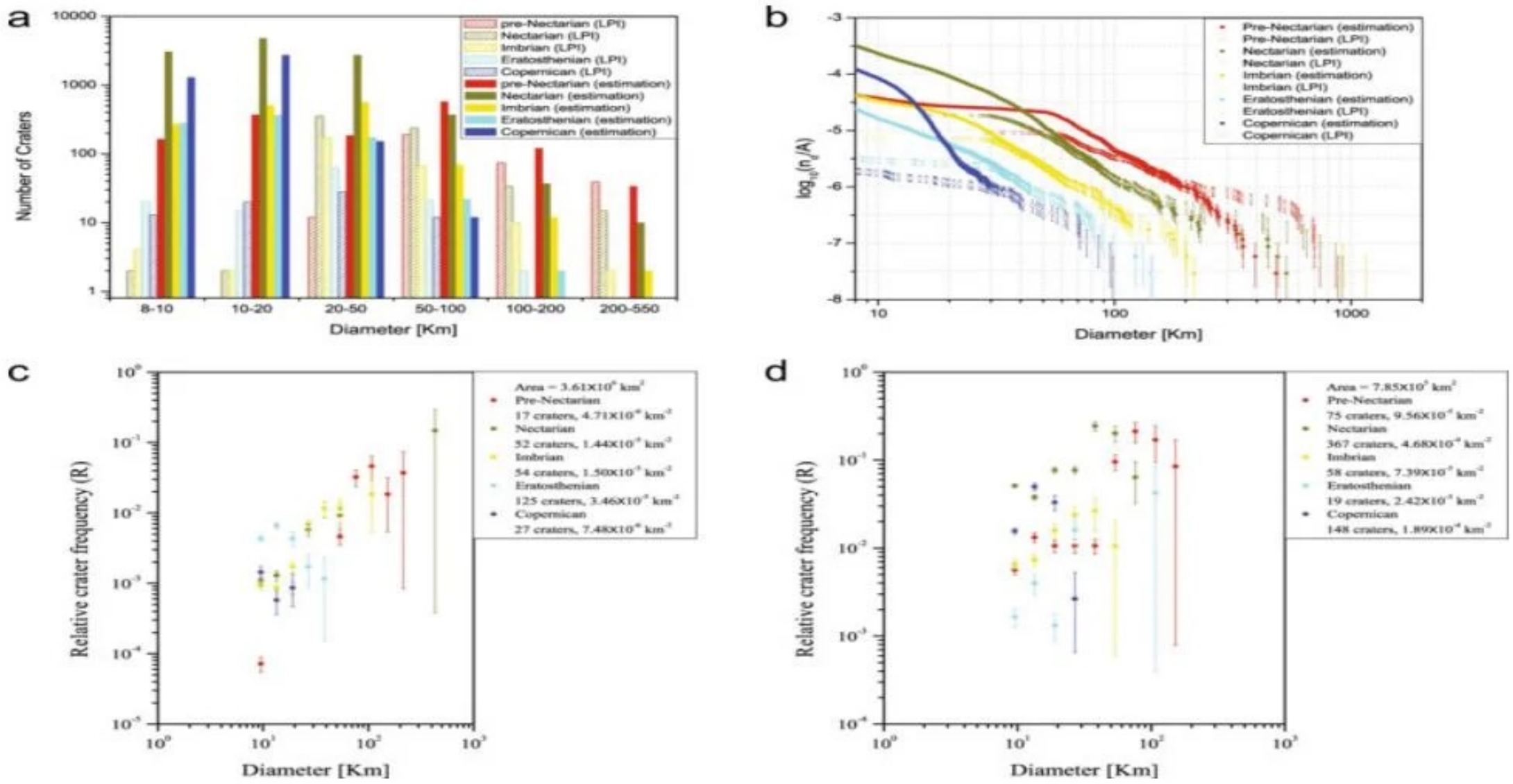


Data-Driven Behavior Change: Machine Learning and Its Impacts on Other Industries

BY AMITTAI WEKESA '24

Cover: A visualization of data on lunar impact craters, analyzed with machine learning and deep learning. Machine learning offers efficient analysis of massive datasets.

