A Path Towards Autonomous Machine Intelligence

Yann LeCun

https://openreview.net/pdf?id=BZ5a1r-kVsf

Summary by Michael Wollowski

AI

- “Although computers cannot think, machines can now mimic functions such as memory and learning.” [THE NOBEL PRIZE IN PHYSICS 2024 Popular Science Background]
- “Inspired by biological neurons in the brain, ANNs are large collections of […] nodes, connected by […] weighted couplings, which are trained to perform certain tasks rather than asked to execute a predetermined set of instructions.” [Scientific Background to the Nobel Prize in Physics 2024]

LeCun[9] on LMMs

- “Not anywhere close to human intelligence.”
- “Thinking, planning, how the world works is very limited.”
- “AI systems need to plan their actions so as to optimize a series of objectives.”
- And yes, LeCun and others are working on developing that technology.
- Once realized, those systems will likely be characterized as AGI, Artificial General Intelligence.

9) Harry Stebbings interview of Yan LeCun. https://www.youtube.com/watch?v=OgWaowYiBPM&t=27s 2023.

High Level Machine Intelligence

- “High-level machine intelligence (HLMI) is achieved when unaided machines can accomplish every task better and more cheaply than human workers. […] Think feasibility, not adoption.”

Source: Katja Grace et al. THOUSANDS OF AI AUTHORS ON THE FUTURE OF AI

About this Presentation

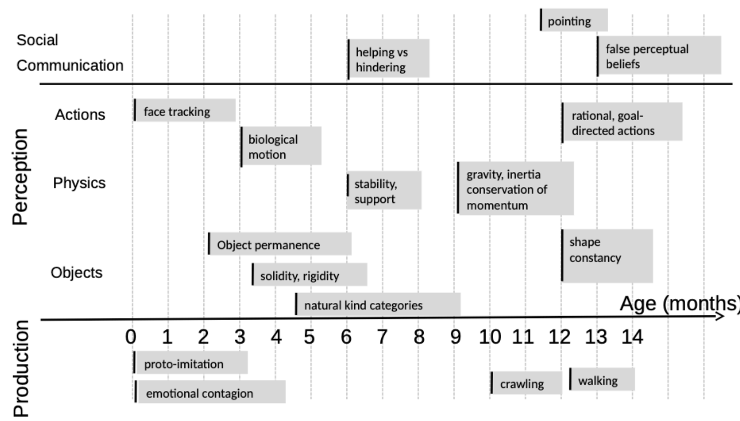
- It summarizes, without much comment a position paper by Yann LeCun.
- He states that it is "… a position paper expressing my vision for a path towards intelligent machines that learn more like animals and humans, that can reason and plan, and whose behavior is driven by intrinsic objectives, rather than by hard-wired programs, external supervision, or external rewards."

Scaling Laws

- The performance of large language models has shown to be mainly determined by 3 factors:
  - model size (the number of parameters),
  - dataset size (the amount of training data), and
  - the number of iterations used for training.
- We can improve a model by adding parameters (adding more layers or having wider contexts or both), by training on more data, or by training for more iterations.
- The relationships between these factors and performance are known as *scaling laws*.
- LeCun believes there is a limit to what can be achieved by scaling.

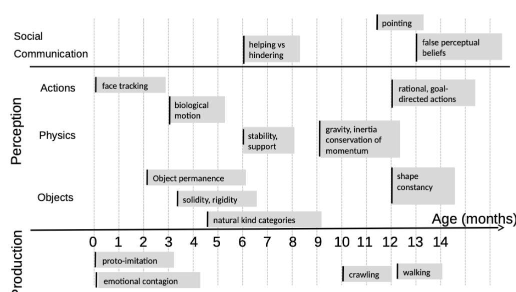
Acquisitions of Concepts about the World

- Learning from the bottom-up
- Turing (and others) argued for that.



Acquisitions of Concepts about the World

- Abstract concepts, such as gravity and inertia, are acquired on top of less abstract concepts, such as object permanence.
- Much of this knowledge is acquired mostly by observation.
- Very little direct intervention, particularly in the first few weeks and months.



System Architecture

- The components and structure of the proposed approach.
- It is an agent architecture.

