Learning Abstractions: A Conversation with Yann LeCun

James Manyika. Yann LeCun is widely considered one of the godfathers of the modern era of artificial intelligence. His pioneering research with Geoffrey Hinton and Yoshua Bengio on deep learning won them the 2018 Turing Award, considered the Nobel Prize for computer science, and in 2025, the three were awarded the Queen Elizabeth Prize for Engineering. Yann has long focused on foundational and scientific advances in AI, from biologically inspired convolutional neural networks and graph transformer networks to energy-based models for machine learning and world models. In this dialogue, we focus on foundational advances in AI, what he sees as progress to date, the limitations of current mainstream approaches, and what's needed to further advance AI and benefit science, including the role of scientists and the importance of open science.

Manyika. The last decade in artificial intelligence has been extraordinary. How would you characterize the progress? What has surprised you?

Yann LeCun. I'm actually going to go back fifteen years to when Geoffrey Hinton, Yoshua Bengio, and I were trying to rekindle interest in neural networks, which we rebranded as "deep learning." And we were struck by how fast it was adopted: within eighteen months of the first papers showing that deep learning could improve speech recognition, basically every mobile speech recognition system was using a neural network.[1] This was my first exposure to how quickly technology is widely adopted when it actually brings something to the table.

There was a similar phenomenon in 2013 when convolutional neural networks became the rage for image recognition. Within months they were widely deployed, and people got excited about the prospect of driving assistance and autonomous driving. Within a few years, we had automatic emergency braking systems in cars, all using convolutional neural networks. These systems have become mandatory in Europe, leading to a 40 percent reduction in frontal collisions.[2] This technology was saving lives, and it made me proud.

The year 2015 was more about natural language processing (NLP). Yoshua's lab made people pay attention to attention mechanisms, which direct deep-learning models to focus their attention on the most relevant data.[3] Then "Attention Is All You Need," which introduced the transformer architecture, surprised quite a few by building large networks as stacks of associative memory modules based on

https://doi.org/10.1162/DAED.a.972

so-called self-attention that compares inputs to each other.[4] I had been advocating for self-supervised learning (SSL) for many years, but I did not expect that its success would be primarily in NLP – though we had hints that "shallow" SSL worked for text embedding, with Word2vec from Google and fastText at Meta.[5] Then there was BERT from Google, which used transformers and attention instead of recurrent neural networks, which a lot of us suspected would not go very far.[6]

I wasn't particularly interested in NLP myself, but it was certainly a very useful thing to do. What I found surprising was that a simple form of temporal prediction works so well for text. I had been working on temporal prediction for video for at least ten years before that, hoping that if you train a system to predict what's going to happen in a video, it will understand how the world works : that the world is three dimensional and that objects move independently, for example. Some objects are inanimate and obey laws that are relatively easy to understand and make predictions about. Animated objects are much more difficult to predict. The hope and the whole idea of SSL is that by training a system through prediction, it can understand the underlying structure of what it is trained on. I was focusing on video because I believe that if we can do video, we can do everything. As it turns out, SSL from video is much harder than from text.

Manyika. As you think about where we are with the development of AI's capabilities, are there scientific disciplines you're most excited for AI to help make breakthrough advances ?

LeCun. Certainly. Materials science, chemistry, biochemistry, and proteins are major topics. In medicine, can we understand and cure chronic diseases or cancers ? Predicting protein structure is major progress, but the next step is designing proteins with particular properties.[7] Can we build predictive mesoscopic AI models for complex systems like organelles, cells, or pathways in cells that are too complex for a single human to understand ? The same applies to new materials and chemical compounds for improving energy storage (like batteries) or for catalysis (such as separating hydrogen from oxygen to store electrical energy as methane).

Importantly, the models that work well for proteins are really specialized ; they're not generic models. There is a hope that in the future we'll come up with models that are more generic, that can make wider sets of predictions. This includes efficient prediction in large systems controlled by partial differential equations (PDEs) that are too complex for us to solve accurately and quickly. In fluid dynamics, for instance, sometimes you only need to predict lift and drag or turbulent behaviors. These collective behaviors can be captured by training a neural net (essentially 3D convolutional networks that mirror PDEs) to make predictions at a coarse scale. This allows simulating much larger or turbulent systems to predict general properties rather than requiring a detailed simulation.

