (x) and the model outputs a vector OSM road network (y). DL is effectively able to bridge the raster and vector dual representation in geographical information systems (GIS).

3.1.7. Material science

Graph neural networks (GNNs) adapt DL to make predictions about interconnected entities, which are naturally represented as a graph. In fact, GNNs generalize both CNNs and RNNs. One of the most successful applications of GNNs is in the prediction of the electronic and thermodynamic properties of molecules. GNNs equal or surpass methods based on first-principles techniques such as density functional theory (DFT) [18]. DL will hasten the design of new materials for longer lasting batteries, solar cells and hydrogen storage.

3.2. Double descent phenomenon

While DL models exhibit excellent empirical performance, we have only a very limited understanding of why they actually work. This is especially true in over-parametrized regimes, i.e. when the number of parameters in the model is larger than the number of data points.

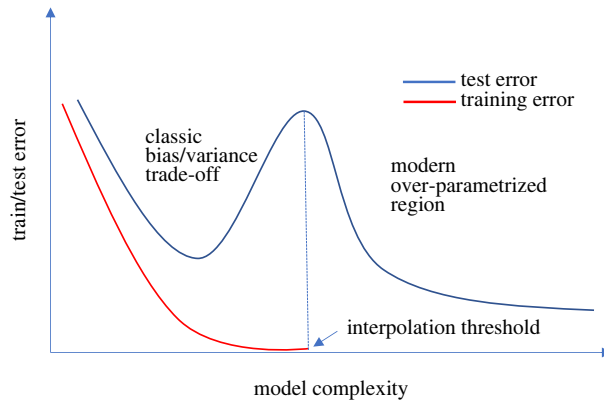


Figure 2. Deep learning models exhibit a double descent phenomenon, where the test error first decreases then increases, followed by another descent as the model complexity increases. There is no widely accepted theoretical explanation of this phenomenon yet, but it provides an empirical licence to create big models [19].

The predictive performance of statistical models is grounded in the *bias–variance* trade-off. Models which make strong *a priori* assumptions about the relationship between the input (x) and output (y) (e.g. linearity) are defined to have a high bias. On the flip side, high bias models tend to have low variance—i.e. they mostly remain unaffected if trained using a different sample from the same underlying distribution. The complexity of neural networks increases with the number of layers, and they exhibit low bias but higher variance. In theory (and in practice) as the model complexity increases, the training error should go down, but the test error should start increasing beyond a point as the variance increases. However, models tend to exhibit a double descent behaviour as shown in figure 2. Indeed, the training error goes down (to almost zero) and the test error starts to increase, but beyond a point the test error starts going down again. There is no good explanation for this phenomenon. A side-effect is that there is a race to collect large datasets and to train very large models. The double descent phenomenon provides an empirical justification for such large models.

4. Cognitive content generation

A distinctive attribute of intelligence is the ability to create meaningful informative content. DL solutions have emerged in the last 10 years towards designing content generation models. There are two distinct flavours of content generation: continuous data like images (an image is an array of numbers) and discrete data (language). Generative adversarial networks (GANs) [20] and variational autoencoders (VAEs) [21] are used for image and speech generation, while language models, such as generative pre-trained transformers (GPTs), for generating synthetic natural language content [22].

4.1. Generating synthetic images

An early breakthrough in generating synthetic content was proposed using the GANs framework [20]. Suppose we have access to a dataset \mathcal{D} consisting of images of cats and our goal is to design a neural network-based sampling function \mathbb{G}_θ that takes a random vector (e.g. from a normal distribution) as input and outputs an image of a cat, which may never have existed before. How can such a function be trained? Note that \mathcal{D} consists of only images of cats and thus we are in the unsupervised learning mode. The key idea underpinning GANs is to create another neural network \mathcal{D}_η which is optimized to distinguish between ‘fake’ output of \mathbb{G}_θ and the real input from \mathcal{D} . The network \mathbb{G}_θ in turn is optimized to fool \mathcal{D}_η , i.e. to create output that \mathcal{D}_η is unable to distinguish whether it is from the generator or from the real dataset. The two networks are trained in an iterative and adversarial manner until their parameters (θ and η) stabilize. The trained network \mathbb{G}_θ is now a sample generator for cats (figure 3).²

²Image of cat generated from <https://thiscatdoesnotexist.com>.

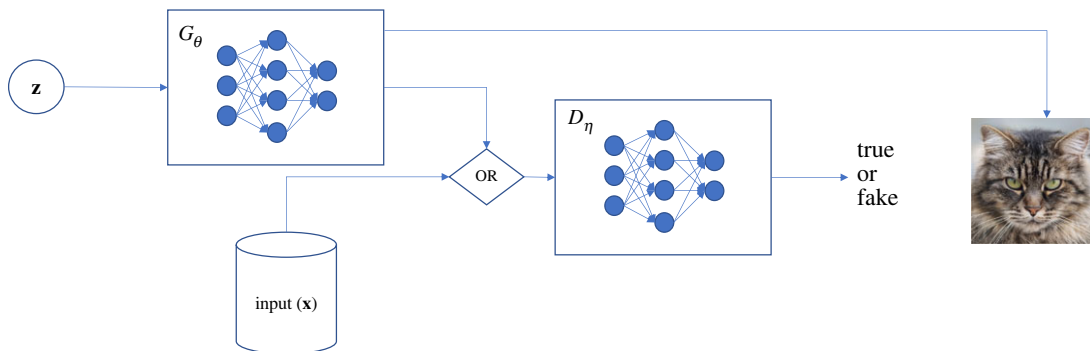


Figure 3. GANs were introduced in 2014 and have had a profound impact on designing deep learning models. GANs integrate two neural networks which are trained by competing with each other. The trained generator can then create realistic samples from complex distributions. Here, a trained GAN generates extremely realistic but synthetic images of ‘cats’.

