A Golden Decade of Deep Learning: Computing Systems & Applications

Jeffrey Dean

The past decade has seen tremendous progress in the field of artificial intelligence thanks to the resurgence of neural networks through deep learning. This has helped improve the ability for computers to see, hear, and understand the world around them, leading to dramatic advances in the application of AI to many fields of science and other areas of human endeavor. In this essay, I examine the reasons for this progress, including the confluence of progress in computing hardware designed to accelerate machine learning and the emergence of open-source software frameworks to dramatically expand the set of people who can use machine learning effectively. I also present a broad overview of some of the areas in which machine learning has been applied over the past decade. Finally, I sketch out some likely directions from which further progress in artificial intelligence will come.

Since the very earliest days of computing, humans have dreamed of being able to create "thinking machines." The field of artificial intelligence was founded in a workshop organized by John McCarthy in 1956 at Dartmouth College, with a group of mathematicians and scientists getting together to "find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves."[1] The workshop participants were optimistic that a few months of focused effort would make real progress on these problems.

The few-month timeline proved overly optimistic. Over the next fifty years, a variety of approaches to creating AI systems came into and fell out of fashion, including logic-based systems, rule-based expert systems, and neural networks.[2] Approaches that involved encoding logical rules about the world and using those rules proved ineffective. Hand-curation of millions of pieces of human knowledge into machine-readable form, with the Cyc project as the most prominent example, proved to be a very labor-intensive undertaking that did not make significant headway on enabling machines to learn on their own.[3] Artificial neural networks, which draw inspiration from real biological neural networks, seemed like a promising approach for much of this time, but ultimately fell out of favor in the 1990s. While they were able to produce impressive results for toy-scale problems, they

https://doi.org/10.1162/DAED_a_01900

were unable to produce interesting results on real-world problems at that time. As an undergraduate student in 1990, I was fascinated by neural networks and felt that they seemed like the right abstraction for creating intelligent machines and was convinced that we simply needed more computational power to enable larger neural networks to tackle larger, more interesting problems. I did an undergraduate thesis on parallel training of neural networks, convinced that if we could use sixty-four processors instead of one to train a single neural network then neural networks could solve more interesting tasks.[4] As it turned out, though, relative to the computers in 1990, we needed about one million times more computational power, not sixty-four times, for neural networks to start making impressive headway on challenging problems! Starting in about 2008, though, thanks to Moore's law, we started to have computers this powerful, and neural networks started their resurgence and rise into prominence as the most promising way to create computers that can see, hear, understand, and learn (along with a rebranding of this approach as "deep learning").

The decade from around 2011 to the time of writing (2021) has shown remarkable progress in the goals set out in that 1956 Dartmouth workshop, and machine learning (ML) and AI are now making sweeping advances across many fields of endeavor, creating opportunities for new kinds of computing experiences and interactions, and dramatically expanding the set of problems that can be solved in the world. This essay focuses on three things: the computing hardware and software systems that have enabled this progress; a sampling of some of the exciting applications of machine learning from the past decade; and a glimpse at how we might create even more powerful machine learning systems, to truly fulfill the goals of creating intelligent machines.

Hardware and software for artificial intelligence. Unlike general-purpose computer code, such as the software you might use every day when you run a word processor or web browser, deep learning algorithms are generally built out of different ways of composing a small number of linear algebra operations: matrix multiplications, vector dot products, and similar operations. Because of this restricted vocabulary of operations, it is possible to build computers or accelerator chips that are tailored to support just these kinds of computations. This specialization enables new efficiencies and design choices relative to general-purpose central processing units (CPUs), which must run a much wider variety of kinds of algorithms.

During the early 2000s, a handful of researchers started to investigate the use of graphics processing units (GPUs) for implementing deep learning algorithms. Although originally designed for rendering graphics, researchers discovered that these devices are also well suited for deep learning algorithms because they have relatively high floating-point computation rates compared with CPUs. In 2004, computer scientists Kyoung-Su Oh and Keechul Jung showed a nearly twenty-fold improvement for a neural network algorithm using a GPU.[5] In 2008, computer scientist Rajat Raina and colleagues demonstrated speedups of as much as 72.6 times from using a GPU versus the best CPU-based implementation for some unsupervised learning algorithms.[6]

These early achievements continued to build, as neural networks trained on GPUs outperformed other methods in a wide variety of computer vision contests.[7] As deep learning methods began showing dramatic improvements in image recognition, speech recognition, and language understanding, and as more computationally intensive models (trained on larger data sets) continued demonstrating improved results, the field of machine learning really took off.[8] Computer systems designers started to look at ways to scale deep learning models to even more computationally intensive heights. One early approach used large-scale distributed systems to train a single deep learning model. Google researchers developed the DistBelief framework, a software system that enabled using large-scale distributed systems for training a single neural network.[9] Using DistBelief, researchers were able to train a single unsupervised neural network model that was two orders of magnitude larger than previous neural networks. The model was trained on a large collection of random frames from YouTube videos, and with a large network and sufficient computation and training data, it demonstrated that individual artificial neurons (the building blocks of neural networks) in the model would learn to recognize high-level concepts like human faces or cats, despite never being given any information about these concepts other than the pixels of raw images.[10]