Source: Wikimedia Commons

Introduction

Machine learning can be an elusive term, often known for being a subfield of artificial intelligence (AI). Artificial intelligence can generally be subdivided into five fields: machine learning, deep learning, computer vision, robotics, and natural language processing. Deep learning is more specialized than machine learning, extending neural networks, a machine learning model, to more sophisticated structures which usually yield better predictions. Computer vision and natural language processing are specific to the areas of image and language processing. On the other hand, robotics usually incorporates the other AI subfields for stimulus detection and response in order to make intelligent decisions in automated machinery and robot systems. Evidently, all subfields drive each other and are not clearly defined without one another; machine learning must be understood in the broader context of artificial intelligence.

A History of Artificial Intelligence

Developments in artificial intelligence theory started as early as the 1930s with Alan Turing and Kurt Gödel. In 1931, Gödel, an Austrian logician, developed a mathematical proof that, although all true statements are derivable in first-order systems, some are unprovable in higher-order systems. This called to attention the common practice scientists and mathematicians had at the time of extending simple first-order arithmetic axioms to higher-order systems (Ertel & Black, 2018). This development was a step towards understanding the possibilities of infinite choice and how axioms from simpler systems can and cannot be extended to more complex setups. On the other hand, Alan Turing's proof that no general algorithm can predict all possible program-input pairs for the halting problem (a problem of predicting whether a program will finish running or continue running forever based on a description of the program)

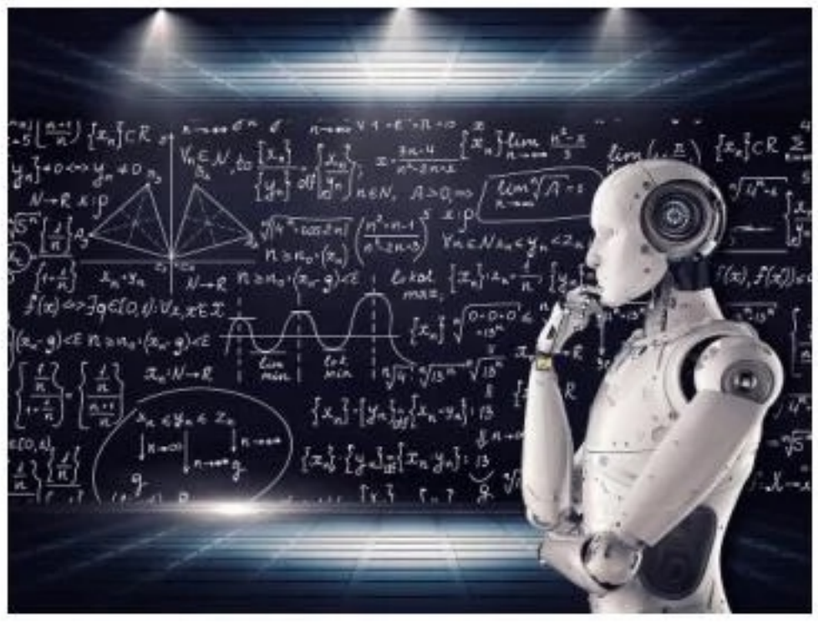


Figure 1: People may compare artificial intelligence to human intelligence, but there are key differences between AI and human intelligence.

Source: Flickr

also went towards defining the computational limits that any intelligent machine would have to contend with (Ertel & Black, 2018). In 1943, McCullogh and Pitts modeled the first-ever theoretical neural network, a predictive structure that is still used in machine learning and deep learning systems today. However, due to computational limitations, they could not build a fully functional model at the time (López-Cajún & Ceccarelli, 2018).