Shirley Ho at the Flatiron Institute demonstrated a striking example of this.[8] You can put the equations of general relativity into a simulator and simulate the first instants of the universe. Then you take that simulation and change the values of some of the parameters, like mass density or the properties of various fields. Then you run the simulation again to see if the result looks like our universe, if you get galaxies like those we see. That could validate or invalidate some theories about the nature of dark matter, for example. You can't do that simulation at the scale of the universe if you reduce it to the equations that we know. But if you run the simulation in a small cube, and you train a neural net to make a prediction of what goes on within that cube, but without requiring a detailed simulation of everything, then you can scale *that* up and perhaps run that simulation at a scale of the universe.

That is model prediction validation. The next step is design. If you have a predictive model, you can use inference by optimization to figure out the necessary interventions – action variables – to achieve a desired property for a chemical compound (for example, a specific energy separation of hydrogen from oxygen). This is the concept of a *causal world model* with very broad applications, such as in treating patients or controlling industrial processes and robots.

Manyika. What capabilities does AI need to help advance science and advance discovery?

LeCun. We can use machine learning to train predictive systems that can be phenomenological models. They're not necessarily models that explain *anything*, but they can make predictions that are sufficiently accurate and computationally much faster than a traditional reductionist model.

I think this is a crucial point and is what I am presently devoting all of my efforts to: devising AI systems that can find an abstract representation of the phenomenon and make predictions in that abstract representation space. This abstract representation eliminates a lot of details about the original observations. And that's a crucial point because LLMs (large language models) and other generative models are trained to predict *every detail* of the input. In language, it's not too much of a problem. You cannot predict exactly which word follows a sequence of words, but you can produce a probability distribution over words. That's easy because there's a finite number of possible words. But when you train the model to predict future frames in a video, you can't represent a useful distribution. You have to make predictions in an abstract representation space, not at the pixel level.

So a lot of people in the last few years instinctively said, "let's just tokenize the world." Let's take images from videos and cut them into little squares and turn that into a vector that doesn't look different from the one that represents a word, and feed this to a gigantic model to predict the next few frames. Frankly, it doesn't work that well. The reason why is that you simply cannot predict what's going to happen in a video at the pixel level. There are so many details that are just not in the

input. We don't know how to produce a probability distribution over all possible video frames because it's mathematically intractable. It's a problem people have struggled with for decades in statistical physics.

Instead, what we do as scientists is to find a representation of the input that eliminates all the details we cannot predict, and we make predictions in that representation space. That's not a generative architecture.

Manyika. Thinking about progress in AI, you've been vocal about the limitations you see in LLMs. What are those limitations, particularly in the context of applying AI to science?

LeCun. There are essentially three limitations. The first is that you cannot predict everything that takes place in the physical world or even in data collected from sensors. There is noise. Systems are stochastic – or we measure only partial information about the system. Trying to train a system to make an exact prediction in these conditions doesn't do anything particularly interesting. You have to let the system find the appropriate representation within which to make predictions. This is really what science is all about.

We could describe everything that takes place here between us in this conversation in terms of quantum field theory instead of, say, psychology. It would be completely impractical because we would need to measure the wave function of the universe within a cube that contains both of us. That's a pretty big thing. Then you'd need to run some insanely large quantum computations to make predictions. We never do this.

What we do is we invent abstractions that allow us to make predictions : particles, atoms, and then molecules ; or proteins, organelles, cells, organs, organisms, societies, and ecosystems. You have this whole hierarchy. What defines a branch of science is the level of abstraction at which you work to describe reality.

The second limitation is that you want the system to be able to learn the equivalent of a mental model or its environment, or what many people call a "world model." Given your idea of the state of the world at time t and an action you imagine taking, a world model will predict the state of the world at time $t + 1$ resulting from that action. You can think of it as a causal model : state, action, next state.

If you have such a world model, you can use it for science. The theory is represented by this predictive model. I can perform a virtual experiment : If I put the world in a particular state, and take a sequence of interventions, what is the predicted result ? Then I can do the experiment and verify in the real world what actually happens.

More important, if you have a world model, you can do reasoning and planning. You can specify a desired target state and use an optimization procedure to infer a sequence of interventions that will lead to that outcome state. We do this all the time

as humans, using our mental model of the world to plan a sequence of actions in a new situation that you've never faced before. This is what psychologists call System Two.[9]

With LLMs, you must train the system for all situations it may encounter, but a pure LLM doesn't really have the prediction and planning abilities I'm describing, of producing an output by optimization. There's no search in a pure LLM. Some, which are no longer pure LLMs, do reasoning through search by having the LLM produce lots of candidates for sequences of tokens, and then a separate neural net that rates them and picks the best one. A generate-and-select approach.

