Chapter 9

Open Problems and Directions about Intelligence

> "*The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problem now reserved for humans, and improve themselves.*"
>
> – Proposal for the Dartmouth AI program, 1956

This manuscript systematically introduces mathematical principles and computational mechanisms for how memory or empirical knowledge can be developed from observed high-dimensional data. The ability to seek parsimony in a seemingly random world is a fundamental characteristic of any intelligence, natural or artificial. We believe the principles and mechanisms presented in this book are unifying and universal, applicable to both animals and machines.

We hope this book helps readers fully clarify the mystery surrounding modern practices of artificial deep neural networks by developing a rigorous understanding of their functions and roles in learning (representations of) low-dimensional distributions from high-dimensional data. With this understanding, the capabilities and limitations of existing AI models and systems become clear:

1. Existing open-loop models and systems fall short of being complete memory systems capable of self-learning and self-improving.
2. Existing realizations of representation learning remain primitive and brute-force, far from optimal in terms of network architectures or optimization

strategies.

3. Existing AI models only learn data distributions and conduct inductive Bayesian inference, which differs from high-level human intelligence.

One goal of this book is to help readers establish an objective and systematic understanding of current machine intelligence technologies and to recognize what open problems and challenges remain for further advancement of machine intelligence. In the last chapter, we provide some of our views and projections for the future.

9.1 Towards Autonomous Intelligence: Close the Loop?

From the practice of machine intelligence in the past decade, it has become clear that, given sufficient data and computational resources, one can build a large enough model and pre-train it to learn the *prior* distribution of all the data, say $p(\boldsymbol{x})$. Theoretically, such a large model can memorize almost all existing knowledge about the world that is encoded in data such as 2D images, 3D shapes, dynamical motions, and natural languages. As we discussed at the beginning of the book, such a large model plays a role similar to DNA, which life uses to record and pass on knowledge about the world.

The model and distribution learned in this way can then be used to create new data samples drawn from the same distribution. One can also use the model to conduct inference (e.g., completion, estimation, and prediction) with the memorized knowledge under various conditions, say by sampling the *posterior* distribution $p(\boldsymbol{x} \mid \boldsymbol{y})$ under a new observation $\boldsymbol{y}$. Strictly speaking, such inference is inductive or statistical.

Any pre-trained model, however large, cannot guarantee that the distribution it has learned so far is entirely correct or complete. If our samples $\hat{\boldsymbol{x}}_t$ from the current *prior* $p_t(\boldsymbol{x})$ or estimates $\hat{\boldsymbol{x}}_t(\boldsymbol{y})$ based on the *posterior* $p_t(\boldsymbol{x} \mid \boldsymbol{y})$ are inconsistent with the truth $\boldsymbol{x}$, we would like to correct the learned distributions:

$$p_t(\boldsymbol{x}) \to p_{t+1}(\boldsymbol{x}) \quad \text{or} \quad p_t(\boldsymbol{x} \mid \boldsymbol{y}) \to p_{t+1}(\boldsymbol{x} \mid \boldsymbol{y}), \tag{9.1.1}$$

based on the error $\boldsymbol{e}_t = \boldsymbol{x}_t - \hat{\boldsymbol{x}}_t$. This is known as error correction based on error feedback, a ubiquitous mechanism in nature for continuous learning. However, any open-ended model itself lacks the mechanism to revise or improve the learned distribution when it is incorrect or incomplete. Improving current AI models still depends largely on human involvement: supervision or reinforcement through experimentation, evaluation, and selection. We may call this process "artificial selection" of large models, as opposed to the natural selection for the evolution of life.

As we studied earlier in this book (Chapter 6 in particular), closed-loop systems align an internal representation with sensed observations of the external world. They can continuously improve the internally learned distribution

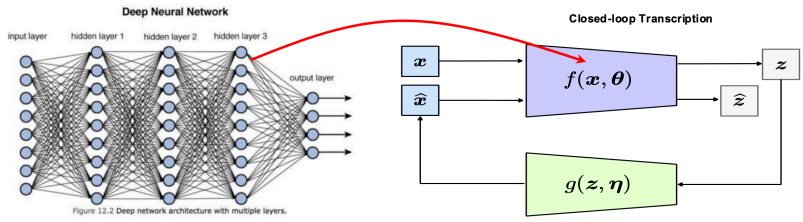

Figure 9.1: From an open-ended deep network to a closed-loop system.

and its representation to achieve consistency or self-consistency. An immediate next step is to develop truly closed-loop memory systems, illustrated in Figure 9.1, that autonomously and continuously learn general data distributions and improve based on error feedback.