Despite these theoretical formulations, artificial intelligence was still a merely academic field comparable to mathematics—theoretical proofs with no practical developments yet. It was an undefined field regarded under Automata Studies, a broader area that included the study of general automata theory not directly linked to artificial intelligence (Shannon & McCarthy, 1956). While this generalization and the underdeveloped technology of the time hindered focused research in artificial intelligence, the field still found its way into numerous symposiums and conferences, such as the 1948 Hixon Symposium on Cerebral Mechanisms in Behavior held at Caltech, which inadvertently found itself comparing the computer and the brain, albeit theoretically.

The first stored-program computers were pioneered in 1949. A year later, Marvin Minsky and Dean Edmunds, two Ph.D. students at Harvard, designed a simple neural network—a simple deterministic prediction engine—long before the formal concept of the neural network was even conceptualized. Claude Shannon, a computing professor at The Massachusetts Institute of Technology, proposed a chess program and built many relay systems demonstrating intelligence features (Sejnowski, 2018). In 1955, Arthur Samuel from IBM made a checkers program that learned to play better than its inventor, the first-ever program to achieve this feat (Rada, 1986, p. 119). Other pioneers emerged with their new ideas and developments. The same year, Allen Newell and Herbert Simon from Carnegie Mellon University created the Logic Theorist, a computer program that could prove the logical theorems in Principia Mathematica, a sophisticated mathematics textbook. However, a lack of a coordinated effort to pursue artificial intelligence hindered progress in the field for a while (Sejnowski, 2018).

In 1956, Claude Shannon proposed a machine intelligence workshop that brought together many pioneers in the field, including Shannon, Minsky, Nathaniel Rochester, and John McCarthy. This workshop, held at Dartmouth College and broadly known as the "Dartmouth Workshop," became the first collaborative effort in AI and even coined the term "artificial intelligence" (McCarthy, 2006). McCarthy went on to establish an AI lab at Stanford University and to invent LISP, a high-level programming language that became a standard tool in developing machine learning and AI systems.

"In 1962, Frank Rosenblatt published Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. This book sought to go beyond the biology of brain matter and explore how human intelligence works."

In 1962, Frank Rosenblatt published Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. This book sought to go beyond the biology of brain matter and explore how human intelligence works. The same year, David Hubel and Torsten Weisel published Receptive Fields, Binocular Interactions, and Functional Architecture in the Cat's Visual Cortex, a book that, for the first time, analyzed the response properties of individual neurons (Rada, 1986). These books offered interesting perspectives to early developers in artificial intelligence, which, even today, is often developed to mimic the processes of brain functionality in some capacities. Even then, scientists were only just beginning to understand AI's capabilities and limitations.

By 1972, French scientist Alain Colmerauer invented yet another logic-based

Figure 2: A recreation of an Abacus machine, one of the earliest computational devices. While such a device supported basic mathematical computations, it could not support sophisticated machine learning and AI algorithms.