Manyika. In those cases, you use Monte Carlo tree search or more tournament-based systems in which you have multiple generations and search among them with a scoring function or other method of evaluating alternatives to then select among them. It's a form of search.

LeCun. That's right. You need two components. One neural net component predicts the most likely symbols that follow a sequence (actions or steps of reasoning). In games like chess and Go, you have a neural net that scores every possible move, and you only explore the ones that look the best. You need a component to generate candidate branches and a second component to evaluate them so that you can select the best ones and prune the exponentially growing tree of possibilities. The basic idea has been known since the 1960s, though training methods are key. Google has contributed a lot with AlphaGo and AlphaZero.[10]

For games, the objective is to win. You play a full game before you know if a sequence of moves was good. Reinforcement learning is all about how you attribute credit to individual moves going backward. If my value function at a given move ultimately proved not great, I need to propagate that outcome to tell the value function it was overestimating or underestimating. That's what reinforcement learning does.

In the context of LLMs, there's a similar but simpler process: the LLM produces lots of outputs and humans rate them – that's RLHF, or reinforcement learning from human feedback.[11] If the outputs are programs or mathematical proofs, you can run and check them. You can design the system to rate itself. But more often you need a human to say, "this one was good, this one was not good." Then the system adjusts parameters to give a higher score to the good outputs and a lower score for the bad ones.

This brings us to the third limitation. A lot of LLMs are post-trained using supervised learning, with experts providing answers to retrain the LLM. The problem is that as you increase the universe of questions, it gets ridiculously expensive. There's an infinite number of questions, and while you fine-tune for common ones, there's always a long tail of other questions and prompts not in your training set. Then you get nonsense. The system that appeared amazingly smart suddenly looks totally stupid. It appears not to know a super basic notion simply because it wasn't trained for a particular version of the question.

Manyika. You've been pursuing a different approach called JEPA (joint-embedding predictive architecture). Tell us more about it and why it's needed, especially for science.

LeCun. When training a generative architecture with self-supervised learning, you show the system a corrupted or transformed version of an input and train it to reconstruct the full input with all of its details. If you apply this to real data – which is multidimensional, continuous, and noisy, like video – you can't predict all the details. The trick is to learn an abstract representation that eliminates the details you cannot predict and perform predictions in that representation space.

Much of science is about choosing a good representation. Let me take a concrete example. Going back centuries, people were quite good at predicting the trajectory of planets and objects in the sky. It's a complex problem to determine a set of rules to describe how some planets appear to be going backward in the sky while others go forward. But if you change your representation of the problem and you know that Earth is just a planet rotating around the Sun, and the other planets also rotate around the Sun, it becomes obvious that some planets should sometimes go backward because you're going faster than them on your orbit. The representation is crucial. Another example is the ideal gas law, $PV = nRT$, which predicts the pressure of a gas as a function of temperature, assuming the molecules rarely bump into each other. This prediction is done in a representation space that eliminates almost everything about the underlying state of the system, like the position and velocities of all the gas molecules. We can't possibly measure them, so we ignore them. Everything we don't know about the system, we call entropy or heat. This whole idea – identifying what information is useful to make predictions and eliminating the stuff we cannot predict – is at the root of science and, I would argue, at the root of intelligence.

JEPA implements the idea of training a neural net to produce a representation of the input that may not contain all the details and simultaneously training a predictor in that representation space.[12] You make the system learn a representation that contains as much information as possible about the input but also allows you to make predictions: it does not contain all the stuff you cannot predict, like the motion of individual molecules in the gas example. We use an encoder to learn a representation space simultaneously with a predictor. The main point is: don't make predictions in input space, make predictions in representation space. The difficulty is preventing "collapse" in which the encoder ignores the input to produce a constant output, making prediction trivial.

Manyika. You used to work on energy-based models and entropy functions as ways to think about this. Is that still applicable here?