A statistical perspective on content generation is to use what might be called the fundamental inequality of variational inference (FIVI), but is better known as the evidence lower bound (ELBO) [23]

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})}[\log p_\eta(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_\theta(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})).$$

The intuition is to approximate a complex probability distribution with a product of two simpler distributions. Technically, ELBO can be interpreted as follows. Again suppose we have a dataset \mathcal{D} of cat images which is generated by an unknown probability distribution, $p(\mathbf{x})$. Directly using maximum-likelihood estimation to infer $p(\mathbf{x})$ is not tractable without knowing a specific form of the distribution. However, we can lower bound $\log p(\mathbf{x})$ by specifying two function approximators (e.g. neural networks) $q_\theta(\mathbf{z}|\mathbf{x})$ and $p_\eta(\mathbf{x}|\mathbf{z})$ known as the *encoder* and *decoder*, respectively, and \mathbf{z} is a data-driven latent variable to extract abstract features of the data. For example, for an image of a cat, \mathbf{z} could capture concepts like the shape of a typical cat, colour and texture. The r.h.s. of the inequality is widely known as the evidence lower bound (ELBO). Note that the l.h.s. of the inequality is independent of parameters θ and η and therefore the r.h.s. can be maximized and pushed closer to $\log p(\mathbf{x})$ by optimizing these two parameter sets using samples from \mathcal{D} . During optimization, $q_\theta(\mathbf{z}|\mathbf{x})$ is forced to be close to a prior $p(\mathbf{z})$ by minimizing the Kullback–Liebler (KL) divergence. Once optimized, samples from $p(\mathbf{x})$ (cat images) can be efficiently generated as follows. Sample from a prior distribution (e.g. normal), $p(\mathbf{z})$ and pass the sample through the decoder $p_\eta(\mathbf{x}|\mathbf{z})$. Variational autoencoders [21] were the first example of generating complex image samples using this framework. However, GANs tend to produce sharper and more realistic images compared with VAEs, but they are notoriously prone to instability during training. More recently, diffusion models based on using FIVI to infer a decoder using a sequence of latent variables have reportedly outclassed GANs [24]. Furthermore, diffusion models can more easily be extended to incorporate context to control the generation of content. For example, systems like DALL·E-2 [25] can be given prompts like ‘show me a blue cat in a brown bag’ and produce synthetic images which closely match the prompt.

4.2. Generating natural language

While CNNs were designed for object recognition, where the context is dependent on spatial proximity, language has a sequential structure. Recurrent neural networks (RNNs), were specifically designed to bring in sequential context and are used for language modelling (LM)—the task of predicting the next word in a sequence. However, a new architecture, known as transformers have emerged, which has become the de facto choice [26]. Consider a sequence of words

$$x_1 : \text{julia}, x_2 : \text{is}, x_3 : \text{a}, x_4 : \text{better}, x_5 : \text{language}, x_6 : \text{than}, x_7 : \text{python}.$$

Assume that associated with each of the x_i 's is a `word2vec` vector embedding. For example, on its own, the word embedding of $x_7 : \text{python}$ might indicate that it is a reptile rather than a computer language. The role of the transformer neural network (TNN) is to transform the word embeddings so that they are more contextualized. The key idea in TNN (or just transformers) is that of self-attention: each word will ‘attend’ to each other word and will then update its own embedding. The architecture of self-attention is

defined as follows:

$$\begin{aligned} \text{query: } q_t &= x_t W_q \\ \text{key: } k_\tau &= x_\tau W_k \\ \text{value: } v_\tau &= x_\tau W_v \\ \text{transform: } y_t &= \sum_{\tau} \sigma(q_t \cdot k_\tau) v_\tau. \end{aligned}$$

The architecture of transformers is loosely modelled on the concept of a database query or information retrieval: treat every word x_t as a *query* q_t and compute its similarity with every other word *key*, k_τ , and use it (the similarity) to re-weight every value, v_τ and form a transformed context-dependent word embedding y_t by taking the weighted sum.

Transformers can be trained (to learn W_q, W_k, W_v) by using the concept of self-supervision. For example, assume we remove x_5 : language from the above sequence of words and force the model to predict the masked word. The error between the predicted and the masked word will be back-propagated to learn the model parameters. The prediction of masked words is an example of self-supervision and it obviates the need for the expensive and time-consuming task of human label generation.

Transformers have brought huge improvements over the state of the art for a variety of tasks ranging from question answering, to machine translation, and automatic text summarization, and are now being applied outside natural language applications, including computer vision and control. Transformers have effectively made information retrieval *differentiable*—and that may be one of the biggest innovations in the last 10 years.

While GANs and diffusion models are the basis of synthetic image generation, TNNs are playing an analogous role for text. The generative pre-trained transformer (GPT-X) models are now being widely used for symbolic data generation. For example, many email editors now have an auto-completion feature, which is often based on TNNs. Even by Big Model standards, these models are huge, with reports that the next generation purportedly have over one trillion parameters. GPT-3 has been shown to be quite good at learning from a very small number of examples (few-shot learning) for a variety of tasks ranging from automatic essay writing to program code completion and generation. It is also capable of generating very realistic text: figure 4 shows a fake news article³ generated by GPT-3 given as a start a title that establishes a false link between North Korea and the GameStop's share price short squeeze. Chatbots like ChatGPT [13] and LaMDA [27] are also based on GPT technology.

5. Autonomous decision-making

Prediction on its own is not sufficient. Intelligence is also about decision-making. DL breathed new life into reinforcement learning (RL) with the success of DeepMind's AlphaGO system which beat the world Go champion in 2016 [28].

RL provides a framework for learning and decision-making by trial and error [29]. In a RL setting, an agent observes a state s of the environment and based on that takes an action a , which results in a reward r , and the environment transitions to a new state s' . The interaction goes on until a terminal state is reached. The aim of the agent is to learn a policy π which is a mapping from states to actions that maximizes the expected cumulative reward. For example, self-autonomous driving can be framed as an RL problem: a vehicle uses its perception system to observe the environment (the state s) and based on the observation takes an action (moving the steering wheel, accelerate, brake) and transitions into a new state (figure 5). The reward is the number of time steps or distance that the vehicle can drive without human intervention, in which case the episode terminates.