Source: Wikimedia Commons

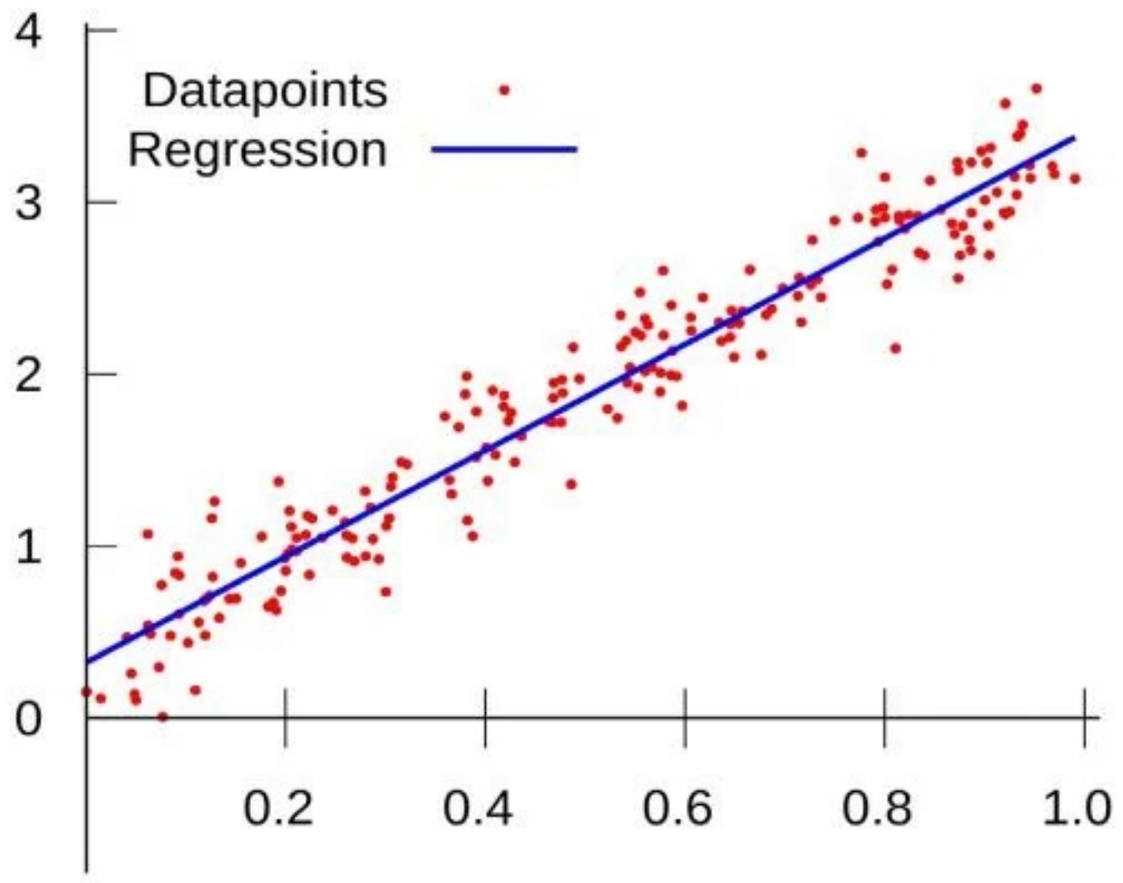


Figure 3: A simple linear regression model where the data points are mapped out onto some spatial space based on a specific feature (such as house prices). Then, a line of best fit (or regression) is determined. Predictions can then be made on new data points depending on where they lie on the generated model.

Source: Wikimedia Commons

"From the awe-inspiring robotics soccer leagues from 2003 to Google's first self-driving cars in 2009, to Daimler's fully autonomous truck pioneered on the Autobahn in 2015, advancements in the field of AI have increased in pace due to high computational power in the current age."

programming language, PROLOG, which sought to address the shortfalls of LISP and other programming languages (Ertel & Black, 2018). The new language provided advantages that encouraged people to pursue projects in computing. For example, around nine years later, Japan embarked on the "Fifth Generation Project" to develop the first fifth-generation computer based on PROLOG.

In 1990, Pearl, Cheeseman, Whittaker, and Spiegelhalter officially brought probability theory into artificial intelligence with Bayesian networks (Sejnowski, 2018). With Bayesian models, developers could design algorithms that consider probabilistic uncertainties in the inputs and outputs, resulting in better predictions with compensation for possible errors. This model popularized multiagent systems and opened the door for new understanding and developments in the field. In 1992, yet another machine learning subfield, reinforcement learning, received its first significant validation when Gerald Tesauro's Temporal Difference Learning algorithm demonstrated the superiority of reinforcement learning over traditional supervised learning (Tesauro, 1995). Another reinforcement learning program—a chess program called Deep Blue—defeated chess grandmaster Garry Kasparov in 1997 (Kasparov, 1998).

