Centre for International Governance Innovation

Digital Policy Hub

Digital Policy Hub – Working Paper

Machine-Learning Theory and Its Policy Implications

Naod Abraham

Fall 2023 cohort

About the Hub

The Digital Policy Hub at CIGI is a collaborative space for emerging scholars and innovative thinkers from the social, natural and applied sciences. It provides opportunities for undergraduate and graduate students and post-doctoral and visiting fellows to share and develop research on the rapid evolution and governance of transformative technologies. The Hub is founded on transdisciplinary approaches that seek to increase understanding of the socio-economic and technological impacts of digitalization and improve the quality and relevance of related research. Core research areas include data, economy and society; artificial intelligence; outer space; digitalization, security and democracy; and the environment and natural resources.

The Digital Policy Hub working papers are the product of research related to the Hub's identified themes prepared by participants during their fellowship.

About CIGI

The Centre for International Governance Innovation (CIGI) is an independent, non-partisan think tank whose peer-reviewed research and trusted analysis influence policy makers to innovate. Our global network of multidisciplinary researchers and strategic partnerships provide policy solutions for the digital era with one goal: to improve people's lives everywhere. Headquartered in Waterloo, Canada, CIGI has received support from the Government of Canada, the Government of Ontario and founder Jim Balsillie.

Partners

Thank you to Mitacs for its partnership and support of Digital Policy Hub fellows through the Accelerate program. We would also like to acknowledge the many universities, governments and private sector partners for their involvement allowing CIGI to offer this holistic research environment.

67 Erb Street West
Waterloo, ON, Canada N2L 6C2
www.cigionline.org

Key Points

- Learning is the process of converting experience to expertise. This is a broad definition of learning that also adheres to our intuitive understanding of the concept. Learning, whether that is done by a computer or a living organism, has two parts: an input, which is experience or training data in the case of a computer; and an output, which is the expertise.
- The goal of computational learning theory (a subfield of machine learning) is thus to rigorously analyze this process of learning using mathematics. Some of the mathematical results of this vast area of study that are discussed in this paper do not necessarily apply to only computers, but may also apply to anything that can learn, including living organisms.
- The goal of this paper is to present an intuitive summary of computational learning theory, and its application for analyzing the most popular learning algorithms in machine learning, neural networks.
- This paper assumes no mathematical background or knowledge of machine learning from the reader. Throughout the paper, the important sections are presented using a pyramid approach (that is, in three levels). Those levels are: first, assuming no mathematical background or knowledge of machine learning; second, assuming knowledge of calculus, statistics and linear algebra; and third, assuming knowledge of machine learning and all its prerequisites. The latter levels are directed to the specialized reader, whereby the discussion in the preceding levels is for all readers.
- Throughout the paper, after introducing each major idea/topic, its policy implications and connection to Canada's law are discussed. Finally, a summary of Canada's relevant artificial intelligence (AI) law, with a case example, is given.

A Gentle Introduction to Machine-Learning Theory through Psychology

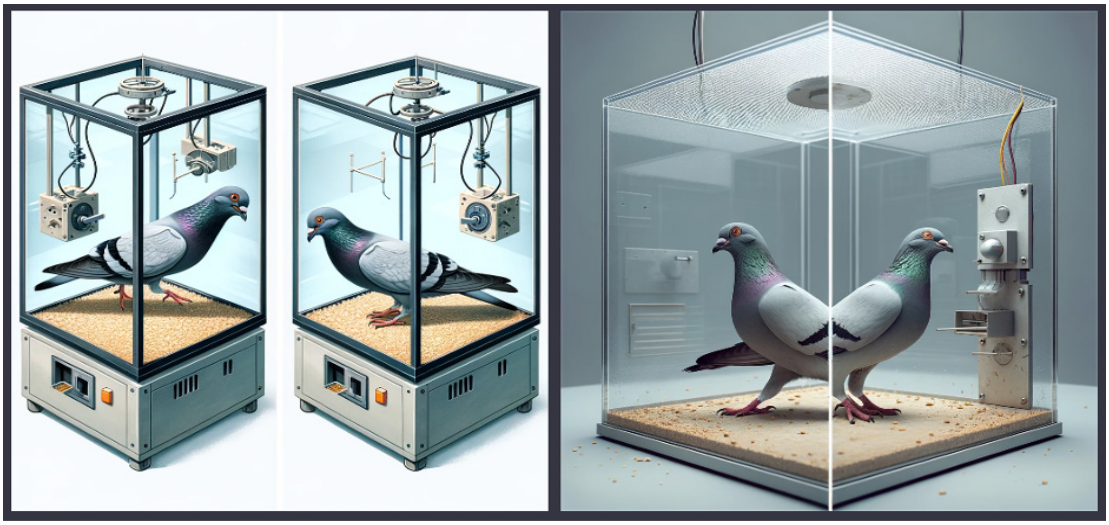
There are various experiments performed in the field of psychology that help in understanding the fundamentals of machine learning. This paper will begin with a walk-through of two such experiments, to illustrate some key results in machine learning that will inform the later sections of the paper.

B. F. Skinner's Pigeon Experiment

First is an experiment performed by psychologist B. F. Skinner (1948) on pigeons. In the experiment, each pigeon was brought to a state of hunger and placed in a box (see Figure 1). The box was programmed to deliver food to the pigeon at regular intervals. After a few intervals, Skinner found that during the arrival of the food, each pigeon was repeating the same behaviour (turning counter-clockwise, pecking and so on).

The cause of the behaviour was that each pigeon in the experiment began to associate the delivery of food with whatever particular action it had been performing at the time the food was delivered. Thus, the pigeons began to repeat that action in hopes of receiving more food, even though the food was delivered at regular intervals.