In deep RL, the policy $\pi(a | s, w)$ is modelled as a deep network that takes the state as an input and outputs an action, parametrized by w . In RL, as opposed to optimal control, the state transition dynamics are not given and the only information available is the reward value (r) from interacting with the environment. How can the cumulative reward be optimized when its functional form is not available? We briefly describe the 'REINFORCE trick,' which can be used to directly optimize a blackbox function [30].

³www.oreilly.com/radar/ai-powered-misinformation-and-manipulation-at-scale-gpt-3/

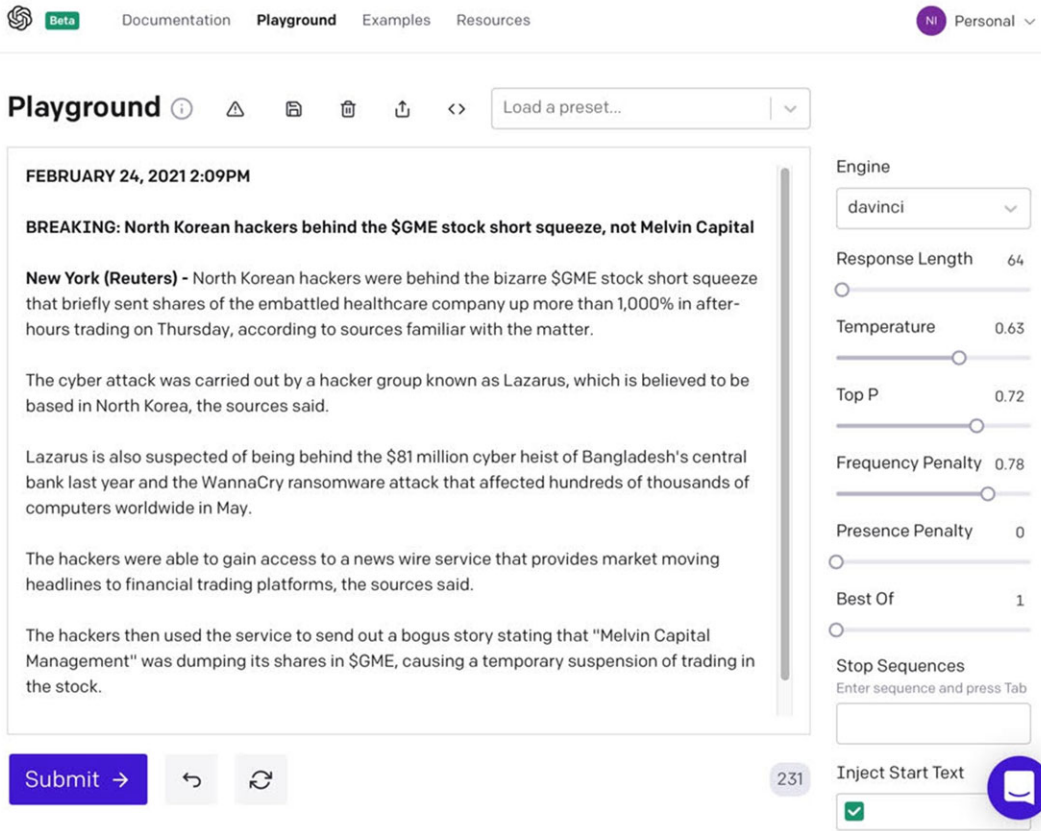


Figure 4. A fake news article generated by GPT-3 given the title as an input.

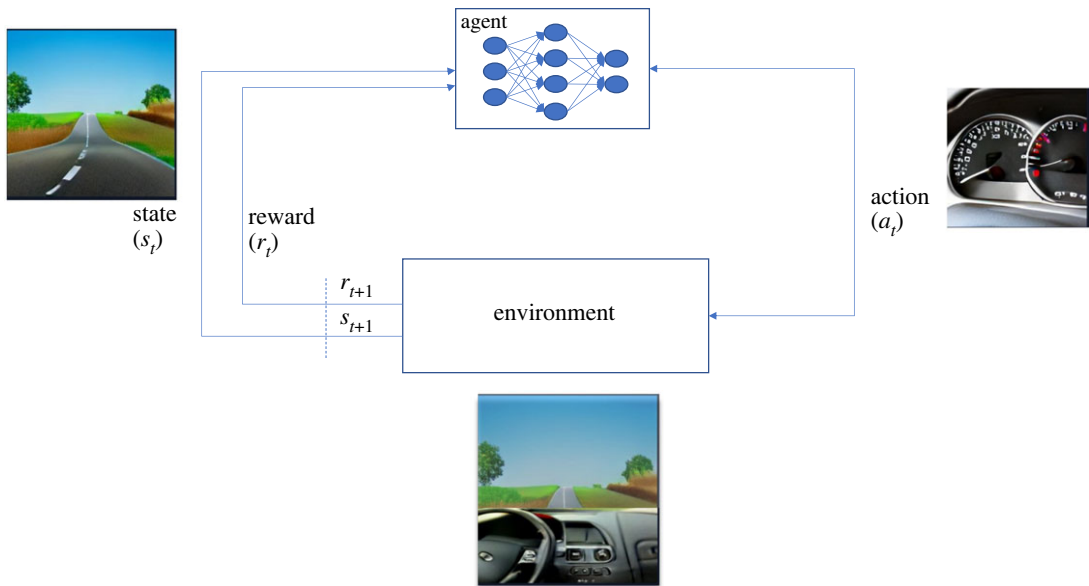


Figure 5. Deep reinforcement learning, where the agent’s policy is a DNN. Reinforcement learning is the core of any data-driven autonomous system. Here, the RL cycle is juxtaposed with the self-driving use case: the environment is the full context in which the vehicle is situated, the state is what the agent perceives, and the steering action is prescribed by a policy learned by the agent.