Advancements in the 2000s and 2010s were mainly in robotics, deep learning, and reinforcement learning, which have risen to be some of the most critical AI fields. From the awe-inspiring robotics soccer leagues from 2003 to Google's first self-driving cars in 2009, to Daimler's fully autonomous truck pioneered on the Autobahn in 2015, advancements in the field of AI have increased in pace due to high computational power in the current age. In 2016, AlphaGo, a reinforcement learning algorithm developed by Google DeepMind, beat the European Go champion 5:0 and the Korean champion 4:1 in 5 games each. This feat showed the infinite power of machine intelligence because Go has trillions of possibilities, higher than other board games such as chess.

Machine Learning Revisited

But what is machine learning, and can a machine actually learn? By itself, "machine learning" refers to computer algorithms parsing and learning from data to build a prediction model, and then using the model to extrapolate to general cases from the test data and make informed decisions (Grossfeld, 2020). Today, machine learning drives most of the typical AI systems around us. For example, on-demand music streaming services such as Spotify use machine learning to create prediction models around specific kinds of music. In this way, Spotify's algorithms only need to compare a user's music taste and usage trends to its recommendation database to determine which music to recommend and when to recommend it. However, there is no magic behind the predictions, as we like to think about machines being "intelligent." There are four fundamental types of machine learning algorithms depending on the training mechanisms: supervised learning, unsupervised learning, reinforcement learning, and recommender systems.

The first type is supervised learning, which means that algorithms are fed labeled data with varying conditions and taught to internalize