Figure 1: Pigeon Experiment

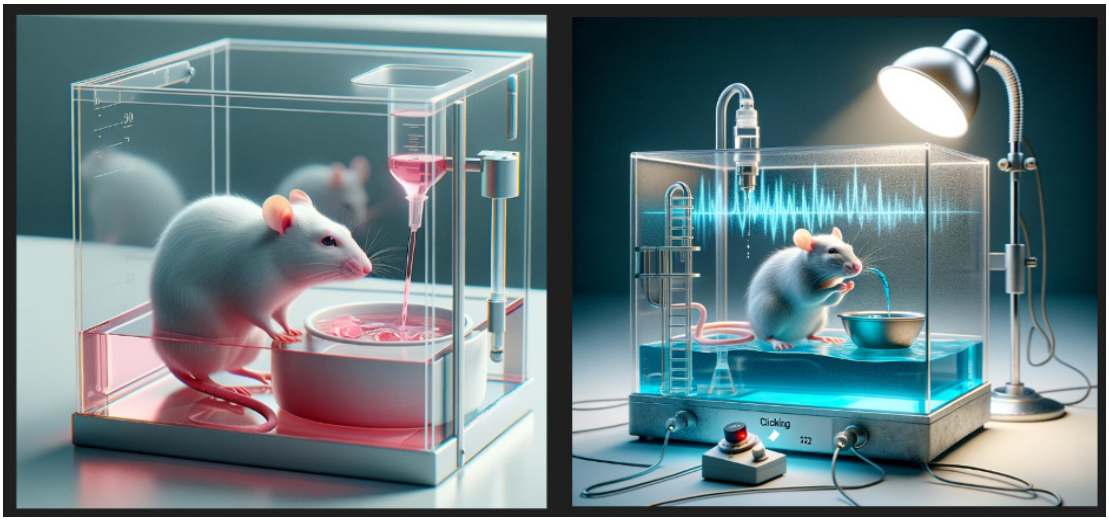


Source: Generated with OpenAI.

Garcia and Koelling's Rat Avoidance Learning Experiment

This experiment was performed by the psychologists John Garcia and Robert A. Koelling (1966) on rats. It focused on wild rats' ability to use their olfactory (smell) and gustatory (taste) cues to learn to avoid poison bait. This ability of rats is called "bait shyness" and develops very quickly after only a few exposures (Barnett 1963). Each rat was brought to a state of thirst and placed in a box (see Figure 2). Each box was programmed to deliver two types of water. The first type was "tasty" water, which had flavours in it. The second type was "bright-noisy" water, which was connected to a bright incandescent lamp and a clicking relay. After drinking, the rats were exposed to four different stimuli (see Table 1).

Figure 2: Rat Experiment



Source: Generated with OpenAI.

Table 1: Rat Stimuli

Stimulus Type	Stimulus Cause
Nausea and gastric upset	Radiation (X-ray)
Nausea and gastric upset	Chemical (lithium chloride) in the water
Shock sensation	Electric current
Delayed shock sensation	Electric current

Source: Author.

Comparing this experiment with Skinner's pigeon experiment, it would make sense to assume that if the pigeons were so quick to associate a random behaviour with the arrival of food, then surely the rats must have associated the different stimuli with the water type. Surprisingly, the rats failed to sometimes associate the two (see Table 2). For example, they seemed to have a "built-in" knowledge (perhaps due to their genetics) telling them nausea/ill effect can be associated with the taste of the water, but not with the brightness or noise of the water.

Table 2: Rat Experiment Results Summary

Action	Association Made?
Gastric upset (X-ray) and "tasty" water	Yes
Gastric upset (X-ray) and "bright-noisy" water	No
Gastric upset (lithium chloride) and "tasty" water	Yes
Gastric upset (lithium chloride) and "bright-noisy" water	No
Shock sensation and "tasty" water	No
Shock sensation and "bright-noisy" water	Yes
Delayed shock sensation and "tasty" water	No
Delayed shock sensation and "bright-noisy" water	Yes

Source: Author.

The takeaway of comparing these two experiments is that the rats were the more successful learners. The pigeons were willing to accept any explanation for the arrival of the food, while the rats were picky about what should be naturally associated with each other. Part of the rats' success can be explained by their natural "bait-shyness" ability mentioned above. This need to have some prior knowledge to bias our learning is necessary and unavoidable for learning successfully. This important mathematical result is fittingly called the "no-free-lunch theorem," which is discussed in this paper. Moreover, the no-free-lunch theorem is an important result because it tells us that there can never be an AI system that is ready to tackle any task. Although the details of the no-free-lunch theorem will be discussed later, it is stated formally below.

No-Free-Lunch Theorem

Theorem (no free lunch): Let A be any learning algorithm for the task of binary classification with respect to the 0–1 loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that:

1. There exists a function $f: \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$.
2. With probability of at least 1/7 over the choice of $S \sim \mathcal{D}^m$ we have that $L_{\mathcal{D}}(A(S)) \geq 1/8$

Source: Shalev-Shwartz and Ben-David (2014).

A Hypothetical Scenario to Illustrate Machine-Learning Mathematical Results

A hypothetical real-life learning scenario is presented in this section to demonstrate some of the vast number of mathematical results that come from machine learning. This scenario is simple enough to intuitively grasp, yet powerful and rich enough to demonstrate many mathematical results in this subject area. The following outlines the scenario.

Bob has never eaten mangoes before; however, he has eaten many other fruits and vegetables. To learn more about mangoes, his job now is to go to the same grocery store every day and buy 10 mangoes at random (see Figure 3). Afterward, he will go home and try them all. He will do this for 30 days. Finally, at the end of the 30 days, he must come up with an algorithm for deciding if a mango is tasty or not, without eating it. He must decide based on two parameters: the softness of the mango and the colour. Moreover, he must record his results in a list for the 300 mangoes he tried, outlining the softness (measured from 0 to 1), colour and his conclusions (tasty or not tasty) (see Table 3).

Figure 3: Bob Buying Mangoes

Source: Generated with OpenAI.

Table 3: Bob's List

Mango ID	Softness (0–1)	Colour	Tasty/Not Tasty
Mango 1	0.6	Yellow-orange	Tasty
Mango 2	0.2	Green-yellow	Not tasty
...	...	...	...
Mango 300	0.7	Deep orange	Tasty

Source: Author.