These successes led system designers to design computational devices that were even better suited and matched to the needs of deep learning algorithms than GPUs. For the purpose of building specialized hardware, deep learning algorithms have two very nice properties. First, they are very tolerant of reduced precision. Unlike many numerical algorithms, which require 32-bit or 64-bit floating-point representations for the numerical stability of the computations, deep learning algorithms are generally fine with 16-bit floating-point representations during training (the process by which neural networks learn from observations), and 8-bit and even 4-bit integer fixed-point representations during inference (the process by which neural networks generate predictions or other outputs from inputs). The use of reduced precision enables more multiplication circuits to be put into the same chip area than if higher-precision multipliers were used, meaning chips can perform more computations per second. Second, the computations needed for deep learning algorithms are almost entirely composed of different sequences of linear algebra operations on dense matrices or vectors, such as matrix multiplications or vector dot products. This led to the observation that making chips and systems that were specialized for low-precision linear algebra computations could give very large benefits in terms of better performance per dollar and better performance per watt. An early chip in this vein was Google's first Tensor Processing Unit (TPUv1), which targeted 8-bit integer computations for deep learning inference and demonstrated one to two order-of-magnitude improvements in speed and performance per watt over contemporary CPUs and GPUs.[11] Deployments of these chips enabled Google to make dramatic improvements in speech recognition accuracy, language translation, and image classification systems. Later TPU systems are composed of custom chips as well as larger-scale systems connecting many of these chips together via high-speed custom networking into pods, large-scale supercomputers designed for training deep learning models.[12] GPU manufacturers like NVIDIA started tailoring later designs toward lower-precision deep learning computations and an explosion of venture capital–funded startups sprung up building various kinds of deep learning accelerator chips, with GraphCore, Cerebras, SambaNova, and Nervana being some of the most well-known.

Alongside the rise of GPUs and other ML-oriented hardware, researchers developed open-source software frameworks that made it easy to express deep learning models and computations. These software frameworks are still critical enablers. Today, open-source frameworks help a broad set of researchers, engineers, and others push forward deep learning research and apply deep learning to an incredibly wide range of problem domains (many of which are discussed below). Some of the earliest frameworks like Torch, developed starting in 2003, drew inspiration from earlier mathematical tools like MatLab and NumPy.[13] Theano, developed in 2010, was an early deep learning–oriented framework that included automatic symbolic differentiation.[14] Automatic differentiation is a useful tool that greatly eases the expression of many gradient-based machine learning algorithms, such as stochastic gradient descent (a technique in which errors in outputs are corrected by comparing the actual output and the desired output and making small adjustments to the model parameters in the direction of the error gradient). DistBelief and Caffe were frameworks developed in the early 2010s that emphasized scale and performance.[15]

TensorFlow is a framework that allows the expression of machine learning computations.[16] It was developed and open-sourced by Google in 2015 and combines ideas from earlier frameworks like Theano and DistBelief.[17] TensorFlow was designed to target a wide variety of systems and allows ML computations to run on desktop computers, mobile phones, large-scale distributed environments in data centers, and web browsers, and targets a wide variety of computation devices, including CPUs, GPUs, and TPUs. The system has been downloaded more than fifty million times and is one of the most popular open-source packages in the world. It has enabled a tremendous range of uses of machine learning by individuals and organizations large and small all around the world.

PyTorch, released in 2016, has gained popularity with researchers for its easy expression of a variety of research ideas using Python.[18] JAX, released in 2018, is a popular open-source Python-oriented library combining sophisticated automatic differentiation and an underlying XLA compiler, also used by TensorFlow to efficiently map machine learning computations onto a variety of different types of hardware.[19]

The importance of open-source machine learning libraries and tools like TensorFlow and PyTorch cannot be overstated. They allow researchers to quickly try ideas and express them on top of these frameworks. As researchers and engineers around the world build on each other's work more easily, the rate of progress in the whole field accelerates!

Research explosion. As a result of research advances, the growing computational capabilities of ML-oriented hardware like GPUs and TPUs, and the widespread adoption of open-source machine learning tools like TensorFlow and PyTorch, there has been a dramatic surge in research output in the field of machine learning and its applications. One strong indicator is the number of papers posted to the machine learning–related categories of arXiv, a popular paper preprint hosting service, with more than thirty-two times as many paper preprints posted in 2018 as in 2009 (a growth rate of more than double every two years).[20] There are now more than one hundred research papers posted to arXiv per day in the machine learning–related subtopic areas, and this growth shows no signs of slowing down.

Application explosion. The transformative growth in computing power, advances in software and hardware systems for machine learning, and the surge of machine learning research have all led to a proliferation of machine learning applications across many areas of science and engineering. By collaborating with experts in critical fields like climate science and health care, machine learning researchers are helping to solve important problems that can be socially beneficial and advance humanity. We truly live in exciting times.