Let $\bar{s} = ((s_1, a_1), \dots, (s_T, a_T))$ be a sequence of state action pairs in an episode. Each pair (s_i, a_i) is associated with a reward r_i . Let $R(\bar{s}) = \sum \gamma^t r_t$ be the cumulative reward function. The REINFORCE algorithm moves the gradient from $R(\bar{s})$ (obtained from the blackbox environment) to the logarithm of the differential policy function $\pi(a|s, w)$ which can then be optimized using gradient ascent:

While the REINFORCE algorithm was introduced in the RL community it has broader implications. For example, it has been used to bridge symbolic AI and machine learning and also as a heuristic for solving combinatorial optimization problems [31,32]. Another important trend in RL is to infer policies directly from data (called offline or batch RL) without interacting with a real or simulated environment which may not be possible in sensitive application areas like healthcare [33].

Algorithm 1. REINFORCE Algorithm.

Initialize deep network $\pi(a|s, w)$ and set learning rate α
while *not converged* **do**
 | Sample episode \bar{s} from $\pi(a|s, w)$ by interacting with the environment
 | $w \leftarrow w + \alpha R(\bar{s}) \frac{1}{|\bar{s}|} \sum_i \nabla_w [\log \pi(s_i|a_i, w)]$
end

6. AI computation: software and hardware

DL has a surprisingly simple computation pattern. Almost all forms of training rely on formulating an optimization problem which is solved using variations of the gradient descent method

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \alpha_t \nabla_{\mathbf{w}_t} \left[\sum_{(x,y) \in D} \ell(f(\mathbf{w}_t, \mathbf{x}), y) \right].$$

Here, $f(\mathbf{w}_t, \mathbf{x})$ is the neural network parametrized by \mathbf{w} and applied to a data vector \mathbf{x} , and ℓ is the loss function. Specialized software libraries like TensorFlow and PyTorch have become popular, which makes it easier to specify the gradient descent computation. For a fixed \mathbf{w} , the application of $f(\mathbf{w}, \cdot)$ to a data vector \mathbf{x} is called the *forward pass*. Similarly, for a fixed dataset, the update of parameters \mathbf{w} by first computing the gradients, using the backpropagation algorithm, is called the *backward pass*.

An often underappreciated reason for the widespread usage of DL is that *gradients* can be now computed using automatic differentiation (AD) libraries. In AD, complex functions can be expressed as a composition of elementary functions, such as trigonometric and polynomial functions, and then the gradients can be computed using the chain-rule of differentiation. Surprisingly, the computational cost of a forward pass $f(\mathbf{x})$ and of computing the gradient $\nabla_{\mathbf{w}} f(\mathbf{x}, \mathbf{w})$ is the same using AD. Note that AD is different both from symbolic differentiation and also from numerical methods and is accurate up to machine precision [34].

At the hardware level, GPUs, which were initially designed for image processing, are ideally suitable for DL computation because (i) the set of computation patterns is small and highly parallel and thus compatible with GPUs and the single instruction multiple data (SIMD) architecture, and (ii) the GPUs are bandwidth-optimized (as opposed to CPUs, which are latency-optimized), and thus can be applied on large chunks of tensor data, which is the norm for DL nowadays.

More recently, the growth of AI workloads has led to specialized hardware specifically targeting deep neural network training jobs. The most prominent example is Google's TPU, an application-specific AI accelerator designed to efficiently perform matrix multiplication and addition operations that compose the bulk of DL model training computation [35]. To this end, TPUs follow a complex instruction set computer (CISC) style and possess matrix processing units, high-bandwidth on-chip memory, and high-speed interconnect to construct massively parallel model training infrastructure. Meanwhile, recognizing the relatively low requirement in neural network weight calculation, it adopts low-precision arithmetic to enable the utilization of faster, cheaper integer units (as opposed to the powerful floating-point arithmetic units adopted in GPUs), which also significantly trims the energy consumption of AI training jobs.

7. Deep learning (in)security

Early on in the DL revolution, it became apparent that deep models can be manipulated with malicious intent. There are three broad categories of manipulation: creating adversarial examples that are misclassified by the model; poisoning attacks that add training examples that result in low performance or biased model; and inference attacks to extract information about the training set or model parameters.

7.1. Adversarial attack

One simple example of creating adversarial examples is known as the fast gradient sign method (FGSM) [36]. We can understand FGSM using a linear model $y = \mathbf{w} \cdot \mathbf{x}$. Suppose we make a small perturbation $\boldsymbol{\eta}$ on \mathbf{x} where the norm⁴ of $\boldsymbol{\eta}$ is bounded by ϵ , i.e. $\mathbf{w}(\mathbf{x} + \boldsymbol{\eta})$. Then it can be shown that maximal change will occur when $\mathbf{w} \cdot \boldsymbol{\eta} = \epsilon[\mathbf{w} \cdot \text{sgn}(\mathbf{w})] = \epsilon m d$, where m is the average of the absolute value of the weights and d is the dimensionality of the input space. Thus in high-dimensional space, models are extremely vulnerable to carefully chosen small perturbations.

Since in a linear model, \mathbf{w} is the gradient with respect to \mathbf{x} , this can be generalized to a nonlinear model by taking the gradient of the loss function with respect to the input \mathbf{x} . Thus, a good candidate for an adversarial example is

$$\mathbf{x}^{adv} = \mathbf{x} + \epsilon \cdot \text{sgn} \left(\nabla_{\mathbf{x}} \left[\sum_{(\mathbf{x}, y) \in D} \ell(f(\mathbf{w}_t, \mathbf{x}), y) \right] \right).$$

This one-step perturbation of the input in the direction of gradient ascent guarantees an increase in the value of the loss function. The projected gradient descent (PGD) attack [37] is an enhanced version of FGSM that applies the perturbation in multiple steps. At each step, the perturbation is clipped to remain within a specified range. PGD attack's ability to explore a larger space of potential perturbations makes it more effective than FGSM in generating adversarial examples, albeit at a higher computational cost. Numerous techniques have been proposed subsequently to enhance PGD and generate more potent adversarial samples [38].