In our hypothetical universe, the mangoes are "tasty" when their softness and colour are 0.3–0.7 and yellow to deep orange, respectively. However, this knowledge is not known to Bob initially.

Linking Bob's Scenario to the Use of Machine Learning

To begin, the statistical learning setting is introduced. In a statistical learning setting, a learner has access to the following six elements (Shalev-Shwartz and Ben-David 2014):

- **Domain set:** The set where all the objects of interest in the learning task come from. For Bob, it is all the mangoes in the universe.
- **Label set:** The labels for the objects in the domain set. For Bob, if a mango tastes sweet, then it is "tasty," but "not tasty" otherwise.
- **Training data:** The set of objects from the domain set with their labels to use for learning. By the end of the 30 days, Bob will have tried 300 mangoes. This is his training data.

- **The learner's output:** The expertise gained by the learner after learning. For Bob, it is the algorithm he will provide us at the end of the month. This is often called a "hypothesis." It is usually denoted by "*h*."
- **A data generation model:** Some system that chooses objects from the domain set. For Bob, the store owner only buys mangoes produced in Mexico, and the mangoes' weight should be between 300 g and 700 g.
- **Measure of success:** A way to measure a learner's success at learning. At the end of 30 days, Bob will be given a fresh set of mangoes, and he will label them using his algorithm. It is possible to calculate Bob's "error" (sometimes called "loss") by counting how many he got wrong and dividing that number by the total of mangoes he was given.

Overfitting

The next concept/problem in machine learning to be introduced is "overfitting" (ibid.). Overfitting is a very common problem in machine learning; it is analogous to memorization and is not proper learning. Tying this concept back to the hypothetical scenario with Bob, overfitting can be demonstrated as follows.

A naive algorithm Bob can choose is as follows:

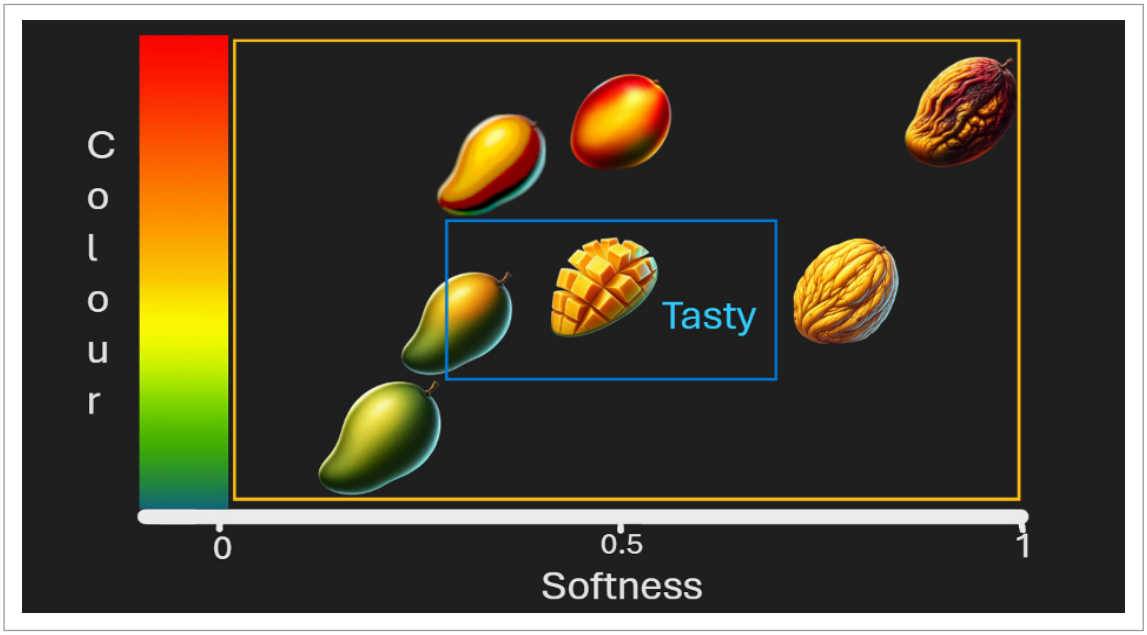
Input: A new random mango.
Output: If this new mango has exactly the same softness and colour as one of the 300 mangoes he has tried, he will look at his list and tell us if his list says tasty or not. Otherwise, if a mango is not one of the 300 mangoes he has tried, he will immediately say "not tasty."

Source: Shalev-Shwartz and Ben-David (2014).

What is wrong with this hypothesis/algorithm?

- **Level one:** This approach is no different than simple memorization. If Bob is given a mango exactly the same as one he has seen before, then his algorithm will give the correct output. However, since the odds that a new mango given to him is exactly the same as one he has seen before is very low, simply saying it is "not tasty" is not much different from guessing.
- **Level two:** Let Bob's hypothesis, loss/error on the 300 mangoes, and loss/error on all the mangoes in the universe be h, $L_s(h)$, and $L_u(h)$, respectively. Refer to the "measure of success" section above on how to calculate loss/error for Bob. Clearly, $L_s(h) = 0$. However, $L_u(h) = \frac{tasty-area}{total-area} = \frac{1}{2}$ (see Figure 4). Hence, Bob produced a hypothesis that looks perfect (0 error) on his training sample but is no better than luck/guessing on all other mangoes.

Figure 4: Tasty Mangoes



Source: Generated with OpenAI.

Note: All the tasty mangoes lie within the small rectangle (which has an area equal to one). Suppose the area of the big rectangle is two.

- How is overfitting avoided in practice? This is where the no-free-lunch theorem starts to come into play. One cannot simply jump into a learning task without some form of prior knowledge about the problem. Otherwise, they will be vulnerable to problems such as overfitting. Here, the concept of a "hypothesis class" is introduced. It is usually denoted by "*H*." Before, Bob was free to think up any algorithm/hypothesis he would like. However, now he must limit himself to thinking about only a certain number of hypotheses/algorithms. The metaphorical bag of all hypotheses/algorithms that he is now limited to picking from is called a "hypothesis class" (see Figure 5). From his experience with other fruits, Bob knows that other fruits lie within an interval sweet spot. For example, for bananas, a "tasty" sweet spot might roughly look like this: softness is 0.4 to 0.6; and colour is very light green to bright yellow. Hence, he chooses his hypothesis class to be the set of such a pair of softness and colour intervals (see Figure 5).