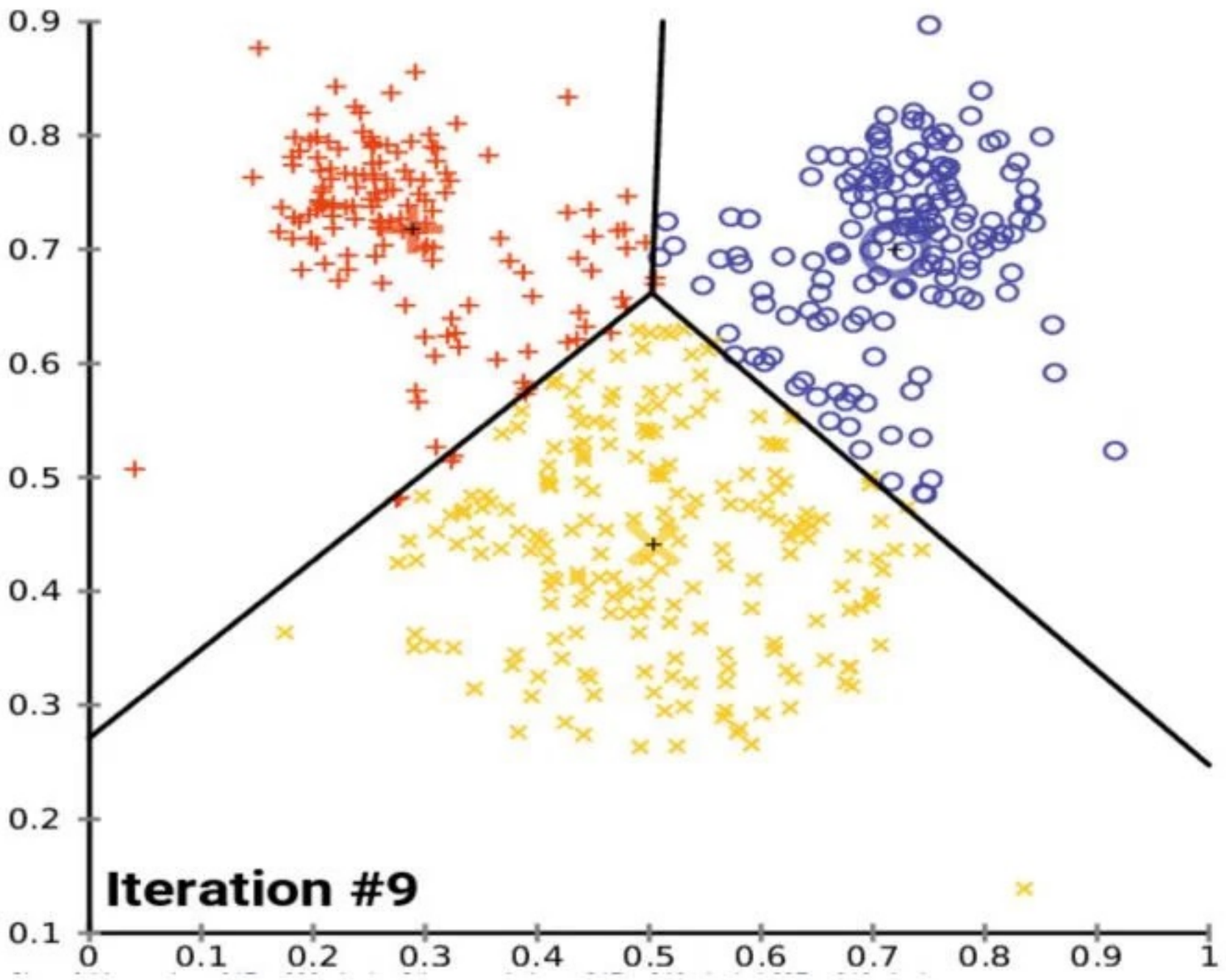


Figure 4: A basic representation of k-means clustering (with k = 3). The machine learning algorithm maps the data depending on specified features and subdivides the data into k groups.

Source: Wikimedia Commons

the labels and their associations to different features in the data. Afterward, the algorithms can make similar predictions on unlabeled data by identifying features and recalling what label was associated with those features (Ertel & Black, 2018). For example, a supervised learning algorithm can use linear regression to predict housing prices in an area knowing that houses with more rooms will cost more. However, this will usually be a very simplistic and objective model. It is common to model with multiple variables and develop a model involving multiple weighted linear regression sub-models for different features such as the number of rooms, the house's square footage, the house's age, and the house's location. It is important to note that supervised learning (and every other machine learning approach) does not involve machines understanding what is going on behind the given data. For example, in the linear regression model mentioned above, the model does not care why, for instance, houses on the Eastern end of the city cost less than those on the Western end. As a result, when a machine learning algorithm, for instance, trains on data from one region, the model it develops can be very inaccurate for data from other areas or time periods if the model is not retrained. Thus, it is essential to keep in mind the subjectivity of data in the functionality of machine learning models (Sejnowski, 2018).

Unsupervised learning is akin to supervised learning: it is also pattern recognition but is from unlabeled data as opposed to labeled data. A common way of achieving this is through clustering methods such as k-means clustering. In k-means clustering, a prediction engine seeks to identify related groups in a dataset instead of individual labels. The algorithm looks at different features in the data and arbitrarily creates k groups. It is then easier for predictions to be made for a single data-point (such as a user in the system) based on the features common among users in the same group as the specific user (Ertel & Black, 2018).

Reinforcement learning is arguably the more human-like method of learning. It is akin to when humans do something, process feedback, determine how favorable the outcome was, and then use the information to decide better in similar situations in the future. Likewise, reinforcement-learning systems use weighted costs to determine the favorability of different outcomes in order to build a comprehensive model (Sejnowski, 2018). A system learning how to play chess can simulate billions of games against people or pre-built game sequences. At first, the system will make arbitrary and uninformed decisions, but over time it will learn which moves have the most favorable outcome in every possible position, becoming nearly perfect. An example of a reinforcement-learning system is DeepMind's AlphaGo program, a gaming engine that learned to play Go better than professional gamers (Schrittwieser et al., 2020). Go, the board game has 250150 or 10360 possible move variations, giving it unprecedented complexity—and

"Reinforcement learning is arguably the more human-like method of learning. It is akin to when humans do something, process feedback, determine how favorable the outcome was, and then use the information to decide better in similar situations in the future."