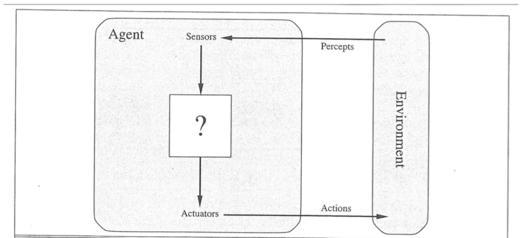
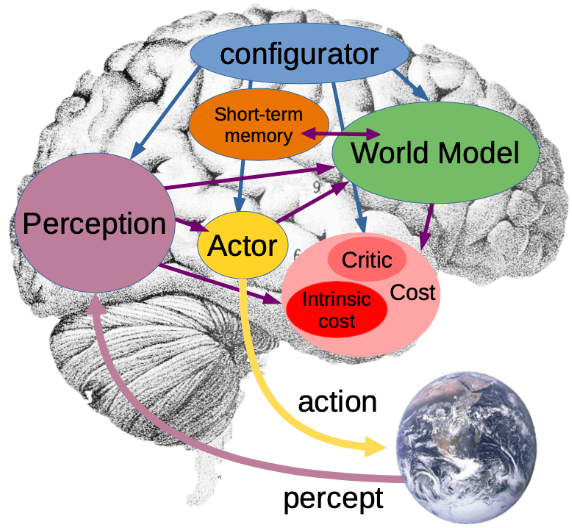


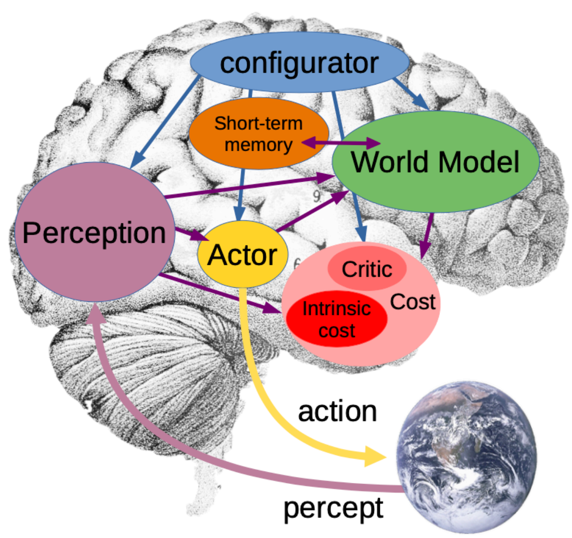
Figure 2.1 Agents interact with environments through sensors and actuators.

Figure 2.1 source: Russell and Norvig. AIMA.



System Architecture

Configurator module takes inputs (not represented for clarity) from all other modules and configures them to perform the task at hand. **Perception module** estimates the current state of the world. **World model** module predicts possible future world states as a function of imagined actions sequences proposed by the actor.

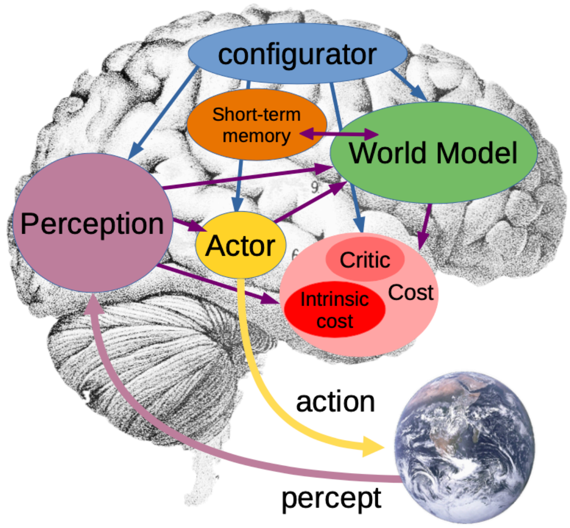


System Architecture

Cost module computes a single scalar output called "energy" that measures the level of discomfort of the agent.
It is composed of two sub-modules, the

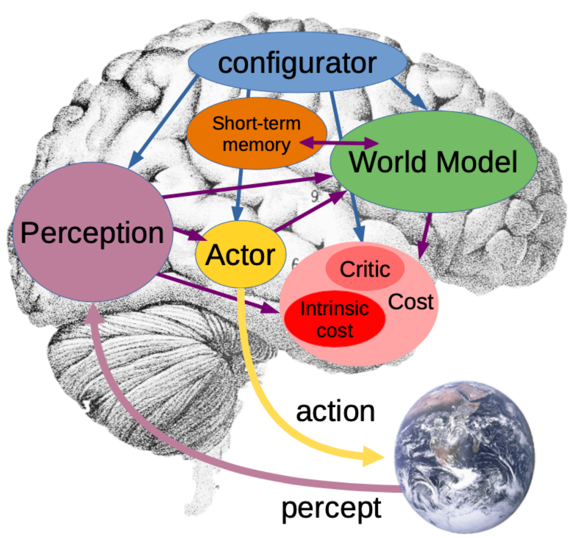
- *intrinsic cost*, which is immutable and
- the *critic*, a trainable module that predicts future values of the intrinsic cost.