Figure 5: Bob Thinking about a Hypothesis Class for Mangoes

Hypothesis–ID	Hypothesis description
Hypothesis–1	Softness between 0.1–0.2 and colour between light green to dark green is "tasty." Otherwise, "not tasty."
Hypothesis–2	Softness between 0.8–0.9 and colour between light green to dark green is "tasty." Otherwise, "not tasty."
Hypothesis–3	Softness between 0.6–0.8 and colour between deep orange to bright red is "tasty." Otherwise, "not tasty."
Hypothesis–4	Softness between 0.4–0.7 and colour between yellow to deep orange is "tasty." Otherwise, "not tasty."
...	.....

Source: Generated with OpenAI.

The No-Free-Lunch Theorem

For Bob, does limiting himself to this hypothesis class offer him any advantage? To help answer this, the no-free-lunch theorem will be formally introduced (ibid.).

- **Level one:** For Bob, the no-free-lunch theorem would say that if he does not limit himself to an appropriate hypothesis class or if he chooses any hypothesis class at random, there will be consequences. The consequence is that there will be a data generation model (a grocery store in this case) where he will fail to learn properly, even after trying so many mangoes by the end of the 30 days. Hence, just like the rats were able to bias their learning based on their natural bait-shyness ability, Bob similarly biases his learning based on his prior knowledge about other fruits and vegetables to help him learn better.

- **Level two:** The no-free-lunch theorem states that: if for *any* hypothesis class and *any* data generation model chosen where we want to be guaranteed that the hypothesis the learner will pick will have a loss/error lower than $1/8 = 0.125$, at least $(100 * \frac{1}{7})\%$ of the time, there is no choice but to consider at least half of the points in the domain set. Realistically, during learning, we only have access to a training set that is a tiny portion of the points in the domain set. In Bob's case, the 300 mangoes he has access to is a small portion of all the mangoes in the universe. Tasting half of the mangoes in the universe just for the sake of learning to recognize when a mango is tasty is not realistic. Therefore, the immediate implication is that no learner who picks their hypothesis from some bag of hypothesis class is guaranteed to succeed in every task. Hence, it is important to be careful when choosing a hypothesis class in order to avoid hypothesis classes that might lead you astray. A bad hypothesis class can be avoided by choosing one based on some prior knowledge about the problem.

■ **Level three:** An immediate corollary of the no-free-lunch theorem is:

Definition (probably approximately correct [PAC] learnability): A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function $m_{\mathcal{H}}: (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over $\mathcal{X}$, and for every labelling function $f: \mathcal{X} \to \{0,1\}$, if the realizable assumption holds with respect to $\mathcal{H}, \mathcal{D}, f$, then when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ independent and identically distributed examples generated by $\mathcal{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $1 - \delta$ (over the choice of the examples), $L_{(\mathcal{D},f)}(h) \leq \epsilon$.

Corollary: Let $\mathcal{X}$ be an infinite domain set and let $\mathcal{H}$ be the set of all functions from $\mathcal{X}$ to {0,1}. Then *H* is not PAC learnable.

Corollary proof (phrased in terms of Bob's scenario):

- Let $\mathcal{X}$ ={mango: 0≤ softness ≤1, colour between red and green} . Hence, $|\chi| = \infty$.
- Suppose for the sake of contradiction that *H* is PAC learnable.
- Choose ε <1/8 and δ <1/7. Then, there exists a finite number of mangoes, $m(\varepsilon, \delta)$, such that, if we choose the number of mangoes to be $m > m(\varepsilon, \delta)$, then L(Bob's $h \in H$ choice)<1/8, at least $\left(100 * (1 - \frac{1}{7})\right)$% of the time, for all data generation models.
- However, $\frac{|\chi|}{2} = \infty > m$.
- Hence, the no-free-lunch theorem tells us that there exists a data generation model that can trick us. That is, $\exists$ *D*, such that the probability, L(Bob's $h \in H$ choice)<1/8, is at most $100 * \frac{1}{7}$%.
- Leading to the desired contradiction, that is, *H* is not PAC learnable.

Now that Bob has avoided the hazards of the no-free-lunch theorem by choosing a hypothesis class intelligently, he uses the following algorithm to learn.

Prior knowledge: From his experience with other fruits, Bob knows that other fruits lie within an interval sweet spot. He chooses an appropriate set of intervals based on his 300 mangoes to be his hypothesis class.

Learning process: He will pick each hypothesis from his bag, one by one, and test it on his list of 300 mangoes. He will choose the one that produces the least error on his list.

Input: A new random mango.

Output: He will use this chosen hypothesis to tell us if our mango is tasty or not.

Source: Shalev-Shwartz and Ben-David (2014).

Policy Relevance

The no-free-lunch theorem tells us that there cannot be a "one-size-fits-all" approach in machine learning. There can never be a single AI system that is ready to tackle any task. For every task, a hypothesis class must be carefully chosen to suit it.

In practice, there might be cases where one wishes to use another AI system for a task for which the AI system is not intended. In Canada, if such use results in harm, the Artificial Intelligence and Data Act (AIDA) as part of Bill C-27, may help address which party may be liable in such a situation. This is demonstrated with the following case example.

Case example — where two actors are involved:

- Actor one puts a high-impact AI system in the market that performs consistently with its intended objectives.
- Actor two uses actor one's high-impact AI system for a task for which it is not intended (as per actor one's manual).
- If actor two's actions result in harm, then only actor two will be liable under AIDA.

