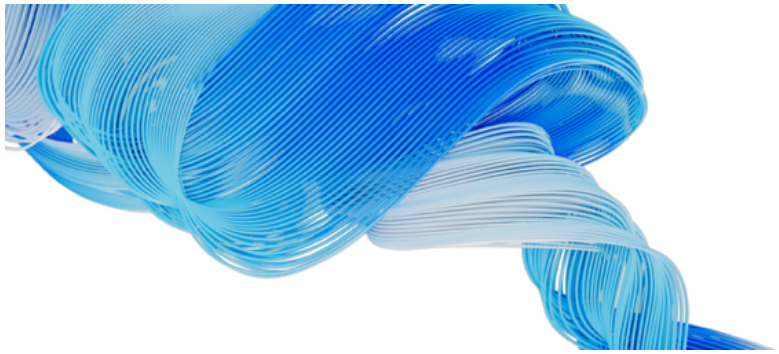
NLP | Computer Vision

Yann LeCun on a vision to make AI systems learn and reason like animals and humans

February 23, 2022



Watch the Meta AI Inside the Lab event here.

For all the remarkable recent progress in AI research, we are still very far from creating machines that think and learn as well as people do. As Meta AI's Chief AI Scientist Yann LeCun notes, a teenager who has never sat behind a steering wheel can learn to drive in about 20 hours, while the best autonomous driving systems today need millions or billions of pieces of labeled training data and millions of reinforcement learning trials in virtual environments. And even then, they fall short of human's ability to drive a car reliably.

What will it take to build AI that approaches human-level capabilities? Is it simply a matter of more data and bigger AI models?

As part of Meta AI's Inside the Lab event on February 23, 2022, LeCun is sketching an alternate vision for building human-level AI. LeCun proposes that the ability to learn "world models" — internal models of how the world works — may be the key.

Meta AI is sharing some of LeCun's ideas in brief here, including his proposal for a modular, configurable architecture for autonomous intelligence, as well as key challenges the AI research community must address to build such a system. We typically share the results of our

research — by publishing papers, code, and data sets, as well as blog posts — when they are completed. But in keeping with Meta AI's open-science approach, we are taking this opportunity to preview our research vision and ideas in the hope that it spurs discussion and collaboration among AI researchers. The simple fact is that we will need to work together to solve these extraordinarily challenging, exciting problems.

We plan to share more details on LeCun's vision in an upcoming position paper.

AI that can model how the world works

"Human and nonhuman animals seem able to learn enormous amounts of background knowledge about how the world works through observation and through an incomprehensibly small amount of interactions in a task-independent, unsupervised way," LeCun says. "It can be hypothesized that this accumulated knowledge may constitute the basis for what is often called common sense."

And common sense can be seen as a collection of models of the world that can guide on what is likely, what is plausible, and what is impossible.

This allows humans to plan effectively in unfamiliar situations. That teen driver may not have driven over snow before, for example, but he (hopefully) knows that snow can be slippery and send his car into a skid he drives too aggressively.

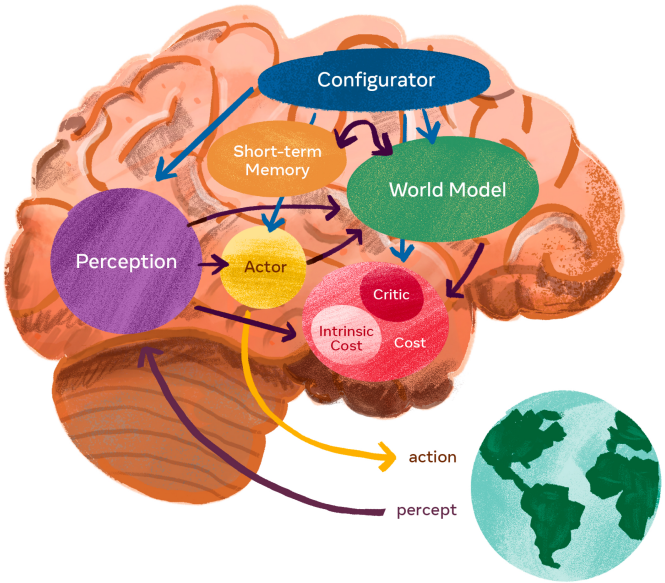
Common sense knowledge allows animals not just to predict future outcomes but also to fill in missing information, whether temporally or spatially. When a driver hears the sound of metal smashing together nearby, he knows immediately that there's been an accident — even without seeing the vehicles involved.