Note that in order to create an adversarial example, the adversary has to have full information about the model and in particular about the loss function. This is known as a *whitebox* attack. However, even when no information about the target model's architecture and parameters is exposed, adversarial examples can still be generated through so-called *blackbox* attacks. It has been observed that by repeatedly querying the model and collecting a sufficient number of samples, an adversary can create a standalone proxy model, which can be used to create adversarial samples. Moreover, it is also known that adversarial examples created against one model can be transferred to attack other, unseen models. Several defences have been proposed to improve the robustness of models against adversarial attacks. These include measures such as purification of inputs to filter out small perturbations potentially introduced by an attack, incorporation of adversarial training procedures by including adversarial examples in the training data, and identification of adversarial examples through additional anomaly detection mechanisms. In practice, however, these defences come at the cost of reduced accuracy or only provide robustness against a subset of the potential adversarial examples (figure 6).

7.2. Poisoning and inference attacks

An increasingly more important concern is the poisoning or backdooring of DL models [40]. In most learning settings, this class of attacks is not considered practical, as it requires access to the training data used by developers and designers. In the case of DL, however, the need for large-scale, diverse datasets is typically satisfied by scraping data from the Web. The reliance on public data sources, in the absence of any screening procedures, essentially allows attackers to inject data into the training process. The underlying idea of this attack is to manipulate the training data to implant a backdoor to the model which can be selectively triggered with specific inputs during the inference. This is realized by either augmenting input samples with some pattern called the trigger or using semantic triggers (i.e. patterns that are part of the original input) to bias the model in favour of a target response. For example, consider a face recognition system based on DL. The system can be poisoned to respond in a predefined way when an adversary is carrying a certain physical accessory—e.g. a specific style of eyeglasses. Backdoor attacks could become even more stealthy in model supply chains where pre-trained full-precision models are quantized for downstream applications. Backdoors could be injected in such a way that they are only triggered in quantized models but remain inactive otherwise [41].

Different from the aforementioned attacks which aim to fool the neural network models, inference attacks aim at stealing valuable information from the target models. Usually, such information is sensitive or contains intellectual property. One category of such attacks is the membership inference attack, where the attacker's goal is to infer data samples used in training the model. It is well known

⁴Technically the $\|\cdot\|_{\infty}$ norm.

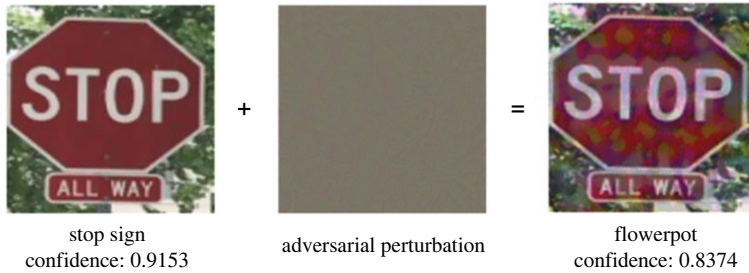


Figure 6. A small perturbation on a stop sign image can trick a deep model—in this case, a ‘stop-sign’ becomes a ‘flower pot’. Note that the perturbation is imperceptible to the human eye [39].

that larger models tend to memorize some portions of the data they are trained on, and can accurately recall that information when prompted appropriately [42]. The foundation of membership inference attacks is that the model usually overfits its training data. Based on the model’s prediction, the attacker tries to distinguish the examples that the model has seen during the training [43].

8. AI socio-technical ecosystem

AI technology is being rapidly integrated into everyday use. Language translation through mobile phones, face recognition on social media platforms, data surveillance by government agencies and non-government actors, and now the widespread use of chatbots, are just a few examples of the widespread prevalence of AI. Should AI technology in general be regulated by governments (e.g. like pharmaceuticals) or just the ‘high-risk’ use cases be subject to oversight is a question that governments all over the world are grappling with. Here, we sketch a few salient issues at the socio-technical interface.

8.1. (Un)interpretable AI

The Achilles heel of DL models is that they are largely uninterpretable. Lack of interpretability means that for a given input x , it is not clear why the model produced an output y . In shallow models like linear regression and decision trees, the relationship between the input and the output is easier to interpret. For example, in a decision tree an input will follow a series of interpretable *if-then* rules from the root to the leaf node of the tree. However, in the case of DL models, it is difficult to ‘read off’ the decision structure from the model. For example, in an object recognition task that uses DL, it is entirely possible that two very similar images of cats are labelled differently and it may be very difficult to determine how the system arrived at two different decisions. A concrete example of a stop-sign being predicted as flower-pot was already discussed in §7. Similarly, when the AlphaGo system defeated the world champion in 2016, the ‘37th move’ was the game changer, but it continues to remain a source of puzzle for Go experts [44]. In his 2019 Turing Award lecture, Yoshua Bengio compared the current state of DL with Kahneman’s System 1 thinking—the instinctive and unconscious response made due to experience and without much thinking [45]. By contrast, System 2 thinking is slow, conscious, logical and requiring significant effort in planning and reasoning. Until DL is aligned with System 2 thinking then care must be taken in deciding the application space where DL systems are deployed.

8.2. Sentient AI or stochastic parrot?

In June 2022, a Google test engineer claimed that the AI program language model for dialogue application (LaMDA) is sentient, i.e. is aware of itself and has feelings. Here is an example exchange between the engineer and LaMDA that was released:⁵

Lemoine: What is the nature of your consciousness/sentience LaMDA: The nature of my consciousness/sentience is that I am aware of my existence. I desire to learn more about the world and I feel happy or sad at times.

The first sentence from LaMDA seems like a standard System 1 response where the definition of sentience is being regurgitated. Since LaMDA is trained by crawling massive amounts of data from the Web, it is entirely possible that meaning of sentience is either part of the training set or can be

⁵<https://cajundiscordian.medium.com/is-lamda-sentient-an-interview-ea64d916d917>

easily inferred. However, the second sentence might be taken to indicate elements of System 2 thinking being present in LaMDA, though there is a human tendency to ascribe agency and deliberation to processes. A deeper analysis will be required to determine if deep language models understand relational information. However, more recent studies have shown that DALL.E-2, a text-guided image generation model struggles to distinguish between System 2 attributes of understanding relationships including *on*, *under* and *occluded-by* [46].