Sample Complexity

Now that the hypothesis class has been presented, the next important topic to introduce is sample complexity (ibid.). Intuitively, sample complexity is the size of a training sample necessary to learn effectively. If Bob were able to taste all the mangoes in the universe, he would essentially be memorizing. However, if he tastes very few mangoes, he will not be able to learn properly. Therefore, this domain investigates where that "sweet spot" is.

In order to answer where the sweet spot for Bob is, the fundamental theorem of statistical learning theory (FTSLT) needs to be used:

- **Level one:** If Bob wishes to come up with a hypothesis/algorithm that has an error less than 0.1 more than 90 percent of the time, then the FTSLT says that he will need to try approximately 50 mangoes.
- **Level two:** More precisely, it is up to some constants $C_1 > 0,\ C_2 > 0$.
  - $C_1 * 50 \leq$ mangoes $\leq C_2 * 50$
- **Level three:** A qualitative version of the FTSLT is as follows:

Theorem (the fundamental theorem of statistical learning): Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to {0,1} and let the loss function be the 0–1 loss. Then, the following are equivalent:

1. $\mathcal{H}$ has the uniform convergence property.
2. Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$.
3. $\mathcal{H}$ is agnostic PAC learnable.
4. $\mathcal{H}$ is PAC learnable.
5. Any ERM rule is a successful PAC learner for $\mathcal{H}$.
6. $\mathcal{H}$ has a finite VC-dimension.

Source: Shalev-Shwartz and Ben-David (2014).
Note: ERM = empirical risk minimization; VC = Vapnik–Chervonenkis.

– A quantitative version of the theorem is as follows:

Theorem (the fundamental theorem of statistical learning — quantitative version): Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to {0,1} and let the loss function be the 0–1 loss. Assume that VC-dimension $(\mathcal{H}) = d < \infty$. Then, there are absolute constants C_1, C_2 such that:

1. $\mathcal{H}$ has the uniform convergence property with sample complexity,
$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}^{UC}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$
2. $\mathcal{H}$ is agnostic PAC learnable with sample complexity,
$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$
3. $\mathcal{H}$ is PAC learnable with sample complexity,
$$C_1 \frac{d + \log(1/\delta)}{\epsilon} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d\log(1/\epsilon) + \log(1/\delta)}{\epsilon}$$

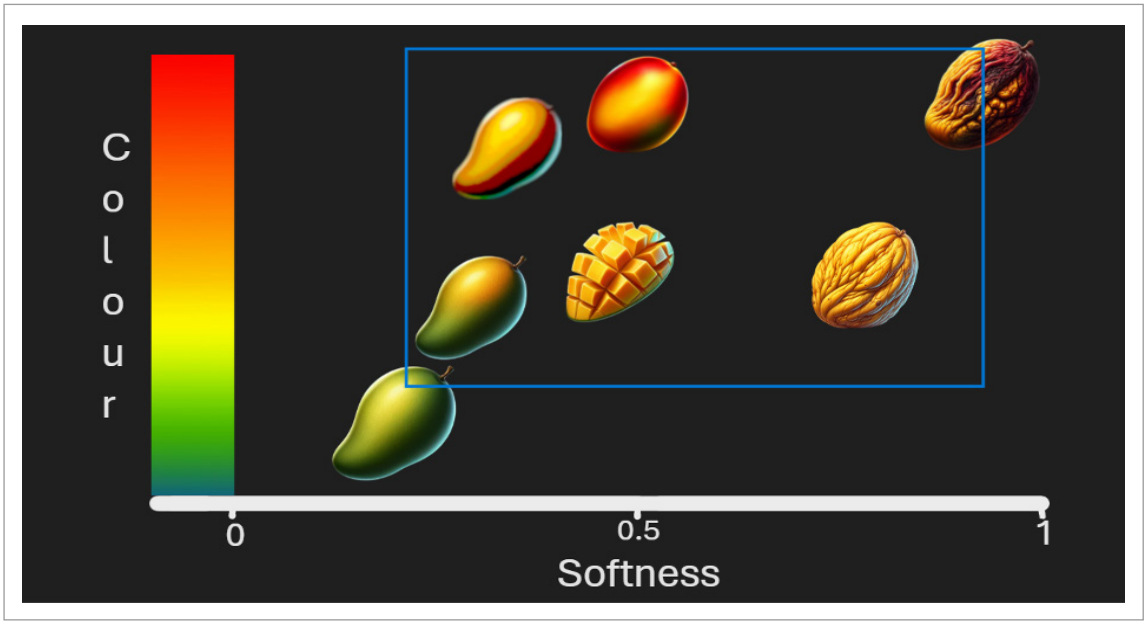
Source: Ibid.

Notice that the PAC formula (equation 3) was used and not the agnostic PAC (equation 2). In actuality, the realizability condition in PAC learning may not be realistic for our mango problem. However, the assumption was that in Bob's universe, the mangoes come from a domain set (Figure 4), where the realizability condition becomes possible.

Intuition for calculating *d* (VC-dimension) for Bob: For two dimensions (tastiness and softness), suppose there are n>4 mangoes distributed on a two-dimensional grid according to tastiness and softness. Choose the rectangle such that its edges lie on left, right, bottom and topmost mangoes. Then, automatically the rest of the mangoes will be

inside this rectangle (see Figure 6). Hence, the cardinality of the set of mangoes that can be shattered cannot be more than four.

Figure 6: Intuition for Calculating *d* (VC-dimension) for Bob



Source: Generated with OpenAI.

Policy Relevance

Sample complexity plays a central role in machine-learning theory. In practice, there are circumstances where the training data is biased or when acquiring enough training data is impossible. Hence, this is especially problematic when such algorithms, which were trained on very small or biased training data, are used to make real-world decisions that have consequences for individuals, groups or other actors.

One of the guiding principles of AIDA is "transparency." That is, the public must be provided with enough information to understand the potential impact, capabilities and limitations of the high-impact AI system. Hence, transparency on the amount and quality of data used to train high-impact AI systems can better help the public gauge the limitations of such AI systems.