Short-term memory module keeps track of the current and predicted world states and associated intrinsic costs.



System Architecture

Actor module computes proposals for action sequences.
World model and the critic compute the possible resulting outcomes.
The actor can find an optimal action sequence that minimizes the estimated future cost, and output the first action in the optimal sequence.



System-1 and System-2 Thinking

- Two possible modes that the model can employ for a perception-action episode.

1. **No complex reasoning**. Produces an action directly from the output of the perception and a possible short-term memory access.

   We call it "Mode-1", by analogy with Kahneman's "System 1".

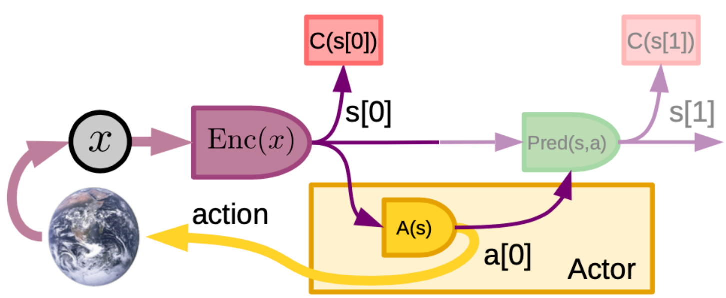
2. **Reasoning and planning** through the world model and the cost.

   We call it "Mode-2" by analogy to Kahneman's "System 2".

- We use the term "reasoning" in a broad sense here to mean constraint satisfaction (or energy minimization).

Mode-1 Perception-Action Episode

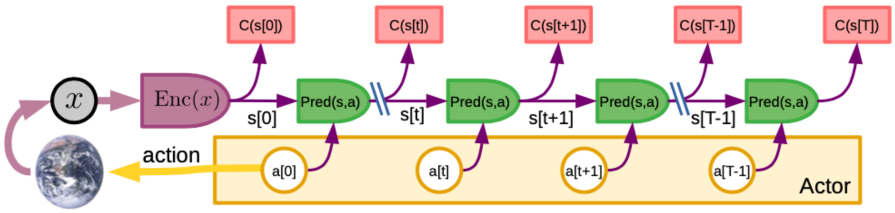
- The perception module estimates the state of the world s[0] = Enc*(*x*).
- The actor directly computes an action, or a short sequence of actions, through a policy module a[0] = A(s[0]).



- This reactive process does not make use of the world model nor of the cost.
- The cost module computes the energy of the initial state f[0] = C(s[0]) and stores the pairs (s[0], f[0]) in the short-term memory.

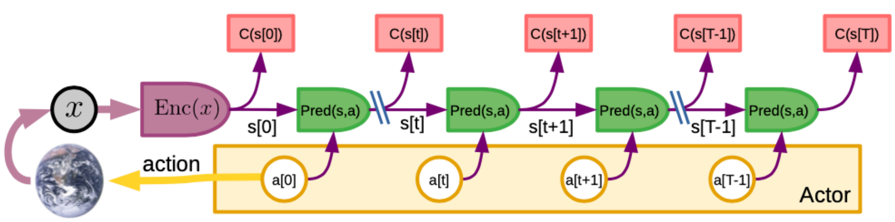
*) Enc stands for Encoder

Mode-2 Perception-Action Episode



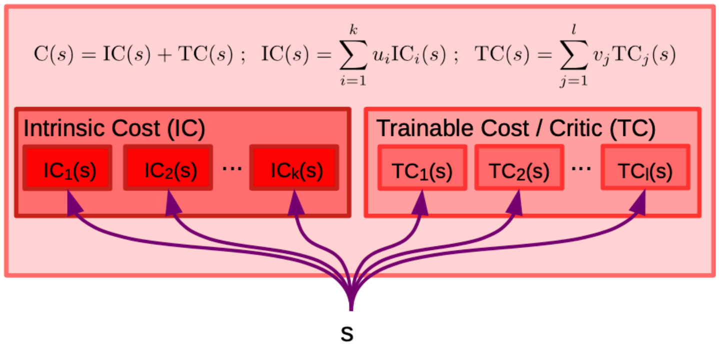
- The perception module estimates the state of the world s[0].
- The actor proposes a sequence of actions a[0], a[1], . . . , a[t], a[t + 1], . . . , a[T].
- The world model recursively predicts an estimate of the world state sequence using s[t + 1] = Pred(s[t], a[t]).
- The cost C(s[t]) computes an energy for each predicted state in the sequence, the total energy being the sum of them.