LeCun. The energy-based model view is the only way to interpret what goes on there. You can think of an autoencoder as an energy function that is the reconstruc-

tion error of an input. To feed it an input if the reconstruction error is zero or small means you've trained your autoencoder on this data. But if your reconstruction error is zero everywhere, that means it doesn't actually capture any interesting dependencies in the input. It just computes the identity function. You want an energy function that takes low energy around the data points that you train it on but takes *higher* values elsewhere. The energy becomes a contrast function that separates where your data are and where there are no data. Contrastive methods achieve this by generating points outside the manifold of data and pushing their energy up. Denoising autoencoders, like BERT-style NLP systems and masked autoencoders (MAE) for images/video, are examples. You take an input, corrupt it in some way to generate a point outside the manifold of data, and then you train your autoencoder to transform this corrupted version of your input into the original one.

An interesting version is diffusion models.[13] When you corrupt the inputs, instead of training the autoencoder to directly reconstruct the original, you train it only to take you closer to the original. You start from a data point, then you noisify it. Now you tell a neural net: I want you to point this way so that when they give you this noisy point, you're making it less noisy. You do this for all the points in the trajectory so that you get a neural net to learn a vector field that takes you back to the data manifold. Or you could do this by training an energy function and then computing the gradient of this energy function, which will take you back to the data manifold as well.

Manyika. What you're describing is trying to learn and build representations from observations, which is intuitively appealing in the context of science. But in some areas of science, there may be sparse data or limitations with respect to observables. Does that limit AI's potential to advance science in those areas?

LeCun. The hope is you might be able to train a generic encoder and predictor, self-supervised, with lots of data from different sources, as long as the input representation does not vary widely. That's the power of pretraining: you have a generic model that is pretrained, and if you have data for a particular experiment, you fine tune it. You may not have huge data to train on, but the pretraining should put you in a good starting position.

Manyika. If you're training on data from different domains or at different levels of abstraction and doing reasoning at very different levels, how do you deal with the compositionality challenge?

LeCun. I think we would need hierarchical models. And when you go up a layer, you eliminate some details about the system's state representation at the level below. You have a less accurate representation, but that allows for longer-term predictions of more complex emergent phenomena. For the same reason, we don't use chemistry to explain human behavior. If we want to make long-term predic-

tions of complex systems, we need to abstract away details and find abstract representations. This is how you would train those systems : lower levels make short-term, detailed predictions. Higher levels are trained to make longer-term predictions for more complex systems. They eliminate more information to make those predictions. Deciding which level to use for each problem is a different story.

Manyika. One of the distinctions you've been making is between what you term AMI (advanced machine intelligence) and AGI (artificial general intelligence). Can you explain and contrast these concepts ?

LeCun. First, I must say that there is no question in my mind that, at some point in the future, machines will be as intelligent as humans in all domains, including science.

But there is a question of vocabulary. I use the phrase *advanced machine intelligence* to mean the kind of intelligence that we observe in animals and humans. Not necessarily human-level, because even nonhuman animals are really smart. Current AI systems don't come close to that, at least in their understanding of the physical world and ability to plan.

Then there is human-level AI, which includes systems that have all the capabilities of humans, perhaps better in some domains. What people really mean by AGI is human-level AI, and there is a fallacy here : the hidden assumption that human intelligence is general. But human intelligence is not general at all. It's very specialized ; it's what evolution found was useful for our species to survive. We know we don't have general intelligence. Otherwise, you wouldn't be able to buy a $30 gadget that beats you at chess, right ?

We're terrible at a lot of tasks that computer science has shown over the last seven or eight decades can be done by computers better than us. It used to be that you could have a career as a mathematician by essentially filling up logarithm tables. Now we have computers that do this much better than we can. It used to be common in science fiction novels, like Isaac Azimov's *I, Robot*, that the way you test the intelligence of a robot is to ask it to compute an integral symbolically.[14] We can do this with computers now. It turns out that it's hard for humans, but it's not that hard computationally. I prefer to label advanced intelligence using the phrase artificial super intelligence (ASI) because it doesn't make any assumptions about the nature of human intelligence.

Manyika. Are you saying that AGI, as long as we compare that to human intelligence, is really a specialized intelligence as opposed to a general one – that it's really a subset of all intelligences ?

LeCun. There can never be a completely general intelligence, for the same reason that in machine-learning theory, the "no free lunch" theorems say that there are only a small number of tasks that any learning system can learn efficiently.[15] It's also

related to some more philosophical, conceptual things, like the fact that any measure of complexity you come up with will say that almost everything is random, except for a tiny number of things that your notion of complexity measures as simple.