Neuroscience is one important area in which machine learning has accelerated scientific progress. In 2020, researchers studied a fly brain to understand more about how the human brain works. They built a connectome, a synapse-resolution-level map of connectivity of an entire fly brain.[21] But without machine learning and the computational power we now have, this would have taken many years. For example, in the 1970s, it took researchers about ten years to painstakingly map some three hundred neurons within the brain of a worm. By contrast, a fly brain has one hundred thousand neurons, and a mouse brain (the next goal for machine learning–aided connectomics) has about seventy million neurons. A human brain contains about eighty-five billion neurons, with about one thousand connections per neuron. Fortunately, deep learning–based advances in computer vision now make it possible to speed up this previously gargantuan process. And today, thanks to machine learning, you can explore the fly brain for yourself using an interactive 3-D model![22]

Molecular biology. Machine learning can also help us understand more about our genetic makeup and, ultimately, address gene-based disease more effectively. These new techniques allow scientists to explore the landscape of potential experiments much more quickly through more accurate simulation, estimation, and data analysis. One open-source tool, DeepVariant, can more accurately process the raw information coming from DNA sequencing machines (which contain errors introduced by the physical process of reading the genetic sequence) and analyze it to more accurately identify the true genetic variants in the sequence relative to a reference genome data using a convolutional neural network. Once genetic variants have been identified, deep learning can also help to analyze genetic sequences to better understand genetic features of single or multiple DNA mutations that cause particular health or other outcomes. For example, a study led by the Dana-Farber Cancer Institute improved diagnostic yield by 14 percent for genetic variants that lead to prostate cancer and melanoma in a cohort of 2,367 cancer patients.[23]

Health care. Machine learning is also offering new ways to help detect and diagnose disease. For example, when applied to medical images, computer vision can help doctors diagnose a number of serious diseases more quickly and accurately than doctors can on their own.

One impressive example is the ability for deep neural networks to correctly diagnose diabetic retinopathy, generally on par with human ophthalmologists. This ocular disease is the fastest growing cause of preventable blindness (projected to impact 642 million people in 2040).

Deep learning systems can also help detect lung cancer as well or better than trained radiologists. The same goes for breast cancer, skin disease, and other diseases.[24] The application of sequential prediction on medical records can help clinicians determine possible diagnoses and risk levels for chronic illness.[25]

Today's deep learning techniques also give us a much more accurate understanding of how diseases spread, giving us a better chance at prevention. Machine learning helps us model complex events, like the global COVID-19 pandemic, which require comprehensive epidemiological data sets, the development of novel interpretable models, and agent-based simulators to inform public health responses.[26]

Weather, environment, and climate change. Climate change is one of the greatest challenges currently facing humanity. Machine learning can help us better understand the weather and our environment, particularly to predict or forecast both everyday weather and climate disasters.

For weather and precipitation forecasting, computationally intensive physics-based models like the National Oceanic and Atmospheric Administration's

High-Resolution Rapid Refresh (HRRR) have long reigned supreme.[27] Machine learning–based forecasting systems can predict more accurately than the HRRR on short timescales, however, with better spatial resolution and faster forecast computations.[28]

For flood forecasting, neural networks can model river systems around the world (a technique called HydroNets), resulting in more accurate water-level predictions.[29] Utilizing this technology, authorities can send faster flood alerts, for example, to more than two hundred million people in India and Bangladesh.[30]

Machine learning also helps us better analyze satellite imagery. We can rapidly assess damage after a natural disaster (even with limited prior satellite imagery), understand the impact and extent of wildfires, and improve ecological and wildlife monitoring.[31]

Robotics. The physical world is messy, full of unexpected obstacles, slips, and breakages. This makes creating robots that can successfully operate in messy, real-world environments like kitchens, offices, and roadways quite challenging (industrial robotics has already had a significant impact on the world, operating in more-controlled environments like factory assembly lines). To hard-code or program real-world physical tasks, researchers need to anticipate all possible situations a robot might encounter. Machine learning efficiently trains robots to operate effectively in real-world environments through a combination of techniques like reinforcement learning, human demonstration, and natural language instruction. Machine learning also allows a more flexible, adaptable approach, in which robots can learn the best ways to engage in grasping or walking tasks rather than being locked into hard-coded assumptions.

Some interesting research techniques include automated reinforcement learning combined with long-range robotic navigation, teaching a robot to follow natural language instructions (in many languages!), and applying a zero-shot imitation learning framework to help robots better navigate simulated and real-world environments.[32]

Accessibility. It is easy to take for granted our ability to see a beautiful image, to hear a favorite song, or to speak with a loved one. Yet more than one billion people are not able to access the world in these ways. Machine learning improves accessibility by turning these signals – vision, hearing, speech – into other signals that can be well-managed by people with accessibility needs, enabling better access to the world around them. Some application examples include speech-to-text transcription, real-time transcriptions while someone is engaged in conversation, and applications that can help visually impaired users identify their surroundings.[33]

Individualized learning. Machine learning can also be used to create tools and applications that aid individualized learning. The benefits of this will be far reaching, and initial examples include early childhood reading coaching such as Google Read Along (formerly Bolo), which is helping children all over the world learn to read in a variety of different languages,[34] and machine learning tools like Socratic that can help kids learn by giving them intuitive explanations and more detailed information about concepts they are grappling with, across a wide variety of subjects such as mathematics, chemistry, and literature.[35] Personalized learning backed by speech recognition, realistic speech output, and language understanding has the potential to improve educational outcomes across the world.