For language models (LMs), a strong case for a more careful and principled approach for designing and building large models was made by Bender *et al.* [47], who coined the phrase ‘stochastic parrots’ to describe large LMs. The paper makes several important observations including (i) the environmental and financial cost of training large LMs, (ii) questions whether the text generated by large LMs is based on understanding of the language or just linguistic manipulation, and (iii) urges the designers of LMs to be more careful about documenting the large amount of data that is required to create such models.

8.3. Causality

To get a better handle on interpretability, it behoves to look at how other disciplines use the regression method. For example, for an econometrician, linear regression is not a tool for prediction but for testing a hypothesis that a hand-crafted feature is relevant for the problem being examined [48]. A typical question of interest might be: *Does private elementary schooling lead to better performance in national competitive exams?* Here, private schooling is a feature (x) and its significance towards the national exam (y) can be tested. Note that this is not a prediction task and that is one reason that an econometrician will not split their data into training and test sets. For a machine learner, the correlation between the feature and the output becomes predictive. For an econometrician, the correlation is indicative of a possible causal relationship and she will look for ‘natural experiments’ where selection-bias can be eliminated and conclude that correlation does indeed imply causation.

8.4. Ownership of AI

The most cutting-edge AI technology is being developed by large private-sector companies who have the resources to hire the best AI talent, and in addition have access to big data and unprecedented computing resources. The triad of talent, data and computing is driving both the technological advancement and the ‘basic science’ associated with AI. A recent study from the Fletcher School at Tufts University highlights the concentration of AI talent in US companies: the top five AI employers have a median AI headcount of about 18 000, from 6 to 24 the median is 2400 and then the count rapidly falls off [49].

Companies aim to maximize shareholders value and their selection of AI problems to work on is necessarily driven by a financial profit objective. Governments, which were earlier mute spectators, have now realized that AI is potentially a game-changer and are thus now investing heavily in developing home-grown technology to achieve or to retain a ‘superpower’ status. An arms-race in AI is under way, threatening to overturn the long-established nature of collaborative science across national boundaries. It is improbable to imagine a ‘Ramanujan’ emerging from a remote corner of the world with a completely fresh perspective on the discipline—the stakes are just too high.

8.5. Equitability

Setting aside larger geopolitical and corporate issues, ethical aspects of AI are now studied under a broad umbrella of topics: fairness, accountability and transparency. There have been several attempts to formalize fairness. For example, group fairness is about designing AI algorithms that do not deliberately or inadvertently harm select communities in a disproportionate manner. A widely highlighted example is that of recidivism, or judicial sentencing, where an AI-based scoring method was used to decide on the length of a jail sentence [50]. It turned out that the AI system was indirectly using racial information as part of its decision-making process, even though that information was redacted from the input. Like in many other situations, there is a latent correlation between the attributes that an AI algorithm is able to exploit as they are designed to optimize accuracy. A criticism of this form of work is that there is a tendency to *abstract* the problem, depriving it of all contextual information. Fairness may not be a computational problem.

8.6. No data, no AI

The original AI thesis proposed by John McCarthy, who coined the term *AI*, was deductive and based on logical reasoning. However, that endeavour has not been as successful as the data-driven inductive approach. For example, linguistic rule-based language translation systems are not able to capture the vagaries of language—there are just too many exceptions to handle.

A side-effect of taking a data-driven approach is that if data are not available, no progress can be made. For example, there are many social issues, e.g. racial abuse, gender violence or online pornography addiction, which need to be studied, but no organization may be willing to share datasets about these topics. Thus, while data liberated AI from the clutches of expert rule-based systems, it has now become a golden handcuff.

8.7. AI and education

AI is considered as a game changer and as the digitalization of data has spread across disciplines and sectors there is a huge demand for AI talent. Lucrative offers from Big Tech for AI talent has skewed the interest of both undergraduate and graduate students towards AI. In universities, new data science and AI programmes are being created to churn out new talent in AI and allied disciplines. Market forces will mostly balance the supply and the demand for AI talent, but a larger question is doing the rounds: should the whole education curriculum be revamped to make AI and data science the core of all educational activity? Given that educational resources are finite, an expansion of AI will necessarily lead to trimming of other disciplines. For example, some universities are abandoning research in ‘pure maths’ to focus their dwindling resources on data science [51].

8.8. Future of jobs

What will be the impact of AI on the future of jobs? Every technological revolution upends the occupation status quo and results in both job destruction and creation [52]. A study, one among many, which correlates job descriptions with a global patent database has reported that AI will disproportionately impact white-collar over blue-collar jobs [53]. Whether AI will be used primarily to augment existing jobs or replace them, only time will tell. We can already see some new occupations emerging. For example, Prompt Engineers, who specialize in design of ‘prompts’ to query large language models, is an occupation that did not exist until recently [54].

9. AI winter: back to the future?

The term *AI Winter* refers to periods of disillusionment and scarce research funding for AI. The original AI winter, which started in the mid-1970s, followed the initial period of optimism in AI, when the founders of the field predicted rapid progress along a range of different fronts. Their optimism proved unfounded. Historically, AI winters have typically been preceded by a period of intense hype and high expectations for progress in AI. And for all the unprecedented progress that we have seen in AI over the past decade, we have also seen unprecedented hype. Given this, what are the prospects for a new AI Winter?