That's at the root of thermodynamics. Only a small number of states of a physical system are considered organized. Just because of statistical fluctuations, it's going to go from what we consider organized to disorganized because there are many more disorganized states. Because of our notion of what is simple, only a small number of things are simple. I think there is a similar argument for intelligence. There cannot be a system that can solve every problem in the world. It has to be specialized. That's the philosophical aspect of it. So it's the word *general* that I just don't agree with.

Manyika. Terminology aside, you've critiqued the idea of getting to more capable intelligence all the way up to ASI via LLMs due to limitations of their autoregressive feed-forward paradigm. Can you explain that, particularly with respect to AI for science?

LeCun. The basic paradigm of intelligent inference that people in classical AI have known for decades is that you look for a solution to a problem by *search*. You need an intelligent machine with two components: one to check if a proposed output is good – *Is this a solution to my equation? Is it the shortest path between cities?* – and another to perform the search for the best solution. But there's no single magic bullet for every problem.

It took some time for the early pioneers of AI to realize this was a problem. In the 1950s, Alan Newell and Herbert Simon came up with this program that they very modestly called the General Problem Solver.[16] It said that if I can formulate a problem in terms of a search, and I have a way of checking whether a proposed solution is a good one, then I can solve *any* problem. What they didn't realize is that the search problem is intractable most of the time. In the traveling salesman problem – in which you must find the shortest route through a group of cities, traveling through each only once and ending back at the city of origin – as you increase the number of cities, the complexity grows exponentially. That started the whole theoretical part of computer science, of complexity theory, when people realized that most problems that are interesting are actually intractable.

This idea that you produce an answer to a new problem by search is absolutely crucial, and LLMs don't do this. They produce autoregressively. There is no optimization procedure; there's no search in a search space. LLMs that produce lots of token sequences and rate which ones are best are using a very primitive form of this kind of search. That's the first essential thing. Second, this search should not be performed in token space or language, but instead in an abstract continuous representation space. That is the way humans operate and is relevant to AI in science.

Manyika. But to do search in representation space presupposes some kind of objective function. Yet for general intelligence or for science, we may not know what that is; we're often experimenting or simply exploring or following our curiosity. Does that suggest that because AI leaves out all that curiosity-driven work, there may be a limit to the use of these tools for science?

LeCun. I don't think it limits that at all, because what is curiosity? When you make a prediction about something that should happen as a consequence of an event or an action, there are two possibilities. If your prediction was wrong, your model is wrong; you need to adjust it and probably repeat the experiment.

But there is another thing that humans are capable of, and that I think future AI systems will be capable of, or should be capable of: predicting the reliability of a prediction. If there are certain actions for which your prediction of the outcome is uncertain, you should take the action, observe the result, and correct your model if your prediction was wrong. That is a form of curiosity. Infants learn about things like gravity around the age of eight months. Most six-month-olds haven't yet figured out that every object falls under gravity. But at ten months, infants have. If you put some eight-month-olds on highchairs with a bunch of toys, they will systematically throw the toys on the ground and watch them fall. They do the experiment and verify that gravity applies to everything. If you show them a helium balloon, a party balloon, and it floats, they are fascinated, because here is an object that violates the rule that they thought they learned.

That's where the notion of a world model comes in. There are two types: the unconditional and the conditional (or causal). In the unconditional type, you start with the state of the world at time t. The world is a dynamical system. Can I predict the state of the world at time $t + 1$? The longer the time horizon you want to run your prediction, the more difficult it is to make accurate, detailed predictions. What you can do is make predictions at a higher level of abstraction, where many details are not present. Then there are conditional models where, given the state of the world at time t and a hypothetical intervention, can I predict the state of the world at time $t + 1$? That's really what we do in experimental science. We have a system in a state, we do an experiment, and we look at the result. If our hypothesis is good, the prediction is accurate. If what our model predicts deviates from reality, our model needs adjustment. That's at the root of the process of science: finding a representation of reality that contains the relevant aspects of the system, doing an intervention, and predicting the next state. That's a causal model.

Manyika. The distinction you are suggesting here is that with the ability to probe, test, explore, or perturb the system, you get closer to building a causally derived sense of intelligence. Whereas one could argue that statistically derived intelligence tells you very little about the underlying nature and behavior of the system. Is that right?