Computer-aided creativity. Deep learning algorithms show surprising abilities to transform images in sophisticated and creative ways, giving us the ability to easily create spaceships in the style of Monet or the Golden Gate Bridge in the style of Edvard Munch.[36] Via an algorithm for artistic style transfer (developed by machine learning researcher Leon Gatys and colleagues), a neural network can take a real-world image and an image of a painting and automatically render the real-world image in the style of the painter. DALL·E by OpenAI enables users to describe an image using text (“*armchairs in the shape of an avocado*” or “*a loft bedroom with a white bed next to a nightstand, with a fish tank standing beside the bed*”) and generate images that have the properties expressed by the natural language description, making sophisticated tools for artists and other creators to quickly create images of what is in their head.[37]

Machine learning–powered tools are also helping musicians create in ways they never have before.[38] Moving beyond “technology,” these new uses of computing can help anyone create new and unique sounds, rhythms, melodies, or even an entirely new musical instrument.

It is not hard to imagine future tools that can interactively help people create amazing representations of our mental imagery – “*Draw me a beach . . . no, I want it to be nighttime . . . with a full moon . . . and a mother giraffe with a baby next to a surfer coming out of the water*” – by just interactively talking to our computing assistants.

Important building blocks. Federated learning is a powerful machine learning approach that preserves user privacy while leveraging many distinct clients (such as mobile devices or organizations) to collaboratively train a model while keeping the training data decentralized.[39] This enables approaches that have superior privacy properties in large-scale learning systems.[40]

Researchers continue to push the state of the art in federated learning by developing adaptive learning algorithms, techniques for mimicking centralized algorithms in federated settings, substantial improvements in complimentary cryptographic protocols, and more.[41]

Transformers. Language has been at the heart of developing AI since the field began, given how ubiquitous language use and understanding is within our daily lives. Because language deals in symbols, it naturally prompted a symbolic approach to AI in the beginning. But over the years, AI researchers have come to realize that more statistical or pattern-based approaches yield better practical uses. The right types of deep learning can represent and manipulate the layered structure of language quite effectively for a variety of real-world tasks, from translating between languages to labeling images. Much of the work in this space from Google and elsewhere now relies on transformers, a particular style of neural network model originally developed for language problems (but with a growing body of evidence that they are also useful for images, videos, speech, protein folding, and a wide variety of other domains).[42]

There have been several interesting examples of transformers used in scientific settings, such as training on protein sequences to find representations encoding meaningful biological properties, protein generation via language modeling, bioBERT for text mining in biomedical data (with pretrained model and training code), embeddings of scientific text (with code), and medical question answering.[43] Computer scientists Maithra Raghu and Eric Schmidt have provided a comprehensive review of the ways in which deep learning has been used for scientific discovery.[44]

Machine learning for computer systems. Researchers are also applying machine learning to core computer science and computer systems problems themselves. This is an exciting virtuous cycle for machine learning and computing infrastructure research because it could accelerate the whole range of techniques that we apply to other fields. This trend is in fact spawning entire new conferences, such as MLSys.[45] Learning-based approaches are even being applied to database indices, learned sorting algorithms, compiler optimization, graph optimization, and memory allocation.[46]

Future of machine learning. A few interesting threads of research are occurring in the ML research community that will likely be even more interesting if combined.

First, work on sparsely activated models, such as the sparsely gated mixture of experts model, shows how to build very large capacity models in which just a portion of the model is "activated" for any given example (say, just two or three experts out of 2,048 experts).[47] The routing function in such models is trained simultaneously and jointly with the different experts, so that the routing function learns which experts are good at which sorts of examples, and the experts simultaneously learn to specialize for the characteristics of the stream of examples they are given. This is in contrast with most ML models today in which the whole model is activated for every example. Research scientist Ashish Vaswani and colleagues showed that such an approach is simultaneously about nine times more efficient for training, about 2.5 times more efficient for inference, and more accurate (+1 BLEU point, a relatively large improvement in accuracy for a language-translation task).[48]

Second, work on automated machine learning (AutoML), in which techniques such as neural architecture search or evolutionary architectural search can automatically learn effective structures and other aspects of machine learning models or components in order to optimize accuracy for a given task, often involves running many automated experiments, each of which may involve significant amounts of computation.[49]

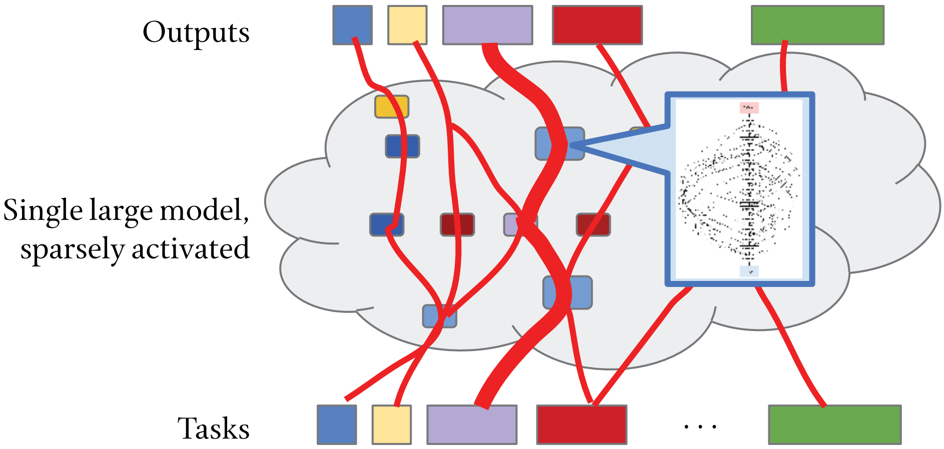
Third, multitask training at modest scales of a few to a few dozen related tasks, or transfer learning from a model trained on a large amount of data for a related task and then fine-tuned on a small amount of data for a new task, has been shown to be very effective in a wide variety of problems.[50] So far, most use of multitask machine learning is usually in the context of a single modality (such as all visual tasks or all textual tasks), although a few authors have considered multimodality settings as well.[51]