The idea that humans, animals, and intelligent systems use world models

The idea that humans, animals, and intelligent systems use world models goes back many decades in psychology and in fields of engineering such as control and robotics. LeCun proposes that one of the most important challenges in AI today is devising learning paradigms and architectures that would allow machines to learn world models in a self-supervised fashion and then use those models to predict, reason, and plan. His outline regroups ideas that have been proposed in various disciplines, such as cognitive science, systems neuroscience, optimal control, reinforcement learning, and “traditional” AI, and combines them with new concepts in machine learning, such as self-supervised learning and joint-embedding architectures.

Proposing an architecture for autonomous intelligence

LeCun proposes an architecture composed of six separate modules. Each is assumed to be differentiable, in that it can easily compute gradient estimates of some objective function with respect to its own input and propagate the gradient information to upstream modules.



A system architecture for autonomous intelligence. The configurator gets inputs from other modules, but we have omitted those arrows in order to simplify the diagram.

- The configurator module performs executive control: Given a task to be executed, it preconfigures the perception module, the world model, the cost, and the actor for the task at hand, possibly by modulating the parameters of those modules.
- The perception module receives signals from sensors and estimates the current state of the world. For a given task, only a small subset of the perceived state of the world is relevant and useful. The configurator module primes the perception system to extract the relevant information from the percept for the task at hand.
- The world model module constitutes the most complex piece of the architecture. Its role is twofold: (1) to estimate missing information about the state of the world not provided by perception, and (2) to predict plausible future states of the world. The world model may predict natural evolutions of the world or predict future world states resulting from a sequence of actions proposed by the actor module. The world model is a kind of simulator of the part of the world relevant to the task at hand. Since the world is full of uncertainty, the model must be able to represent multiple possible predictions. A driver approaching an intersection may slow down in case another car approaching the intersection doesn't stop at the stop sign.
- The cost module computes a single scalar output that predicts the level of discomfort of the agent. It is composed of two submodules: the intrinsic cost, which is hard-wired and immutable (not trainable), and computes the immediate discomfort (such as damage to the agent, violation of hard-coded behavioral constraints, etc.), and the critic, which is a trainable module that predicts future values of the intrinsic cost. The ultimate goal of the agent is to minimize the intrinsic cost over the long run. "This is where basic behavioral drives and intrinsic motivations reside," LeCun says. So it will factor in

intrinsic costs, such as not wasting energy, as well as costs specific to the task at hand. “Because the cost module is differentiable, the gradient of the cost can be back-propagated through the other modules for planning, reasoning, or learning.”

- The actor module computes proposals for action sequences. “The actor can find an optimal action sequence that minimizes the estimated future cost, and output the first action in the optimal sequence, in a fashion similar to classical optimal control,” LeCun says.

- The short-term memory module keeps track of the current and predicted world state, as well as associated costs.

World model architecture and self-supervised training

The centerpiece of the architecture is the predictive world model. A critical challenge with constructing it is how to enable it to represent multiple plausible predictions. The real world is not entirely predictable: There are many possible ways a particular situation can evolve, and there are many details of a situation that are irrelevant to the task at hand. I may need to anticipate what cars around me are going to do while I drive, but I don’t need to predict the detailed position of individual leaves in the trees that are near the road. How can a world model learn abstract representations of the world so that important details are preserved, irrelevant details are ignored, and predictions can be performed in the space of abstract representations?

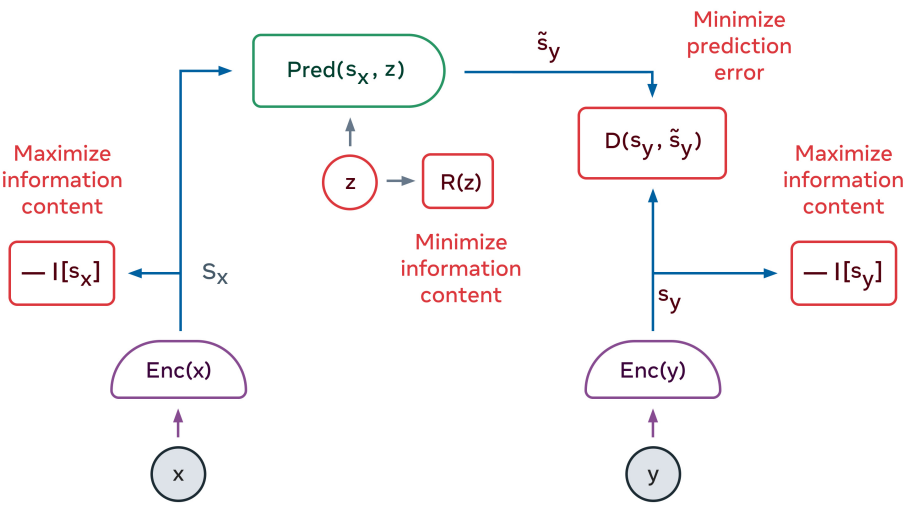
One key element of a solution is the Joint Embedding Predictive Architecture (JEPA). The JEPA captures the dependencies between two inputs, x and y. For example, x could be a segment of video, and y the next segment of the video. Inputs x and y are fed to trainable encoders that extract abstract representations of them, s_x and s_y. A predictor module is trained to predict s_y from s_x. The predictor may use a latent variable, z, to

represent information present in s_y that is not present in s_x. The JEPA handles uncertainty in predictions in two ways: (1) The encoder may choose to drop information about y that is difficult to predict, (2) the latent variable z, when varied over a set, will cause the prediction to vary over a set of plausible predictions.