LeCun. Yes. You can have observational predictive models that make predictions about how the world evolves, even without interventions. Astronomy is an example. It's not an experimental science; you can't actually influence the system you're observing. But you can still come up with models that are kind of causal. You can predict the trajectory of planets. The simplicity with which you can do this depends completely on the representation space that you choose. So predicting the trajectory of planets became a hell of a lot simpler once we figured out all the planets rotate around the Sun, including Earth. The crucial question of science is, can we come up with a good representation of reality that allows us to make predictions, and possibly predictions conditioned on actions? Any intelligent system needs to be able to do that.

Manyika. Given the advances in AI, and particularly if we go beyond human cognitive levels and AI systems come to understand more than we do, what are the implications for philosophy of science, how we do science, and the nature of scientific understanding?

LeCun. I think that question is not a new one. When we solved PDEs (partial differential equations) numerically with computers, did the computational fluid dynamics simulator understand physics better than we did? It can make a prediction and it's using an algorithm based on equations that humans came up with.

The next step AI enables is training a machine-learning system to make predictions from data without the manual step of reducing the process to equations. AI allows us to skip having to first build a model of reality that can then be computed. This is powerful because many phenomena in science are collective complex phenomena.

A pile of sand behaves in a particular way, and the theory for this is not entirely clear. The property of materials, particularly complex ones, cannot be directly derived from the elementary equations of quantum mechanics. It's just too complicated. Another example is the magic angle, 1.1 degrees, at which you rotate two stacked monolayers of carbon, called graphene, to form a superconductor. That's a collective phenomenon that is extremely difficult to explain. There are various properties of materials of this type that cannot be usefully reduced to a small number of equations from which you can derive this collective behavior. How does intelligence emerge from neurons in interaction? That's a philosophical question of how a super complex property like intelligence can emerge from a large number of relatively simple elements in interaction, but that's a pretty high-level thing. At a lower level are questions of how life emerges from the interaction between proteins. This transition is what has baffled scientists for a long time: the transition from the microscopic to the mesoscopic. This is where interesting things happen, like life, for example.

So now there's a new way of doing science, which is neither completely qualitative and observational nor reductionist, but is a data-driven, AI-powered phenomenological model that may allow us to bridge the gap between microscopic and macroscopic.

Manyika. In the future, do you imagine AI doing science and making discoveries by itself, or will it always be a tool for scientists to use for discovery?

LeCun. I think it's going to be a tool for a long time. There are probably some domains in which you can let the thing run for a long time and come up with something meaningful, and maybe it will be able to evaluate whether the observed behavior or the model is interesting. There could be some automation in some domains, but I think the relationship of scientists with AI systems for a long time will be the same as between a PhD advisor and a PhD student. Except that the goal of the latter relationship is for the PhD student to eventually be autonomous. "There is something interesting to explore in this area. Why don't you work on this and then come back when you get something interesting?" This is the difference between directing the research of a team and doing the research yourself.

AI will essentially operate as a staff of diligent research assistants. Even if these AI systems are better than us at solving individual problems or practicing scientific research – and it's great to work with people who are smarter than you! – we will still be their boss. We will still tell them what to work on, because ultimately, we use these tools to help ourselves as a species or society. Directing what AI systems produce is a role we'll preserve.

Manyika. In that world, could we be limited by the scientist who's conducting this orchestra of intelligent systems, whether they're PhD students or AI systems? Does that present some sort of ceiling to the science that we can do?

LeCun. No, I don't think that presents a ceiling, or not any more than an advisor is a ceiling to what a PhD student can produce. Sometimes PhD students come up with a cool idea the advisor hadn't thought about. You put them on the right track, but they redirect things a bit. We think about the entire process of science, so when individual scientists in an industry lab, for example, work on a topic, there is a little bit of influence from their managers, but not a huge amount, if it's a self-respecting research lab.

If there is too much influence on the scientists working in your lab, you are only going to get expected results. If you want breakthroughs, you have to give people a very long leash and basically get out of their way. That's a bit of the magic I observed at Bell Labs and tried to produce at FAIR (Meta's Fundamental AI Research): hire the best people, give them the means to succeed, orient them in a good direction, and get out of the way.

Manyika. Does that also imply that to progress AI and science, once they are capable enough, we should just "hire" the best AI systems, give them direction, and get out of the way*?*

LeCun. Only if you think we've already figured out all the main ideas and all we need to do is scale up and engineer the thing better. Frankly, a lot of people in AI are currently thinking that we already have the proper paradigm, and we just need to scale it up. I famously don't believe this. I believe that we need a few conceptual breakthroughs before we get to something like AMI or ASI. We need to do research that is not as corralled as the current default mode in the industry. Academia tends to be less directive, but there is still some direction and some constraints on the work you can do in that you need to raise money to pay for your students and postdocs, which generally comes from a particular program that NSF or some other agency decided was critical.