A particularly interesting research direction puts these three trends together, with a system running on large-scale ML accelerator hardware, with a goal of training a single model that can perform thousands or millions of tasks. Such a model might be made up of many different components of different structures, with the flow of data between examples being relatively dynamic on an example-by-example basis. The model might use techniques like the sparsely gated mixture of experts and learned routing in order to have a very large capacity model,[52] but one in which a given task or example only sparsely activates a small fraction of the total components in the system (and therefore keeps computational cost and power usage per training example or inference much lower). An interesting direction to explore would be to use dynamic and adaptive amounts of computation for different examples, so that "easy" examples use much less computation than "hard" examples (a relatively unusual property in the machine learning models of today). Figure 1 depicts such a system.

Each component might itself be running some AutoML-like architecture search in order to adapt the structure of the component to the kinds of data that are being routed to that component.[53] New tasks can leverage components trained on other tasks when that is useful. The hope is that through very large scale multitask learning, shared components, and learned routing, the model can very quickly learn to accomplish new tasks to a high level of accuracy, with relatively few examples for each new task (because the model is able to leverage the expertise and internal representations it has already developed in accomplishing other, related tasks).

Building a single machine learning system that can handle millions of tasks, and that can learn to successfully accomplish new tasks automatically, is a true grand challenge in the field of artificial intelligence and computer systems engineering. It will require expertise and advances in many areas, spanning machine learning algorithms, responsible AI topics such as fairness and interpretability, distributed systems, and computer architectures in order to push the field of artificial intelligence forward by building a system that can generalize to solve new tasks independently across the full range of application areas of machine learning.

Figure 1
A Multitask, Sparsely Activated Machine Learning Model



Note: This diagram depicts a design for a large, sparsely activated, multitask model. Each box in the model represents a component. Models for tasks develop by stitching together components, either using human-specified connection patterns or automatically learned connectivity. Each component might be running a small architectural search to adapt to the kinds of data that are being routed to it, and routing decisions making components decide which downstream components are best suited for a particular task or example, based on observed behavior. Source: Author's diagram, including Barret Zoph and Quoc V. Le, "Neural Architecture Search with Reinforcement Learning," arXiv (2016), Figure 7, 15, https://arxiv.org/abs/1611.01578.

Responsible AI development. While AI has the ability to help us in many facets of our lives, all researchers and practitioners should ensure that these approaches are developed responsibly – carefully reviewing issues of bias, fairness, privacy, and other social considerations on how these tools might behave and impact others – and work to address these considerations appropriately.

It is also important to document a clear set of principles to guide responsible development. In 2018, Google published a set of AI principles that guide the company's work in and use of AI.[54] The AI principles lay out important areas of consideration, including issues such as bias, safety, fairness, accountability, transparency, and privacy in machine learning systems. Other organizations and governments have followed this model by publishing their own principles around the use of AI in recent years. It is great to see more organizations publishing their own guidelines and I hope that this trend will continue until it is no longer a trend but a standard by which all machine learning research and development is conducted.

C*onclusions*. The 2010s were truly a golden decade of deep learning research and progress. During this decade, the field made huge strides in some of the most difficult problem areas set out in the 1956 workshop that created the field of AI. Machines became capable of seeing, hearing, and understanding language in ways that early researchers had hoped for. The successes in these core areas enabled a huge range of progress in many scientific domains, enabled our smartphones to become much smarter, and generally opened our eyes to the possibilities of the future as we continue to make progress on creating more sophisticated and powerful deep learning models that help us with our daily lives. The future ahead of us is one in which we will all be more creative and capable thanks to the help provided by incredibly powerful machine learning systems. I cannot wait to see what the future holds!

AUTHOR'S NOTE

Alison Carroll, Heather Struntz, and Phyllis Bendell helped edit this manuscript and made many helpful suggestions for how to present much of the material.

ABOUT THE AUTHOR

Jeffrey Dean, a Fellow of the American Academy since 2016, is a Google Senior Fellow and Senior Vice President for Google Research at Google, Inc.; and Distinguished Fellow at the Stanford University Institute for Human-Centered Artificial Intelligence. He has published in such outlets as *Communications of the ACM*, *ACM Transactions on Computer Systems*, and *Transactions of the Association for Computational Linguistics*. His research papers can be found on Google Scholar at https://scholar.google.com/citations?user=NMS69lQAAAAJ.

ENDNOTES

[1] "Dartmouth Workshop," Wikipedia, last updated October 7, 2021, https://en.wikipedia.org/wiki/Dartmouth_workshop.