Mode-2 Perception-Action Episode



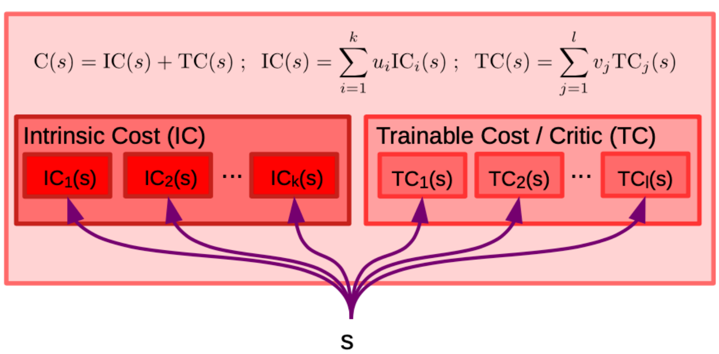
- Through an optimization or search procedure, the actor infers a sequence of actions that minimizes the total energy.
- It then sends the first action in the sequence (or the first few actions) to the effectors.
- Since the cost and the model are differentiable, gradient-based methods can be used to search for optimal action sequences.
- Pairs of states (computed by the encoder or predicted by the predictor) and corresponding energies from the intrinsic cost and the trainable critic are stored in the short-term memory for subsequent training of the critic.

Architecture of Cost Module



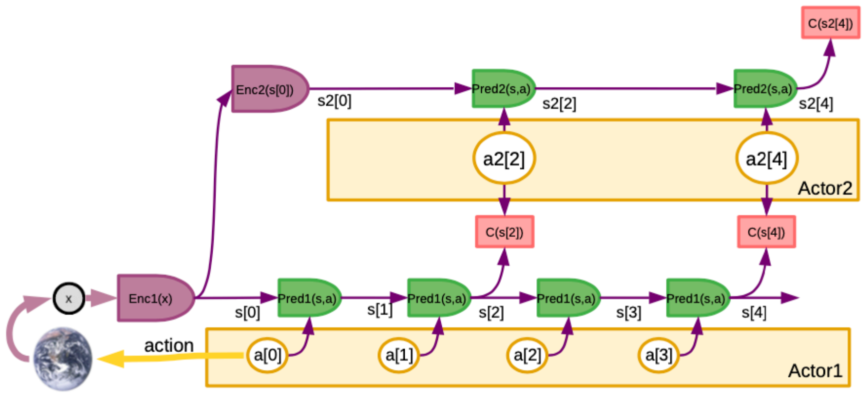
- The cost module comprises the:
  - intrinsic cost module which is immutable $\mathrm{IC}_i(s)$ (left) and
  - the critic or Trainable Cost $\mathrm{TC}_j(s)$ (right), which is trainable.
- Both IC and TC are composed of multiple submodules whose output energies are linearly combined.

Architecture of Cost Module



- Each submodule imparts a particular behavioral drive in the agent.
- The weights in the linear combination, u_i and v_j, are determined by the configurator module and allow the agent to focus on different subgoals at different times.

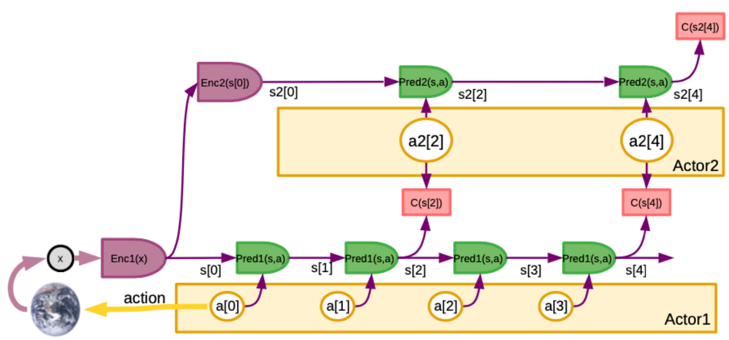
Hierarchical JEPA for Mode-2 hierarchical planning



- A complex task is defined by a high-level cost computed from a high-level world-state representation C(s2[4]).
- A sequence of high-level abstract actions (a2[2],a2[4]) is inferred that minimizes C(s2[4]).

Hierarchical JEPA for Mode-2 hierarchical planning

- The inferred abstract actions are fed to lower-level cost modules C(s[2]), C(s[4]) which define subgoals for the lower layer.



- The lower layer then infers an action sequence that minimizes the subgoal costs.
- Although only a 2-layer hierarchy is shown here, it is straightforward to extend the concept to multiple levels.
- The process described here is sequential top-down, but a better approach would be to perform a joint optimization of the actions in all the layers.