Manyika. You've been a very strong proponent of open science – for example, sharing preprints and open-source code. How does openness, in science and in AI, accelerate progress?

LeCun. The very process of science is centered around communication. You have an idea, you've done an experiment, you got some result, and you publish it. The reason you need to publish it is because you need other people in the community to be able to challenge the idea, try to verify the result themselves, and see if it's possible to do better. The progress of science relies on communication between scientists and therefore on publication.

If you do some groundbreaking research and you don't talk about it, it's as if you never thought about it. If you keep it a secret, *maybe* some years down the line you'll be able to build a product out of it. But how do you convince anyone that your idea is good and that they should fund you to realize your idea? You need to publish it. You need to explain: "Here is how it works and I tried it on this dataset, but I'm a poor PhD student in a university somewhere in a far corner of the world, and I can't test this on a big dataset without support." Here is the idea, someone else will have the means to bring it to something real, and then at some point it may spur some investment from industry and end up changing an industry or creating a new one. It's a long process, and people don't realize that process can take five, ten, twenty, thirty years. This was the case for deep learning.

By 2012, posting AI preprints on arXiv had become standard procedure; peer review and official publication would follow months later. The arXiv submission of the paper "Attention Is All You Need" had something like seven hundred citations before it was even presented at conference. I publicly claimed that if an advance is not published, it's not research, because you can fool yourself. I've seen this in industry research labs where ideas bubble up the hierarchy and appear amazing, but no one challenges or verifies them. Peer review is imperfect and feels brutal, but it can stop you from fooling yourself. The whole process of science is to prevent you from fooling yourself. It's really what the scientific method is all about.

The practice of open source has accelerated progress in AI in unimaginable ways. Tools like TensorFlow, PyTorch, and Jax have hugely enabled progress, and the fact that everyone publishes their code means you can reproduce and extend the work of others. That is behind the incredible speed of progress.

Manyika. For most scientists, the benefits are clear. But some in industry and government make arguments against open science based on competitiveness and intellectual property, as well as safety/geopolitical concerns. How do you respond?

LeCun. To the question of competition : There's a spectrum from blue sky (basic) research to engineering applications. Publications are more useful near the blue sky end of the spectrum. As you move toward application, the work requires know-how and details specific to your infrastructure. So there is a point at which if you want to keep your competitive edge, you don't want to publish. But you wouldn't publish anyway if you can't really communicate your application results through a classical paper.

The ability to publish also attracts the best scientists. You'll have a hard time hiring a self-respecting scientist if you forbid them from discussing their work. This is why famously secretive companies struggle to have top-level AI research groups. Also, the not-fooling-yourself aspect of science means higher-quality research generally. If scientists are mandated to publish, they will follow better methodology to survive peer review. And there's another consideration : the commercial impact of research might be several years down the line. If you want scientists to be motivated, they can't be motivated by product impact alone, because the probability of that happening can be small, even if theirs is a good idea. It could be twenty years down the line. Scientists have to score something sooner. Presenting a paper at a conference and earning citations are ways of scoring. And in industry, publications are a way for management to evaluate a piece of research. It is notoriously difficult to evaluate the potential long-term impact of a piece of research. Relying on the external research community can be a good way to avoid mistakes. If a paper from your lab wins an award at ICLR (International Conference on Learning Representations) or CVPR (Conference on Computer Vision and Pattern Recognition), you should probably pay attention to it internally. Same thing if an arXiv preprint earns two hundred citations in a few months ; or if that open-source package your product groups never paid attention to got ten thousand stars on GitHub and is used by the whole industry except you.

Now, about the defense and geopolitical question. On the spectrum from research to product or deployment, you could raise the threshold for making research public for geopolitical or safety reasons rather than for acceleration of scientific research. But you pay a price. The more secretive you become, the more you slow yourself down. And if you can do research and solve a problem in secret, then that problem probably was not that difficult to begin with. You might imagine having a monop-

oly on good ideas to solve your problem, but if it's a hard problem, it will take the entire scientific community to take interest and exchange ideas.