Therefore, the transition from the currently popular end-to-end trained open-loop models to continuously learning closed-loop systems

$$\textbf{open-ended} \text{ models} \implies \textbf{closed-loop} \text{ systems} \tag{9.1.2}$$

is the key for machines to truly emulate how the animal brain learns and applies knowledge in an open world. We believe that

open-ended models are for a closed world, however large;
closed-loop systems are for an open world, however small.

Hence, the so-called "artificial general intelligence" (AGI) can never be achieved by simply having a system to memorize all existing data and knowledge of the world. No knowledge, however much, is truly generalizable.[1] Nevertheless, it is the ability to improve one's existing memory or knowledge and adapt to any new environments that is truly generalizable. Hence, truly genuine intelligence is itself generalizable if realized completely and correctly.[2] This is precisely the essence of the Cybernetics program laid out by Norbert Wiener in the 1940s that we discussed at the very beginning of this book.

9.2 Towards Natural Intelligence: Beyond Back Propagation?

The practice of machine intelligence in recent years has led many to believe that one must build a single large model to learn the distribution of all data and memorize all knowledge. Even though this is technologically possible, such a solution is likely far from necessary or efficient. As we know from training deep networks, the only known scalable method to train such networks at scale

[1] According to Sir Karl Popper, an influential Philosopher of Science, all scientific theory and knowledge are falsifiable!

[2] In our opinion, AGI should represent "artificial genuine intelligence."

is through back propagation (BP) [RHW86b]. Although BP offers a way to correct errors via gradient signals propagated back through the whole model, it is nevertheless rather brute-force and differs significantly from how nature learns: BP is an option that nature cannot afford due to its high cost and simply cannot implement due to physical limitations.

More generally, we cannot truly understand intelligence unless we also understand how it can be efficiently implemented. That is, one needs to address the computational complexity of realizing mechanisms associated with achieving the objectives of intelligence. Historically, our understanding of (machine) intelligence has evolved through several phases, from the incomputable Kolmogorov complexity to Shannon's entropy, from Turing's computability to later understanding of tractability,[3] and to the strong emphasis on algorithm scalability in modern practice of high-dimensional data analysis [WM22] and artificial intelligence. This evolution can be summarized as follows:

$$\textbf{incomputable} \implies \textbf{computable} \implies \textbf{tractable} \implies \textbf{scalable}. \tag{9.2.1}$$

To a large extent, the success and popularity of deep learning and back propagation is precisely because they have offered a scalable implementation with modern computing platforms (such as GPUs) for processing and compressing massive data. Nevertheless, such an implementation is still far more expensive than how nature realizes intelligence.

There remains significant room to improve the efficiency of machine intelligence so that it can emulate the efficiency of natural intelligence, which should be orders of magnitude greater than current brute-force implementations. To this end, we need to discover new learning architectures and optimization mechanisms that enable learning data distributions under natural physical conditions and resource constraints, similar to those faced by intelligent beings in nature—for example, without accessing all data at once or updating all model parameters simultaneously (via BP).

The principled framework and approach laid out in this book can guide us to discover such new architectures and mechanisms. These new architectures and mechanisms should enable online continuous learning and should be updatable through highly localized and sparse forward or backward optimization.[4]

As we have learned from neuroscience, the cortex of our brain consists of tens of thousands of cortical columns [Haw21]. All cortical columns have similar physical structures and functions. They are highly parallel and distributed, though sparsely interconnected. Hence, we believe that to develop a more scalable and structured memory system, we need to consider architectures that

[3] We say a problem is tractable if it allows an algorithm whose complexity is polynomial in the size of the problem.

[4] For learning a distribution, a simple instantiation of these desiderata is in the simplest case of PCA, with the online PCA method introduced in Chapter 6. Once the linear model for the distribution is learned, the easier task of conducting online prediction with noisy observations can then be easily accomplished by least-square type methods, such as the Wiener filter [Wie42; Wie49]. In the case of linear dynamical systems, that becomes the famous the Kalman filter, see [MKS+04, Appendix B] or [PV25].