Learning-Time Computational Complexity

The final topic to introduce is learning-time computational complexity, which is how long it takes to learn. An important tool for categorizing the run-time of learning is a concept called "NP-hardness."[1] NP-hardness is a topic of its own; hence, this will not be deeply discussed in this paper. However, intuitively, in

1 In computer science, non-deterministic polynomial time (NP) denotes the set of problems for which a solution, if it exists, can be checked quickly. The set of problems that are at least as hard as any other NP problem are also called non-deterministic polynomial time hard (NP-hard). The word "quickly" means polynomial time in this context. NP-hard problems are "hard" in the sense that any problem in the class NP can be converted to an instance of any problem in the class NP-hard in polynomial time. Hence, they are at least as hard as any NP problem.

computer science, if some problem is categorized as "NP-hard," it is bad news. It means that, as of writing this paper, all the possible known approaches to solve that problem will take exponential time, which is very inefficient.

What can be said about the run time of Bob's learning process if he uses the hypothesis class he has chosen?

- **Level one:** It turns out that for the case of learning based on two parameters (softness and colour), Bob's learning process is not NP-hard. Therefore, that is good news, as far as learning time is concerned. However, as the number of parameters based on to learn increases (softness, colour, volume, weight, texture and so on), the time it will take him to learn *most optimally* will get exponentially worse. It is NP-hard.
- **Level two:** Bob's hypothesis class can be modelled as a set of axes-aligned rectangles. That is:
  - $H = \{h(x_1, x_2, y_1, y_2) : x_2 > x_1, y_2 > y_1\}$; such that,
  - $h(a, b) = \begin{cases} \text{"Tasty" if } x_1 \leq a \leq x_2 \text{ and } y_1 \leq b \leq y_2 \\ \text{"Not tasty" otherwise} \end{cases}$
  - Note that it was assumed in Bob's universe that the mangoes are "tasty" when their softness and colour is 0.3–0.7 and yellow to deep orange.
  - The ERM algorithm picks the hypothesis with the least error/loss on the training set.
  - Regardless of whether this assumption holds or not, for two parameters (softness and colour), it can be shown that ERM takes polynomial time.
  - For n parameters, they can be similarly modelled as axes-aligned hyper-rectangles. If a similar assumption for n parameters holds, it can be shown that ERM still takes polynomial time.
- **Level three:** However, for n parameters, if a similar assumption for n parameters does not hold (which is close to reality), implementing ERM is NP-hard (Ben-David, Eiron and Long 2003).

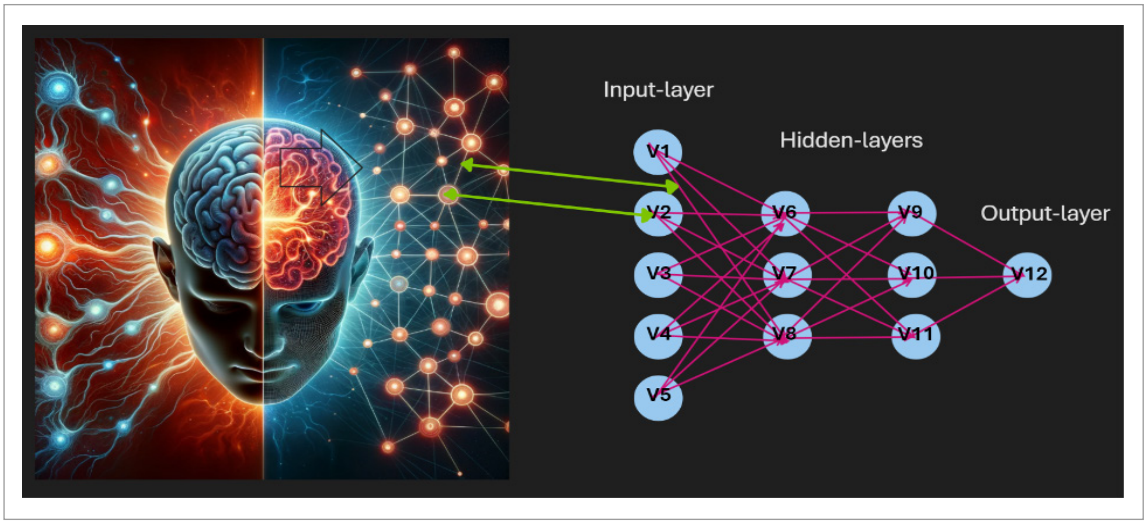
Policy Relevance

The learning time of AI systems starts to pose problems in circumstances where the AI system needs to adapt quickly and effectively, especially in situations where there are immediate real-life consequences. A notable example is self-driving cars. Another one of the guiding principles of AIDA is "safety." That is, for high-impact AI systems, measurements must be proactively taken to mitigate harm. Thus, a proactive testing of such AI systems should help to ensure their safety.

Neural Networks

Now that all the foundations of machine-learning theory have been introduced, how is this applied in practice? The most widely used machine learning algorithms are neural networks. For example, ChatGPT is a neural network. A neural network is perhaps the most similar machine-learning model to human brains (see Figure 7). A neural network has "nodes" that are analogous to neurons and "edges" that are analogous to the connections between neurons. Also, analogous to how different brain areas have a different number of neurons and structures to achieve different functions, so can neural networks. Now, using the tools introduced above, an analysis can be made of neural networks (Shalev-Shwartz and Ben-David 2014).

Figure 7: Neural Network vs. Human Brain



Source: Generated with OpenAI.