The problem of ASI or AMI is of that nature. It's still a scientific question. It's not a question of scaling up or engineering resources. It's a scientific question, and nobody has a monopoly on good ideas, which is why I think some of the outfits that are claiming that they'll come up with AGI, or whatever they call it, within a few years, with a small team of ten people working in secret, that's complete BS. That just can't possibly happen. It's one of the most challenging scientific and technological problems of our times. You won't solve it with a handful of people working in secret; that only happens in bad sci-fi movies.

Manyika. As you look to the future, what are the big problems in science that you hope AI will have helped to solve?

LeCun. There are three major questions in science. What's the universe made of? What's life all about? And how does the brain work? Pretty much all of science falls into one of those three categories. I think the promise of AI for science is enormous.

ABOUT THE AUTHORS

Yann LeCun is Executive Chairman of Advanced Machine Intelligence (AMI) Labs and the Jacob T. Schwartz Professor of Computer Science, Data Science, Neural Science, and Electrical and Computer Engineering at New York University. He previously served as Chief AI Scientist for Meta's Fundamental AI Research (FAIR) team. He has recently published in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, *International Conference on Learning Representations* 2025, and *Advances in Neural Information Processing Systems*.

James Manyika, a Member of the American Academy since 2019, is SVP at Google-Alphabet and President for Research, Labs, Tech & Society.

ENDNOTES

[1] 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): 82–97, https://doi.org/10.1109/MSP.2012.2205597.

[2] Brian Fildes, Michael Keall, Niels Bos, et al., "Effectiveness of Low Speed Autonomous Emergency Braking in Real-World Rear-End Crashes," *Accident Analysis & Prevention* 81 (2015): 24–29, https://doi.org/10.1016/j.aap.2015.03.029.

[3] Dzmitry Bahdanau, Cho Kyunghyun, and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv (2014 ; rev. 2016), https ://arxiv.org /abs/1409.0473.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al., "Attention Is All You Need," arXiv (2017), https ://arxiv.org/abs/1706.03762.

[5] 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 ; Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, "Enriching Word Vectors with Subword Information," arXiv (2016 ; rev. 2017), https ://arxiv.org /abs/1607.04606 ; and Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov, "Bag of Tricks for Efficient Text Classification," arXiv (2016 ; rev. 2017), https ://arxiv.org/abs/1607.01759.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT : Pre-Training of Deep Bidirectional Transformers for Language Understanding," arXiv (2018 ; rev. 2019), https ://arxiv.org/abs/1810.04805.

[7] John Jumper, Richard Evans, Katherine Tunyasuvunakool, et al., "Highly Accurate Protein Structure Prediction with AlphaFold," *Nature* 596 (7873) (2021) : 583–589, https ://doi .org/10.1038/s41586-021-03819-2.

[8] Siyu He, Yin Li, Yu Feng, et al., "Learning to Predict the Cosmological Structure Formation," *Proceedings of the National Academy of Sciences* 116 (28) (2019) : 13825–13832, https ://doi.org/10.1073/pnas.1821458116.

[9] Daniel Kahneman, *Thinking, Fast and Slow* (Farrar, Straus and Giroux, 2011).

[10] David Silver, Aja Huang, Chris J. Maddison, et al., "Mastering the Game of Go with Deep Neural Networks and Tree Search," *Nature* 529 (7587) (2016) : 484–489, https ://doi .org/10.1038/nature16961 ; and David Silver, Thomas Hubert, Julian Schrittwieser, et al., "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play," *Science* 362 (6419) (2018) : 1140–1144, https ://doi.org/10.1126/science.aar6404.

[11] Long Ouyang, Jeffrey Wu, Xu Jiang, et al., "Training Language Models to Follow Instructions with Human Feedback," arXiv (2022), https ://arxiv.org/abs/2203.02155.

[12] Mahmoud Assran, Quentin Duval, Ishan Misra, et al., "Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture," arXiv (2023), https ://arxiv .org/abs/2301.08243.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models," arXiv (2020), https ://arxiv.org/abs/2006.11239.

[14] Isaac Asimov, *I, Robot* (Gnome Press, 1950).

[15] David H. Wolpert and William G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation* 1 (1) (1997) : 67–82, https ://doi .org/10.1109/4235.585893.

[16] Allen Newell, John C. Shaw, and Herbert A. Simon, "Elements of a Theory of Human Problem Solving," *Psychological Review* 65 (3) (1958) : 151–166, https ://doi.org/10.1037 /h0048495.