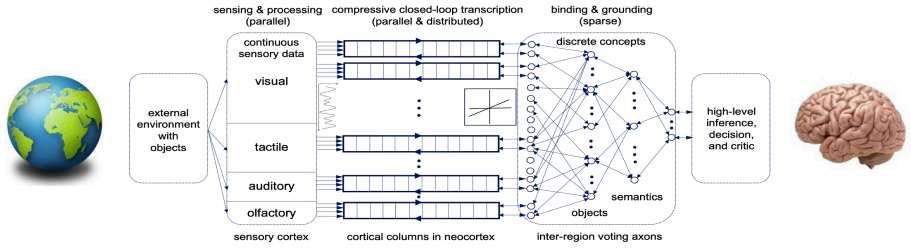

Figure 9.2: Conjectured architecture of the brain cortex. The cortex is a massively parallel and distributed auto-encoding system consisting of a hierarchy of closed-loop auto-encoders. They extract information from their inputs and maximize the information gain of the representations they output. Such processes are done at multiple levels of the hierarchy with different levels of abstraction of the information.

emulate those of the cortex. Figure 9.2 shows such a hypothesized architecture: a massively distributed and hierarchical system consisting of many largely parallel closed-loop auto-encoding modules.[5] These modules learn to encode different sensory modalities or many projections of data from each sensory modality. Our discussion in Section 7.5 of Chapter 7 suggests that such parallel sensing and learning of a low-dimensional distribution is theoretically possible. Higher-level (lossy) autoencoders can then be learned based on outputs of lower-level ones to develop more sparse and higher-level "abstractions" of the representations learned by the lower levels.

The distributed, hierarchical, and closed-loop system architecture illustrated in Figure 9.2 shares many characteristics of the brain's cortex. Such a system architecture may open up many more possibilities than the current single large-model architecture. It enables exploration of much more efficient learning and optimization mechanisms and results in a more structured modular organization of the learned data distribution and knowledge, more closely emulating the memory in our brain. This would allow us to bring the implementation of machine intelligence to the next level of evolution:

$$\textbf{incomputable} \Longrightarrow \textbf{computable} \Longrightarrow \textbf{tractable} \Longrightarrow \textbf{scalable} \Longrightarrow \textbf{natural}. \tag{9.2.2}$$

[5] Elements of such hypothetical architectures exist in the literature in various forms, such as predictive coding networks [MSS+22], which have an optimization strategy that utilizes efficient local updates.

9.3 Towards Scientific Intelligence: Beyond the Turing Test?

9.3.1 The Evolution of Intelligence in Nature

As we have discussed at the beginning of this book, Chapter 1, intelligence in nature has evolved through multiple phases and manifested in four distinct forms:

$$\textbf{phylogenetic} \implies \textbf{ontogenetic} \implies \textbf{societal} \implies \textbf{scientific intelligence}. \tag{9.3.1}$$

All forms of intelligence share the common objective (and characteristic) of learning useful knowledge as low-dimensional distributions of sensed high-dimensional data about the world. Nevertheless, they differ in the specific coding schemes adopted, the information encoded, the computational mechanisms for learning and improving, and the physical implementations of such mechanisms. Using the concepts and terminology developed in this book, the four stages of intelligence differ in the following three aspects:

1. The *code book* or scheme used to encode the intended information or knowledge.
2. The *information* or knowledge learned and encoded using the code book.
3. The *optimization mechanisms* used to create and correct the encoded information or knowledge.

Table 9.1 summarizes their main characteristics.

	Phylogenetic	**Ontogenetic**	**Societal**	**Scientific**
Codebook	DNAs	Neurons/Brain	Natural Languages	Math Abstractions
Information	Genes	Memory	Empirical Knowledge	Scientific Facts
Optimization	Reinforcement	Error Feedback	Trial & Error	Theorize & Falsify

Table 9.1: Main characteristics of the four stages of intelligence in nature.

As we now know, humans have achieved two quantum leaps in intelligence.

1. The first was the development of spoken and written language, which enabled humans to share and transmit learned knowledge across generations, much as DNA does in nature.
2. The second was the development of mathematics and formal logic roughly three thousand years ago, which became the precise language of modern science.

The language of mathematics freed us from summarizing knowledge from observations only in empirical form and allowed us to formalize knowledge as theories verifiable or falsifiable through mathematical deduction or experimentation.

Through hypothesis formulation, logical deduction, and experimental verification or falsification, we can now proactively discover and develop new knowledge that is far beyond what can be learned from the distributions of observed data. For example, causal relationships cannot be learned from distributions alone [Pea09].