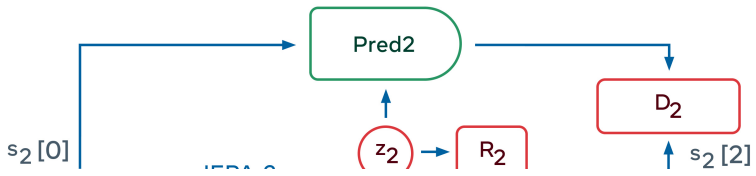
How do we train a JEPA? Until recently, the only approach would have been to use contrastive methods, which consist of showing examples of compatible x and y, together with numerous examples of x and incompatible y's. But this is rather impractical when the representations are high-dimensional. An alternative training strategy has emerged in the last two years: regularized methods. When applied to JEPA, the method uses four criteria:

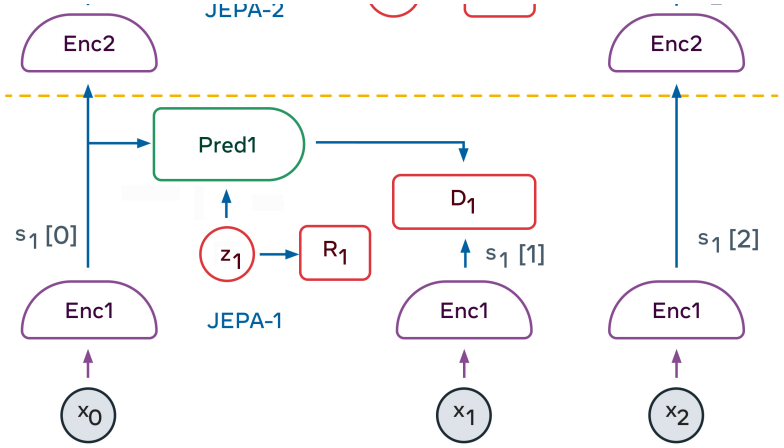
1. Make the representation of x maximally informative about x
2. Make the representation of y maximally informative about y
3. Make the representation of y maximally predictable from the representation of x
4. Make the predictor use as little information as possible from the latent variable to represent uncertainty in the prediction.

These criteria can be translated into differentiable cost functions in various ways. One way is the VICReg method, which stands for Variance, Invariance, Covariance Regularization. In VICReg, the information content of the representations of x and y are maximized by maintaining the variances of their components over a threshold and by making these components as independent of each other as possible. Simultaneously, the model tries to make the representation of y predictable from that of x. Additionally, the information content of the latent variable is minimized by making it discrete, low-dimensional, sparse, or noisy.



The beauty of the JEPA is that it naturally produces informative abstract representations of the input that eliminate irrelevant details and with which predictions can be performed. This enables JEPAs to be stacked on top of one another so as to learn representations with higher levels of abstraction that can perform longer-term prediction. For example, a scenario can be described at a high level as “a cook is making crêpes.” One can predict that the cook will fetch flour, milk, and eggs; mix the ingredients; ladle batter into a pan; let the batter fry; flip the crêpe; and repeat. At a lower level, pouring a ladle involves scooping some batter and spreading it around the pan. This continues all the way down to the precise trajectories of the chef’s hands millisecond by millisecond. At the low level of hand trajectories, our world model can only make accurate predictions in the short term. But at a higher level of abstraction, it can make long-term predictions.





The Hierarchical JEPA can be used to perform predictions at several levels of abstraction and several time scales. How can it be trained? Largely by passive observation, and less often by interaction.