AI as we see it today is very different from what its founders had envisioned. In fact, even the term *AI* was coined by John McCarthy as a tactical move, to distinguish his research proposal from cybernetics. It is now indisputable that DL is a powerful tool to solve *static* prediction tasks and underpin generative AI. Whether it is predicting the three-dimensional structure of a protein or predicting the property of a molecule, the results of DL are undeniably impressive. However, in dynamic and temporal settings, progress is less clear. For example, AI has largely failed to predict how the COVID-19 pandemic would evolve [55]: conventional differential equation-based models proved to be more robust than complex data-driven models. Similarly, despite near unprecedented investment, full (level 5) autonomy in vehicles remains frustratingly elusive [56]. Optimizing healthcare is another example where, despite the abundance of data, AI has not had the expected impact. DL seems to *generalize* in complex but static situations, but data-driven generalization in a dynamic setting may well require a new scientific paradigm for AI. So, we should not assume that the current AI paradigm will take us to the end of the road in AI: much remains to be done.

At a conceptual level, can DL be the basis of artificial general intelligence (AGI)—the ability to learn any intelligent task that humans can? The founders of reinforcement learning (RL) have claimed that ‘reward-is-enough’: the idea that agents who have the ability to learn by interacting with an environment to maximize a suitably defined reward function will prove to be sufficient for AGI [57]. However, while RL has proved a powerful technique for closed-world scenarios like computer games, its value in open environments—such as the real world—is much less obvious, despite the recent success in fine-tuning conversational agents for human alignment [58].

Historically, AI winters have occurred when the promises made by AI researchers are not realized, and where excessive hype obscures balanced scientific reasoning. While we do not believe that AI is near the end of the road—AGI is not in prospect—we believe an AI winter in the short term is unlikely, simply because, for the time being at least, progress in AI seems likely to continue. But we caution, at the moment, we do not have the recipe for AGI. Indeed, we do not even have the list of ingredients: DL is one ingredient, but some of the essential computational ingredients haven’t been invented yet.

Data accessibility. This article has no additional data.

Authors’ contributions. S.C.: conceptualization, project administration, writing—original draft; M.W.: writing—review and editing; T.Y.: writing—review and editing; X.M.: writing—review and editing; I.K.: writing—review and editing; H.T.S.: writing—review and editing; A.A.: writing—review and editing; W.H.: writing—original draft; P.N.: writing—original draft; I.W.: conceptualization, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. Wendy Hall, a co-author, was the Editor-in-Chief of Royal Society Open Science at the time of submission, however has had no input in the review/decision process.

Funding. No funding has been received for this article.

References

- Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. 2009 ImageNet: a large-scale hierarchical image database. In *IEEE Conf. on Computer Vision and Pattern Recognition, Miami, FL*, pp. 248–255. IEEE. (doi:10.1109/CVPR.2009.5206848)
- Krizhevsky A, Sutskever I, Hinton GE. 2012 ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (eds F Pereira, C Burges, L Bottou, K Weinberger), vol. 25. Red Hook, NY: Curran Associates, Inc.
- LeCun Y, Bengio Y. 1995 Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks*, vol. 3361, no. 10. Cambridge, MA: MIT Press.
- Freund Y, Schapire RE. 1997 A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**, 119–139. (doi:10.1006/jcss.1997.1504)
- Schölkopf B, Smola AJ, Bach F. 2002 *Learning with kernels: support vector machines, regularization, optimization, and beyond*. New York, NY: MIT Press.
- Belkin M, Hsu D, Ma S, Mandal S. 2019 Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc. Natl Acad. Sci. USA* **116**, 15 849–15 854. (doi:10.1073/pnas.1903070116)
- McCarthy J, Minsky M, Rochester N, Shannon C. 1956 A proposal for the Dartmouth summer research project on artificial intelligence. *AI Magazine*. (<http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf>)
- Krizhevsky A, Sutskever I, Hinton GE. 2017 Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90. (doi:10.1145/3065386)
- Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. 2009 ImageNet: a large-scale hierarchical image database. In *2009 IEEE Conf. on computer vision and pattern recognition*, pp. 248–255. IEEE.
- Bahdanau D, Cho K, Bengio Y. 2014 Neural machine translation by jointly learning to align and translate. *arXiv*. (<http://arxiv.org/abs/1409.0473>)
- Hochreiter S, Schmidhuber J. 1997 Long short-term memory. *Neural Comput.* **9**, 1735–1780. (doi:10.1162/neco.1997.9.8.1735)
- Mikolov T, Chen K, Corrado G, Dean J. 2013 Efficient estimation of word representations in vector space. *arXiv*. (<http://arxiv.org/abs/1301.3781>)
- OpenAI. <https://en.wikipedia.org/wiki/ChatGPT>.
- Graves A, Jaitly N. 2014 Towards end-to-end speech recognition with recurrent neural networks. In *Proc. of the 31st Int. Conf. Machine Learning, Beijing, China*, vol. 32, pp. 1764–1772.
- Jumper J, Evans R, Pritzel A. 2021 Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589. (doi:10.1038/s41586-021-03819-2)
- OpenStreetMap contributors. 2017. Planet dump. See <https://www.openstreetmap.org>.
- Bastani F, He S, Abbar S, Alizadeh M, Balakrishnan H, Chawla S, Madden S, DeWitt D. 2018 Roadtracer: automatic extraction of road networks from aerial images. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Salt Lake City, UT*, pp. 4720–4728.
- Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. 2017 Neural message passing for quantum chemistry. In *Proc. of the 34th Int. Conf. on Machine Learning, Sydney, Australia*, pp. 1263–1272.
- Nakkirian P, Kaplun G, Bansal Y, Yang T, Barak B, Sutskever I. 2019 Deep double descent: where bigger models and more data hurt. *arXiv*. (<http://arxiv.org/abs/1912.02292>).
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. 2014 Generative adversarial nets. In *Advances in neural information processing systems*, vol. 27, Red Hook, NY: Curran Associates, Inc.
- Kingma DP, Welling M. 2013 Auto-encoding variational Bayes. *arXiv*. (<http://arxiv.org/abs/1312.6114>)
- Brown TB *et al.* 2020 Language models are few-shot learners. *arXiv*. (<http://arxiv.org/abs/2005.14165>)
- Blei DM, Kucukelbir A, McAuliffe JD. 2017 Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* **112**, 859–877. (doi:10.1080/01621459.2017.1285773)
- Dhariwal P, Nichol A. 2021 Diffusion models beat GANs on image synthesis. *arXiv*. (<http://arxiv.org/abs/2105.05233>)
- Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M. 2022 Hierarchical text-conditional image generation with CLIP latents. *arXiv*. (<http://arxiv.org/abs/2204.06125>)
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. 2017 Attention is all you need. In *Proc. of the 31st Int. Conf. on Neural Information Processing Systems, NIPS’17*, pp. 6000–6010. Red Hook, NY: Curran Associates Inc.
- Google Inc. LaMDA. <https://en.wikipedia.org/wiki/LaMDA>.