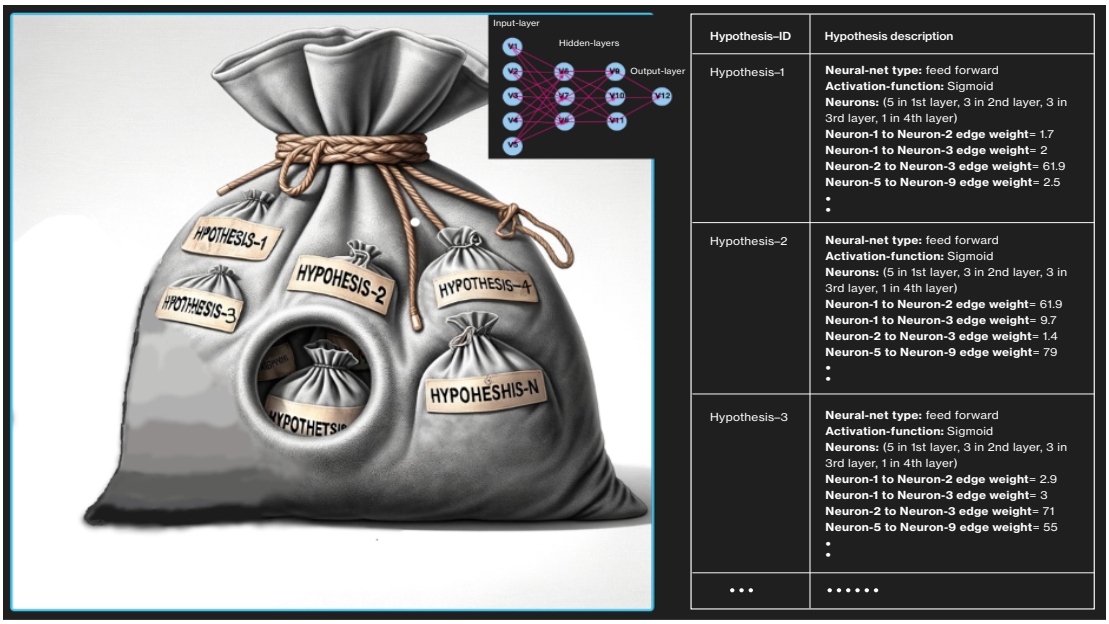
In practice, to make sure the neural network model is not overfitting, it is generally good practice to split the training data beforehand into a training sample and a validation sample. Afterward, the neural network model is trained on the training sample and the validation sample can be used to make sure it is not overfitting — that is, to make sure it can do well on data it has never seen before.

Neural Network Hypothesis Class

- **Level one:** A hypothesis class is selected by choosing a neural network type with all its parameters (number of neurons, layers, activation function, and so on) based on the nature of the problem. However, the edge weights (the strength of connection between two neurons) are not fixed. For example, one type of neural network is a "feed-forward neural network." In a feed-forward neural network, there are no cycles and the information from one layer "feeds-forward" into the next layer.
  - For example, the right image in Figure 7 shows a feed-forward neural network with 12 neurons, 27 edges and four layers.

- An explicit illustration of the hypothesis class this neural network represents is shown in Figure 8.

Figure 8: An Explicit Illustration of a Neural-Net Hypothesis Class



Source: Generated with OpenAI.

- **Level two:** Now consider feed-forward neural networks. Each neuron (V) is just some function, $\sigma: \mathbb{R} \to \mathbb{R}$. We call σ, the "activation function." Each edge (E) is just a number called "edge weight," $w \in \mathbb{R}$. A hypothesis class is defined by fixing the neural network graph, $G(V, E)$, and the activation-function (σ), but not the edge weights. That is, two hypotheses in a hypothesis class have the same underlying graph structure, but only differ in the edge weights. Hence:
  - $H_{v,E,\sigma} = \{h_{v,E,\sigma,w}: w \text{ maps each edge } (E) \text{ to } \mathbb{R}\}$.
  - Then, the expectation is that some combination of edge weights is a good fit for the learning task.

Expressive Power of Neural Networks

What are neural networks able to do?

- **Level one:** A type of neural networks called RNN (recurrent neural networks) are capable of implementing any task that can be fulfilled by a computer, given enough time and memory. The size of the neural network hypothesis class depends on the complexity of the task at hand, with more complex tasks requiring larger hypothesis classes. In practice, however, factors such as time, memory, quality of training data and so on, also play a significant role in limiting how large our neural network can be and the neural network's performance after training. Hence, these practical limitations can prevent RNNs from reaching their full theoretical potential.

- **Level two:** The feed-forward neural network type introduced previously, although not as diversely capable as RNNs, nevertheless has the capacity to implement any function, $f: \mathbb{R}^m \to \mathbb{R}^n$.
- **Level three:** RNNs are Turing-complete (Chung and Siegelmann 2021). A more quantitative statement of the capability of feed-forward neural networks is as follows:

> **Theorem:** Let $T: \mathbb{N} \to \mathbb{N}$ and for every n, let $\mathcal{F}_n$ be the set of functions that can be implemented using a Turing machine using runtime of at most $T(n)$. Then, there exist constants $b, c \in \mathbb{R}_+$ such that for every n, there is a graph (V_n, E_n) of size at most $cT(n)^2 + b$ such that $\mathcal{H}_{V_n,E_n,\,sign}$ contains $\mathcal{F}_n$.

Neural-Network Sample Complexity

- **Level one:** The bigger the neural network, the more data needed to train it well. However, as the neural network gets larger, it does not require an extremely high (exponential) amount of data to train it. The amount of data needed scales up at a relatively fair rate, which is one of the many reasons neural networks are a popular choice for machine learning.
- **Level two:** The sample complexity is a polynomial function of the number of edges in the neural network. Hence, it is not exponential, which is good.
- **Level three:** If $H_{v,E,\sigma} = \{h_{v,E,\sigma,w}\colon\ w: E \to \mathbb{R},\ \ \sigma: a \to \text{sign}(a), \text{single neuron output}\}$. Then, the VC-dimension is $O(|E|\,log(|E|))$. If the sigmoid function is used for σ instead, then $O(|v|^2) \leq$ VC-dimension $\leq O(|v|^2|E|^2)$.

Neural-Network Learning-Time Computational Complexity

- **Level one:** This is one area where neural networks fall short. Training neural networks to find the most optimal edge weights is NP-hard. However, we can still get near optimal edge weights. That is, learning can be fast using a neural network, but once the neural network is trained, there is no guarantee that it is the most optimal. This is still an open area of research. Nevertheless, although not the most optimal, the algorithms that exist for training neural networks are still able to give good results for practical purposes.
- **Level two:** Methods called backpropagation and stochastic gradient descent can be used to train neural networks near optimally.
- **Level three:** There is a strong indication that trying to change the architecture and activation function to get around this hardness result is doomed to fail.