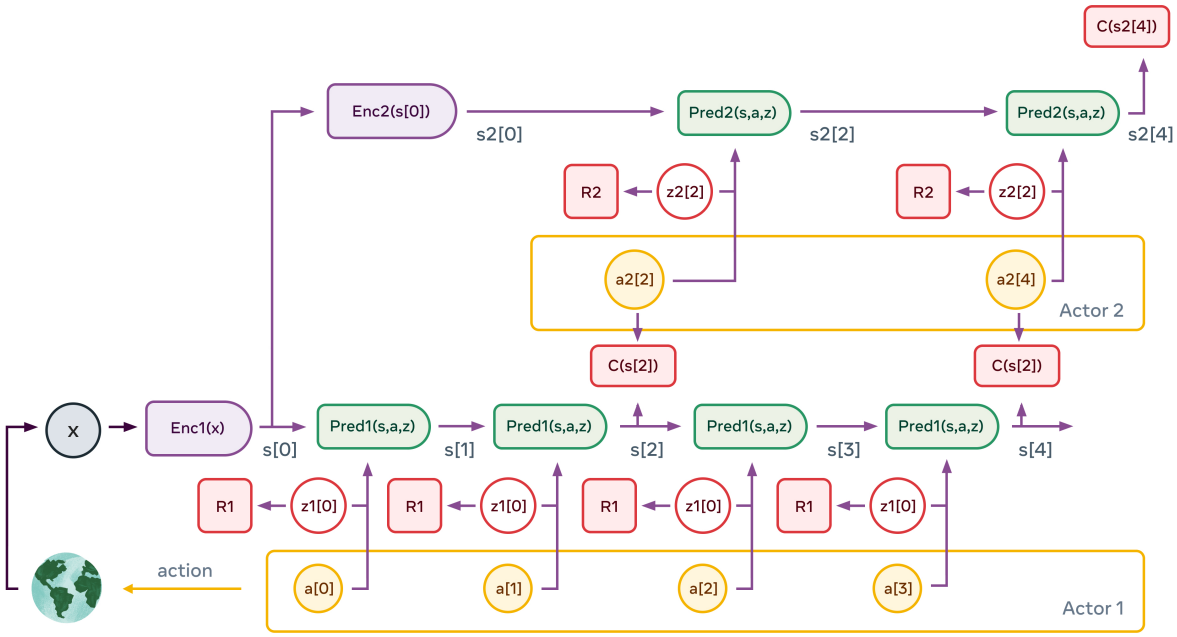
A baby learns how the world works largely by observation in the first few months of life. She learns that the world is three-dimensional, that some objects are in front of others, that when an object is occluded it still exists. Eventually, around nine months of age, babies learn intuitive physics — for example, that unsupported objects fall through gravity.

The hope is that a hierarchical JEPA could learn how the world works by watching videos and interacting with its environment. By training itself to predict what will happen in the video, it will produce hierarchical representations of the world. By taking actions in the world and observing the result, the world model will learn to predict the consequences of its actions, which will allow it to reason and plan.

A perception-action episode

With a Hierarchical JEPA properly trained as a world model, an agent could perform hierarchical planning of complex actions, decomposing a

complex task into a sequence of less complex and less abstract subtasks, all the way down to low-level actions on effectors.



A typical perception-action episode would go as follows. The diagram illustrates the situation for a two-level hierarchy. The perception module extracts a hierarchical representation of the state of the world (s1[0]=Enc1(x) and s2[0]=Enc2(s[0]) in the diagram). Then, the second-level predictor is applied multiple times to predict future states, given a sequence of abstract actions proposed by the second-level actor. The actor optimizes the second-level action sequence so as to minimize the overall cost (C(s2[4]) in the diagram). This process is akin to Model-Predictive Control in optimal control. The process is repeated for multiple drawings of the second-level latent variables, which may produce different high-level scenarios. The resulting high-level actions do not constitute real actions but merely define constraints that the lower-level state sequence must satisfy (e.g., are the ingredients properly mixed?). They really constitute subgoals. The entire process is repeated at the lower level: running the lower-level predictor, optimizing the low-level

action sequence to minimize the intermediate costs coming from the upper layer, and repeating the process for multiple drawings of the low-level latent variables. Once the process is complete, the agent outputs the first low-level action to the effectors, and the whole episode can be repeated.

If we are successful at building such a model, all the modules would be differentiable, so that this whole action optimization process could be performed using gradient-based methods.

Moving closer to human-level intelligence in AI

LeCun's vision requires much deeper exploration than is possible in a brief blog post, and many difficult challenges lie ahead. One of the most interesting and difficult of these is instantiating the details of the architectures and training procedures for the world model. In fact, it could be argued that training world models constitutes the main challenge toward real progress in AI over the next decades.

But many other aspects of the architecture are still to be defined, including how precisely to train the critic, how to construct and train the configurator, and how to use the short-term memory to keep track of the world state and to store a history of world states, actions, and associated intrinsic cost to tune the critic.

LeCun and other Meta AI researchers look forward to exploring these in the months and years ahead, as well as exchanging ideas and learning from others in the field. Creating machines that can learn and understand as effectively as humans is a long-term scientific endeavor — and one with no guarantees of success. But we are confident that fundamental research will continue to produce a deeper understanding of both minds and machines, and will lead to advances that benefit everyone who uses AI.

Watch the Meta AI Inside the Lab event here.

Search AI content

Our approach

Research

Meta AI

Latest news

Foundational models