Despite the seeming differences among these different stages of intelligence, the evolution of intelligence shares a common characteristic: *they all use certain forms of feedback*[6] *to create and correct information learned, and the feedback and correction are becoming increasingly frequent and efficient through evolution—from open-loop reinforcement to closed-loop self-supervision, from collective trial and error to proactive hypothesizing and falsification.*

9.3.2 From Inductive to Deductive Intelligence

As discussed in the introduction (Chapter 1), the 1956 "artificial intelligence" (AI) program aimed precisely to study how to endow machines with scientific intelligence, i.e., high-level functions such as mathematical abstraction, causal inference, logical deduction, and problem solving that are believed to differentiate humans from animals:

$$\textbf{low-level}\ \text{(animal) intelligence} \implies \textbf{high-level}\ \text{(human) intelligence}. \tag{9.3.2}$$

As we have clarified repeatedly in this book, most technological advances in machine intelligence over the past decade or so, although carried out under the name "AI", are actually more closely related to having machines imitate low-level forms of intelligence largely shared by both animals and humans, including phylogenetic, ontogenetic, and societal intelligence. At these levels, intelligence creates memory or *empirical* knowledge (from empirical data distributions) and conducts inference that is mostly *inductive* in nature. Principles and methods introduced in this book aim to reveal scientific objectives and computational mechanisms behind intelligence at these levels. They provide strong theoretical justification and computational evidence for the claim:

Machines can learn empirical knowledge and conduct inductive inference.

So far, however, no rigorous theoretical or scientific evidence suggests that these mechanisms alone would suffice to achieve the high-level human intelligence that the original AI program truly aimed to understand and wanted for machines to imitate or even surpass.

In fact, as of today, we know little about how to rigorously verify or certify whether a system is truly capable of such high-level intelligence, even though the Turing Test, also known as the Imitation Game,[7] was proposed in 1950 [Tur50], as an initial attempt to address the fundamental question:

[6] Loss for error and reward for success.

[7] In Turing's proposal, the evaluator, or interrogator, is a human. However, most human evaluators have limited scientific training and knowledge, and their conclusions can be subjective.

Can machines think?

For a long time, a precise definition of such a test was not deemed necessary since machine capabilities were far below those of humans (or even animals). However, given recent technological advances, many models and systems now claim to reach or even surpass human intelligence. Therefore, it is high time to develop a scientific and executable definition of the Turing test, i.e., a systematic and rigorous protocol to evaluate and certify the intelligence level of a model or a system that claims to be "intelligent."

As Turing argued in his original paper, it is rather difficult to define what "thinking" is. We may instead start with something more tangible and try to determine whether a system can truly "understand" certain concepts or knowledge. For example, how can we rigorously verify whether an intelligent model or system has truly understood an abstract concept such as:

1. the notion of equality between two quantities,

2. the notion of numbers (natural, rational, real, or imaginary),

3. the notion of infinity and mathematical induction,

4. the notion of logic consistency[8] and proof by contradition,

just to name a few? Or has it simply memorized a massive number of examples of such notions? Note that state-of-the-art large language models still struggle with simple mathematical questions like: "Is 3.11 larger or smaller than 3.9?"[9]

From this book, we know modern generative AI systems primarily rely on Bayesian (hence inductive) inference to produce their answers. Can machines also truly grasp mathematical induction or general logical deduction and understand its difference from Bayesian induction? How do we verify whether a system truly understands the rules of logical and mathematical deduction and can apply them rigorously? Or, again, has it merely memorized a large number of instances of such deductions, such as chain-of-thought data, and learned and exploited their distributional patterns? Hence we have an open question:

Is there any difference between ***memorizing*** *and* ***understanding****?*

Only by knowing the difference can we possibly attempt to provide a meaningful answer to the question: "*Can machines understand?*"

[8] In mathematics, if a set of axioms and their deduced results are consistent and do not contradict one another, then they exist. This is known as Hilber's Principle. This is different from the notion of "consistency" in Chapter 6, which requires the learned representations to be consistent with the empirical data distributions. One notion of consistency is inductive in nature and the other is deductive.

[9] Some models have corrected their answers to such questions through targeted engineering, or have incorporated additional reasoning mechanisms that verify immediate answers and correct them during inference. However, we leave it to the reader as an exercise to rigorously test whether any state-of-the-art language model truly understands the notion of numbers (natural, rational, real, and complex) and their associated arithmetic.