[2] "History of Artificial Intelligence," Wikipedia, last updated December 2, 2021, https://en.wikipedia.org/wiki/History_of_artificial_intelligence.

[3] "Cyc," Wikipedia, last updated October 21, 2021, https://en.wikipedia.org/wiki/Cyc.

[4] Jeffrey Dean, "Parallel Implementations of Neural Network Training: Two Back-Propagation Approaches" (senior thesis, University of Minnesota, 1990), https://drive.google.com/file/d/1I1fs4sczbCaACzA9XwxR3DiuXVtqmejL/view.

[5] Kyoung-Su Oh and Keechul Jung, "GPU Implementation of Neural Networks," *Pattern Recognition* 37 (6) (2004), https://www.sciencedirect.com/science/article/abs/pii/S0031320304000524.

[6] Rajat Raina, Anand Madhavan, and Andrew Y. Ng, "Large-Scale Deep Unsupervised Learning Using Graphics Processors," in *Proceedings of the 26th International Conference on Machine Learning* (Princeton, N.J.: International Machine Learning Society, 2009), http://robotics.stanford.edu/~ang/papers/icml09-LargeScaleUnsupervisedDeepLearningGPU.pdf.

[7] Jürgen Schmidhuber, "History of Computer Vision Contests Won by Deep CNNs on GPU," AI Blog, 2017, last updated 2021, https://people.idsia.ch/~juergen/computer-vision-contests-won-by-gpu-cnns.html.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolution Neural Networks," in *Advances in Neural Information Processing Systems 25 (NIPS 2012),* ed. F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Neural Information Processing Systems Foundation, 2012), https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf; Geoffrey Hinton, Li Deng, Dong Yu, et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine* 29 (6) (2012), https://ieeexplore.ieee.org/document/6296526; Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv (2013), https://arxiv.org/abs/1301.3781; and Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to Sequence Learning with Neural Networks," arXiv (2014), https://arxiv.org/abs/1409.3215.

[9] Jeffrey Dean, Greg S. Corrado, Rajat Monga, et al., "Large Scale Distributed Deep Networks" (Mountain View, Calif.: Google, Inc., 2012), https://static.googleusercontent.com/media/research.google.com/en//archive/large_deep_networks_nips2012.pdf.

[10] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, et al., "Building High-Level Features Using Large Scale Unsupervised Learning," in *Proceedings of the 29th International Conference on Machine Learning* (Princeton, N.J.: International Machine Learning Society, 2012), https://static.googleusercontent.com/media/research.google.com/en//archive/unsupervised_icml2012.pdf.

[11] Norman P. Jouppi, Cliff Young, Nishant Patil, et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," in *ISCA '17: Proceedings of the 44th Annual International Symposium on Computer Architecture* (New York: Association for Computing Machinery, 2017), https://dl.acm.org/doi/10.1145/3079856.3080246.

[12] Norman P. Jouppi, Doe Hyun Yoon, George Kurian, et al., "A Domain-Specific Supercomputer for Training Deep Neural Networks," *Communications of the ACM* 63 (7) (2020), https://cacm.acm.org/magazines/2020/7/245702-a-domain-specific-supercomputer-for-training-deep-neural-networks/fulltext.

[13] Ronan Collobert, Samy Bengio, and Johnny Mariéthoz, "Torch: A Modular Machine Learning Software Library," IDIAP Research Report 02-46 (Martigny, Switzerland: Dalle Molle Institute for Perceptual Artificial Intelligence, 2002), https://infoscience.epfl.ch/record/82802/files/rr02-46.pdf.

[14] James Bergstra, Oliver Breuleux, Frédéric Bastien, et al., "Theano : A CPU and GPU Math Compiler in Python," in *Proceedings of the 9th Python in Science Conference (SciPy 2010)* (Austin : Python in Science Conference, 2010), http ://conference.scipy.org/proceedings/scipy2010/pdfs/bergstra.pdf.

[15] Dean et al., "Large Scale Distributed Deep Networks" ; and "Caffe," Berkeley Artificial Intelligence Research, https ://caffe.berkeleyvision.org/.

[16] TensorFlow, https ://www.tensorflow.org/.

[17] "TensorFlow : A System for Large-Scale Machine Learning," Usenix, https ://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[18] Adam Paszke, Sam Gross, Francisco Massa, et al., "PyTorch : An Imperative Style, High-Performance Deep Learning Library," arXiv (2019), https ://arxiv.org/abs/1912.01703.

[19] Matthew Johnson, Peter Hawkins, Jake Vanderplas, et al., "Jax," GitHub, last updated December 15, 2021, http ://github.com/google/jax ; and "XLA : Optimizing Compiler for Machine Learning," TensorFlow, last updated December 2, 2021, https ://www.tensorflow.org/xla.

[20] "Machine Learning," arXiv, https ://arxiv.org/list/cs.LG/recent.

[21] Michal Januszewski, "Releasing the Drosophila Hemibrain Connectome–The Largest Synapse-Resolution Map of Brain Connectivity," Google AI Blog, January 22, 2020, https ://ai.googleblog.com/2020/01/releasing-drosophila-hemibrain.html.

[22] Janelia FlyEM Hemibrain Dataset Information, Hemibrain Neuroglancer Demo, https ://tinyurl.com/25cjs3uk.