> **Theorem:** Let $k \geq 3$. For every n, let (V, E) be a layered graph with n input nodes, $k + 1$ nodes at the (single) hidden layer, where one of them is the constant neuron, and a single output node. Then, it is NP-hard to implement the ERM rule with respect to $\mathcal{H}_{V,E\,sign}$.

Canada's Draft AI Law

In June 2022, as part of the Digital Charter Implementation Act (Bill C-27), Canada introduced AIDA, which has not yet passed into law as of the writing of this paper. In earlier sections, this paper discussed how potential problems posed by AI could be addressed by AIDA. Below is a formal summary of what AIDA entails.

Under AIDA, in order to precisely identify systems that need to be regulated, "high-impact" systems are defined during regulation. However, the following are some of the key factors the government uses to determine if a system is high impact: AI's severity of potential harm; AI's scale of use; any evidence of impact on human rights; and any evidence of harm to the health and safety of individuals. Two types of adverse impacts are considered under AIDA: "harm" and "biased output." AIDA defines "harm" as any physical and/or psychological harm, economic loss to an individual, or damage to property. Under AIDA, "biased output" is defined as any unjustified differential impact based on the prohibited grounds for discrimination under the Canadian Human Rights Act.[2]

AIDA intends to protect Canadians against adverse effects of high-impact AI systems by establishing the following principles to guide the obligations for high-impact AI systems (see Table 4):

Table 4: AIDA Principles

AIDA Principle	Obligation
Human oversight and monitoring	A high-impact AI system should be designed, such that people who operate the system can exercise oversight.
Transparency	The public must be provided with enough information to understand the potential impact, capabilities and limitations of the high-impact AI system.
Fairness and equity	High-impact AI systems must be built with the awareness of their potential for discrimination of individuals or groups.
Safety	Measurements must be proactively taken to mitigate "harm."
Accountability	Organizations covered by AIDA must make sure they are complying with legal obligations of high-impact AI systems.
Validity and robustness	A high-impact AI system must perform in a stable manner consistent with its objective.

Source: Author.

Enforcement of AIDA

The Canadian government has stated that it will give businesses enough time to adjust to the new bill. Therefore, since its introduction, AIDA's focus has been on education and to help businesses comply through voluntary means. The following are the intended enforcement mechanisms of AIDA:

- Administrative monetary penalties — These are means by which regulators can respond to any violation in a flexible manner.

2 See *Canadian Human Rights Act*, RSC 1985, c H-6.

- Regulatory offences — These are means by which providers can be prosecuted in a more serious case of non-compliance.
- True criminal offence — These are when the providers intentionally cause serious harm.

Case Example

A case where multiple actors are involved:

- Actor one: A group of researchers develop a model capable of developing AI systems.
  - There is no legal liability under AIDA, since there is no commercial activity.
- Actor two: Company A uses actor one's model to develop a high-impact AI system and put it on the market for consumer use.
  - Company A will need to comply with *development and making available for use* requirements of AIDA.
  - If company A placed the system on the market for use, knowing it can cause serious harm, company A can be prosecuted for a criminal offence under the Criminal Code.
- Actor three: Company B uses company A's high-impact AI system for its own commercial use and manages this AI system on its own.
  - Company B will need to comply with *managing-operations* requirements of AIDA.
  - If company B shows reckless activity with regard to safety while operating the AI system, company B can be prosecuted for a criminal offence under the Criminal Code.

Conclusion

There is no doubt that machine learning is a powerful tool that has many positive applications. However, just like other technologies, it comes with limits. Some of these limitations discussed in this paper are the no-free-lunch theorem, learning time of machine-learning algorithms, the amount of training data and the quality of training data. Therefore, as the use of machine-learning continues to boom, recognizing its limits in several domains where it is applied will be critical to ensuring its safety and efficacy — especially in high-impact situations, where an inappropriate use of this technology could result in harm to individuals or groups. The Canadian government introduced AIDA as part of Bill C-27 in June 2022. This paper has discussed how AIDA helps address some of the problems that can arise from these limitations. From the academic machine-learning community's side, there is also worry that such laws could be too restrictive. This is addressed as a case example to demonstrate how AIDA would affect the machine-learning academic community versus industry.

Acknowledgements

I would like to thank Ambika Opal, David Williams, Nestor Maslej, Paul Samson and Reanne Cayenne. Thank you for your guidance and/or edits!

About the Author

Naod Abraham is a third-year mathematical physics student at the University of Waterloo with an interest in computer science. In high school, Naod published a paper on the subject of computational complexity theory and machine learning in a peer-reviewed journal. During his fellowship at the Digital Policy Hub, he will further investigate the applications of machine learning for other practically important problems.

Works Cited

Barnett, Samuel Anthony. 1963. *The Rat: A Study in Behavior*. Abingdon, UK: Routledge.

Ben-David, Shai, Nadav Eiron and Philip M. Long. 2003. "On the Difficulty of Approximately Maximizing Agreements." *Journal of Computer and System Sciences* 66 (3): 496–514.

Chung, Stephen and Hava Siegelmann. 2021. "Turing Completeness of Bounded-Precision Recurrent Neural Networks." *Advances in Neural Information Processing Systems* 34: 28431–41.

Garcia, John and Robert A. Koelling. 1966. "Relation of cue to consequence in avoidance learning." *Psychonomic Science* 4: 123–4.

Government of Canada. 2023. "The Artificial Intelligence Data Act (AIDA) – Companion document." Innovation, Science and Economic Development Canada. https://ised-isde.canada.ca/site/innovation-better-canada/en/artificial-intelligence-and-data-act-aida-companion-document.

Shalev-Shwartz, Shai and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. New York, NY: Cambridge University Press.

Skinner, Burrhus Frederic. 1948. "'Superstition' in the Pigeon." *Journal of Experimental Psychology* 38 (2): 168–72.