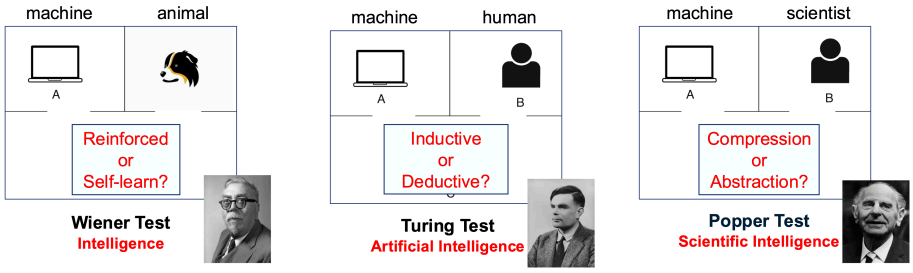

Figure 9.3: Three tests for different levels or types of intelligence capabilities: the Wiener test for basic intelligence, the Turing test for human-level intelligence, and the Popper test for scientist-level intelligence.

Probably much more importantly, how do we verify whether a self-claimed intelligent system is capable of *creating abstract concepts or improving its own knowledge* proactively via deductive means? For example, can it create new mathematical concepts and discover new physical laws or causal relationships? The connection between understanding and creating can be rather fundamental, as the physicist Richard Feynman famously put it:

"What I cannot create, I do not understand."

9.3.3 Scientific Tests of Intelligence

Hence, it is high time we develop scientifically sound evaluation methods to determine which of the following categories a system's intelligent capability belongs to:

Level-1. having merely learned the distribution of some knowledge-carrying data and being able to regenerate them;

Level-2. being capable of autonomously and continuously learning new knowledge or correcting existing knowledge from new observations and experiences;

Level-3. being able to truly understand and create abstract concepts, rules and laws and knowing how to apply them correctly in a deductive manner;

Level-4. being able to generate new scientific hypotheses or math conjectures and verify or falsify them based on experimentation or logical deduction.

To evaluate and distinguish these different levels of intelligence, we suggest that there should probably be at least three different tests, as illustrated in Figure 9.3:

1. *The Norbert Wiener Test:* to determine whether a system can self-correct and create new (empirical) knowledge on its own or merely receives and updates information passively through external reinforcement or supervision;

2. *The Alan Turing Test:* to determine whether a system can create and understand abstract concepts and knowledge or merely learns and memorizes statistics of the samples and uses them for Bayesian inference;

3. *The Karl Popper Test:* to determine whether a system can proactively explore and create new abstract knowledge by forming and verifying hypotheses or theories based on experimental consistency or logical self-consistency.

We believe that, for such tests, the evaluator, or interrogator, should not be a single human or an arbitrary group of humans but rather a scientifically certified entity following a scientifically sound protocol and process so that the conclusions would be rigorous and trustworthy.

As we have seen throughout this book, *compression* has played a fundamental role in developing a memory or creating empirical knowledge. It is the governing principle and a universal mechanism for identifying an empirical data distribution and organizing the information encoded therein with a structured representation. To a large extent, it explains the practice of "artificial intelligence" with deep networks, which is largely inductive in nature.[10] An outstanding question for future study is whether *compression alone* is sufficient to achieve all the higher-level deductive intelligence mentioned above. In particular, are mathematical abstraction, causal inference, hypothesis generation, and logical reasoning and deduction some kind of extended or transcended forms of compression? Is there some fundamental difference between learning data distributions through compression and forming high-level concepts and theories through abstraction? Hence we have another open question:

Is there any difference between ***compression*** *and* ***abstraction****?*

Only by knowing the difference can we possibly attempt to provide a meaningful answer to the question: "*Can machines create knowledge proactively via deduction?*"

To a large extent, Science—and its associated code book, Mathematics—can be viewed as the most advanced form of intelligence, unique to educated and enlightened humans. Philosopher Sir Karl Popper once suggested:

"Science may be described as the art of systematic oversimplification."

We believe that uncovering and understanding the underlying mathematical principles and computational mechanisms of such higher-level intelligence, and successfully reproduce them through machines, will be the final frontier for Science, Mathematics, and Computation altogether!

[10] Systems that have built-in deductive mechanisms for conducting formal verification are not in this category.