[23] Saud H. AlDubayan, Jake R. Conway, Sabrina Y. Camp, et al., "Detection of Pathogenic Variants with Germline Genetic Testing Using Deep Learning vs. Standard Methods in Patients with Prostate Cancer and Melanoma," *JAMA* 324 (19) (2020), https ://jamanetwork.com/journals/jama/article-abstract/2772962 ?guestAccessKey=39889aad-2894-4380-b869-5704ed2f9f6b.

[24] Shravya Shetty and Daniel Tse, "Using AI to Improve Breast Cancer Screening," The Keyword, January 1, 2020, https ://blog.google/technology/health/improving-breast-cancer-screening/ ; and Yuan Liu, Ayush Jain, Clara Eng, et al., A Deep Learning System for Differential Diagnosis of Skin Diseases," *Nature Medicine* 26 (2020), https ://www.nature.com/articles/s41591-020-0842-3.

[25] Alvin Rajkomar and Eyal Oren, "Deep Learning for Electronic Health Records," Google AI Blog, May 8, 2018, https ://ai.googleblog.com/2018/05/deep-learning-for-electronic-health.html.

[26] "Covid-19-open-data," GitHub, updated December 1, 2021, https ://github.com/GoogleCloudPlatform/covid-19-open-data ; Sercan Arik, Chun-Liang Li, Jinsung Yoon, et al., "Interpretable Sequence Learning for Covid-19 Forecasting," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, ed. H. Larochelle, M. Ranzato, R. Hadsell, et al. (Neural Information Processing Systems Foundation, 2020), https ://research.google/pubs/pub49500/ ; and "Agent-based-epidemic-sim," GitHub, updated September 28, 2021, https ://github.com/google-research/agent-based-epidemic-sim.

[27] "The High-Resolution Rapid Refresh (HRRR)," Global Systems Laboratory, U.S. Department of Commerce, https ://rapidrefresh.noaa.gov/hrrr/.

[28] Jason Hickey, "Using Machine Learning to 'Nowcast' Precipitation in High Resolution," Google AI Blog, January 13, 2020, https://ai.googleblog.com/2020/01/using-machine-learning-to-nowcast.html.

[29] Sella Nevo, "The Technology Behind Our Recent Improvements in Flood Forecasting," Google AI Blog, September 3, 2020, https://ai.googleblog.com/2020/09/the-technology-behind-our-recent.html.

[30] Yossi Matias, "A Big Step for Flood Forecasts in India and Bangladesh," The Keyword, September 1, 2020, https://blog.google/technology/ai/flood-forecasts-india-bangladesh/.

[31] Joseph Xu and Pranav Khaitan, "Machine Learning-Based Damage Assessment for Disaster Relief," Google AI Blog, June 16, 2020, https://ai.googleblog.com/2020/06/machine-learning-based-damage.html; Jihyeon Lee, Joseph Z. Xu, Kihyuk Sohn, et al., "Assessing Post-Disaster Damage from Satellite Imagery Using Semi-Supervised Learning Techniques," arXiv (2011), https://arxiv.org/abs/2011.14004; Yossi Matias, "Mapping Wildfires with the Power of Satellite Data," The Keyword, August 20, 2020, https://blog.google/products/search/mapping-wildfires-with-satellite-data/; and Sara Beery and Jonathan Huang, "Leveraging Temporal Context for Object Detection," Google AI Blog, June 26, 2020, https://ai.googleblog.com/2020/06/leveraging-temporal-context-for-object.html.

[32] Aleksandra Faust and Anthony Francis, "Long-Range Robotic Navigation via Automated Reinforcement Learning," Google AI Blog, February 28, 2019, https://ai.googleblog.com/2019/02/long-range-robotic-navigation-via.html; Corey Lynch and Pierre Sermanet, "Language Conditioned Imitation Learning over Unstructured Data" (2020), https://language-play.github.io/; and Xinlei Pan, Tingnan Zhang, Brian Ichter, et al., "Zero-Shot Imitation Learning from Demonstrations for Legged Robot Visual Navigation," *ICRA* (2020), https://research.google/pubs/pub48968/.

[33] Julie Cattiau, "How AI Can Improve Products for People with Impaired Speech," The Keyword, May 7, 2019, https://www.blog.google/outreach-initiatives/accessibility/impaired-speech-recognition/; Sagar Savla, "Real-Time Continuous Transcription with Live Transcribe," Google AI Blog, February 4, 2019, https://ai.googleblog.com/2019/02/real-time-continuous-transcription-with.html; and "Lookout by Google," Google Play, https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.reveal.

[34] Avni Shah, "Making Learning to Read Accessible and Fun with Bolo," The Keyword, September 9, 2019, https://www.blog.google/technology/ai/bolo-literacy/.

[35] Shreyans Bhansali, "When Students Get Stuck, Socratic Can Help," The Keyword, August 15, 2019, https://www.blog.google/outreach-initiatives/education/socratic-by-google/.

[36] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge, "A Neural Algorithm of Artistic Style," arXiv (2015), https://arxiv.org/abs/1508.06576; and Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur, "Supercharging Style Transfer," Google AI Blog, October 26, 2016, https://ai.googleblog.com/2016/10/supercharging-style-transfer.html.