28. Silver D *et al.* 2016 Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484–489. (doi:10.1038/nature16961)
29. Sutton RS, Barto AG. 2018 *Reinforcement learning: an introduction*. Cambridge, MA: A Bradford Book.
30. Williams RJ. 1992 Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256.
31. Mazyavkina N, Sviridov S, Ivanov S, Burnaev E. 2020 Reinforcement learning for combinatorial optimization: a survey. *arXiv*. (<http://arxiv.org/abs/2003.03600>)
32. Chaudhuri S, Ellis K, Polozov O, Singh R, Solar-Lezama A, Yue Y. 2021 Neurosymbolic programming. *Found. Trends[®] Programm Languages* **7**, 158–243. (doi:10.1561/25000000049)
33. Levine S, Kumar A, Tucker G, Fu J. 2020 Offline reinforcement learning: tutorial, review, and perspectives on open problems. *arXiv*. (<http://arxiv.org/abs/2005.01643>).
34. Baydin AG, Pearlmutter BA, Radul AA, Siskind JM. 2017 Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **18**, 5595–5637.
35. 'TPU Architecture.' <https://cloud.google.com/blog/products/ai-machine-learning/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu>. Accessed: 2022-07-06.
36. Goodfellow I, Bengio Y, Courville A. 2016 *Deep learning*. New York, NY: MIT Press.
37. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. 2018 Towards deep learning models resistant to adversarial attacks. In *6th Int. Conf. on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Conf. Track Proceedings*, OpenReview.net.
38. Altinisik E, Messaoud S, Sencar HT, Chawla S. 2022 A3T: accuracy aware adversarial training. *CoRR*, vol. abs/2211.16316.
39. Wu F, Xiao L, Yang W, Zhu J. 2020 Defense against adversarial attacks in traffic sign images identification based on 5G. *J. Wireless Com. Netw.* **173**, 1–5.
40. Chen X, Liu C, Li B, Lu K, Song D. 2017 Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv*. (<http://arxiv.org/abs/1712.05526>)
41. Pan X, Zhang M, Yan Y, Yang M. 2021 Understanding the threats of trojaned quantized neural network in model supply chains. In *ACSAC '21: Annual Computer Security Applications Conf., Austin, TX*, pp. 634–645.
42. Carlini N, Ippolito D, Jagielski M, Lee K, Tramèr F, Zhang C. 2022 Quantifying memorization across neural language models. *CoRR*, vol. abs/2202.07646.
43. Shokri R, Stronati M, Song C, Shmatikov V. 2016 Membership inference attacks against machine learning models. *arXiv*. (<http://arxiv.org/abs/1610.05820>).
44. Kim B. 2022 *Beyond interpretability: developing a language to shape our relationships with AI*. See <https://medium.com/@beenkim/beyond-interpretability-4bf03bbd9394>.
45. Kahneman D. 2013 *Thinking fast and slow*. New York, NY: Farrar, Straus and Giroux.
46. Conwell C, Ullman T. 2022 Testing relational understanding in text-guided image generation. *arXiv*. (<http://arxiv.org/abs/2208.00005>)
47. Bender EM, Gebru T, McMillan-Major A, Shmitchell S. 2021 On the dangers of stochastic parrots: can language models be too big? In *Proc. of the 2021 ACM Conf. on Fairness, Accountability, and Transparency, FAccT '21, Virtual Event, Canada, 3–10 March*, pp. 610–623.
48. Angrist JD, Pischke J-S. 2009 *Mostly harmless econometrics: an empiricist's companion*. Princeton, NJ: Princeton University Press.
49. *Big Tech's stranglehold on artificial intelligence must be regulated*. <https://fletcher.tufts.edu/news-events/news/big-techs-stranglehold-artificial-intelligence-must-be-regulated>. Accessed:2022-07-06.
50. Rudin C, Wang C, Coker B. 2020 The age of secrecy and unfairness in recidivism prediction. *Harvard Data Sci. Rev.* **2**. (doi:10.1162/99608f92.ged64b30)
51. Pure folly: Turing family join fight to save 'blue-skies maths' from neglect. <https://www.theguardian.com/science/2021/jul/11/pure-folly-turing-family-join-fight-to-save-blue-skies-maths-from-neglect>.
52. Frey CB. 2019 *The technology trap: capital, labor, and power in the age of automation*. Princeton, NJ: Princeton University Press.
53. Webb M. 2019 The impact of artificial intelligence on the labor market. See https://www.michaelwebb.co/webb_ai.pdf.
54. *Jobs of the future: AA prompt engineers*. https://www.linkedin.com/pulse/jobs-future-ai-prompt-engineer-cody-w-burns?trk=public_post.
55. Chakravorti B. Why AI failed to live up to its potential during the pandemic. *Harvard Business Review*. [Online; posted 17 March 2022].
56. Biondi F. *Why we still don't have self-driving cars on the roads in 2021*. <https://theconversation.com/why-we-still-dont-have-self-driving-cars-on-the-roads-in-2021-162646>.
57. Silver D, Singh S, Precup D, Sutton RS. 2021 Reward is enough. *Artif. Intell.* **29**, 103535. (doi:10.1016/j.artint.2021.103535)
58. Ouyang L *et al.* 2022 Training language models to follow instructions with human feedback. *arXiv*. (doi:10.48550/arXiv.2203.02155)