[37] "DALL·E: Creating Images from Text," OpenAI, https://openai.com/blog/dall-e/.

[38] "Magenta," Google Research Team, https://research.google/teams/brain/magenta/.

[39] Jakub Konečný, Brendan McMahan, and Daniel Ramage, "Federated Optimization: Distributed Optimization beyond the Datacenter," arXiv (2015), https://arxiv.org/abs/1511.03575.

[40] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, et al., "Towards Federated Learning at Scale: System Design," arXiv (2019), https://arxiv.org/abs/1902.01046. For a less technical description on how the technology works, you can also check out this online comic description: "Federated Learning: Building Better Products with On-Device Data and Privacy by Default," Google AI, https://federated.withgoogle.com/.

[41] Sashank Reddi, Zachary Charles, Manzil Zaheer, et al., "Adaptive Federated Optimization," arXiv (2020), https://arxiv.org/abs/2003.00295; Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, et al., "Mime: Mimicking Centralized Stochastic Algorithms in Federated Learning," arXiv (2020), https://arxiv.org/abs/2008.03606; and James Bell, K. A. Bonawitz, Adrià Gascón, et al., "Cryptology ePrint Archive: Report 2020/704," in *CCS '20: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (New York: Association for Computing Machinery, 2020), https://eprint.iacr.org/2020/704.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al., "Attention Is All You Need," arXiv (2017), https://arxiv.org/abs/1706.03762; Jakob Uszkoreit, "Transformer: A Novel Neural Network Architecture for Language Understanding," Google AI Blog, August 31, 2017, https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html; Neil Houlsby and Dirk Weissenborn, "Transformers for Image Recognition at Scale," Google AI Blog, December 3, 2020, https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html; Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit, "Scaling Autoregressive Video Models," arXiv (2019), https://arxiv.org/abs/1906.02634; Anmol Gulati, James Qin, Chung-Cheng Chiu, et al., "Conformer: Convolution-Augmented Transformer for Speech Recognition," arXiv (2020), https://arxiv.org/abs/2005.08100; and AlphaFold Team, "AlphaFold: A Solution to a 50-Year-Old Grand Challenge in Biology," DeepMind, November 30, 2020, https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology.

[43] Kyle Lo, Iz Beltagy, Arman Cohan, et al., "Scibert," GitHub, last updated June 14, 2020, https://github.com/allenai/scibert.

[44] Maithra Raghu and Eric Schmidt, "A Survey of Deep Learning for Scientific Discovery," arXiv (2020), https://arxiv.org/abs/2003.11755.

[45] "MLSys–2020," Fifth Conference on Machine Learning and Systems, Santa Clara, California, April 11–14, 2020, https://mlsys.org/.

[46] Tim Kraska, Alex Beutel, Ed H. Chi, et al., "The Case for Learned Index Structures," arXiv (2017), https://arxiv.org/abs/1712.01208; Ani Kristo, Kapil Vaidya, Ugur Çetintemel, et al., "The Case for a Learned Sorting Algorithm," in *SIGMOD '20: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (New York: Association for Computing Machinery, 2020), https://dl.acm.org/doi/abs/10.1145/3318464.3389752; Samuel J. Kaufman, Phitchaya Mangpo Phothilimthana, Yanqi Zhou, et al., "A Learned Performance Model for Tensor Processing Units," arXiv (2020), https://arxiv.org/abs/2008.01040; Yanqi Zhou and Sudip Roy, "End-to-End, Transferable Deep RL for Graph Optimization," Google AI Blog, December 17, 2020, https://ai.googleblog.com/2020/12/end-to-end-transferable-deep-rl-for.html; and Martin Maas, David G. Andersen, Michael Isard, et al., "Learning-Based Memory Allocation for C++ Server

Workloads," *Proceedings of the 25th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (New York: Association for Computing Machinery, 2020), https://research.google/pubs/pub49008/.

[47] Vaswani et al., "Attention Is All You Need."

[48] Ibid., Table 4.

[49] Barret Zoph and Quoc V. Le, "Neural Architecture Search with Reinforcement Learning," arXiv (2016), https://arxiv.org/abs/1611.01578; Hieu Pham, Melody Guan, Barret Zoph, et al., "Efficient Neural Architecture Search via Parameters Sharing," *Proceedings of Machine Learning Research* 80 (2018), http://proceedings.mlr.press/v80/pham18a.html; Adam Gaier and David Ha, "Weight Agnostic Neural Networks," arXiv (2019), https://arxiv.org/abs/1906.04358; and Esteban Real, Sherry Moore, Andrew Selle, et al., "Large-Scale Evolution of Image Classifiers," arXiv (2017), https://arxiv.org/abs/1703.01041.

[50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv (2018), https://arxiv.org/abs/1810.04805.

[51] Carl Doersch and Andrew Zisserman, "Multi-Task Self-Supervised Visual Learning," arXiv (2017), https://arxiv.org/abs/1708.07860; and Sebastian Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," arXiv (2017), https://arxiv.org/abs/1706.05098.

[52] Vaswani et al., "Attention Is All You Need."

[53] Pham et al., "Efficient Neural Architecture Search via Parameters Sharing."

[54] "Artificial Intelligence at Google: Our Principles," Google AI, https://ai.google/principles/.