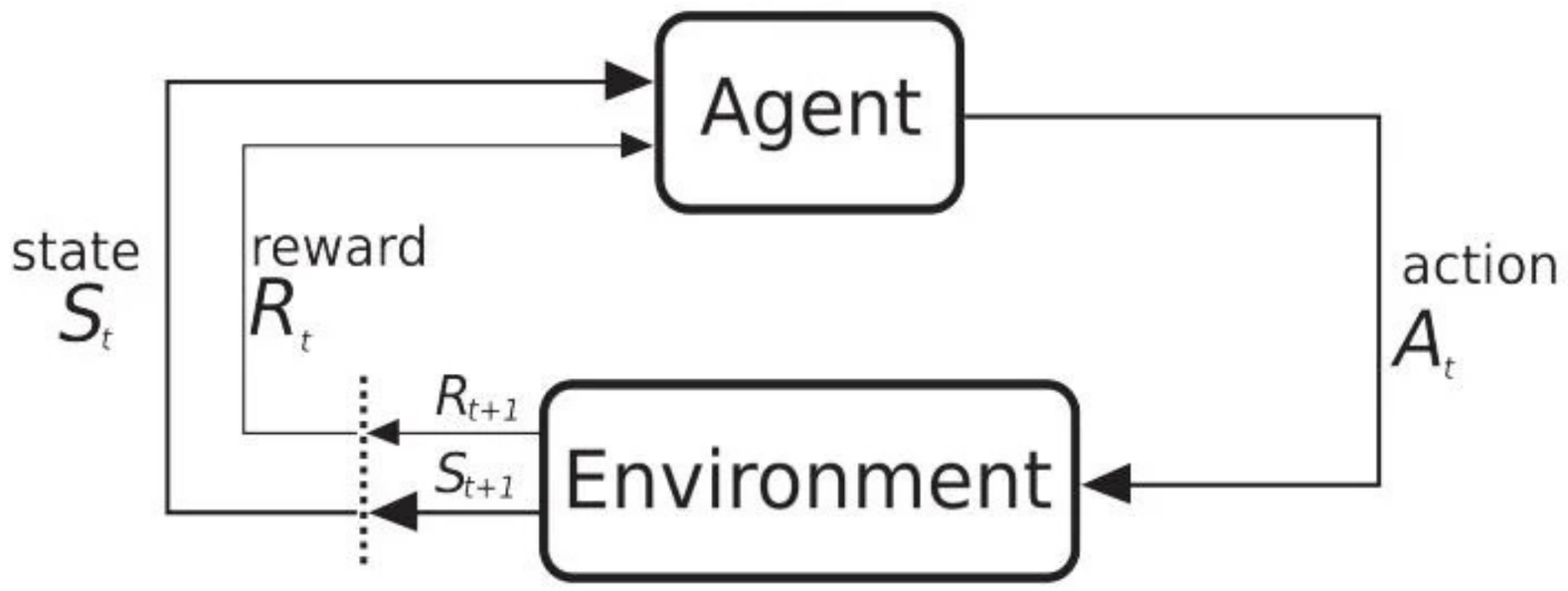


Figure 5: A basic representation of reinforcement learning in which an agent interacts with its environment, creating a new state and a measure of reward. The measure of reward determines how the system reacts in the immediate state and in similar situations in the future.

Source: Wikimedia Commons

the algorithm mastered every last one move variation (Koch & Koch, 2016). After losing two consecutive games to AlphaGo, Lee Sedol, then the world's second-best Go player, admitted in an interview: "From the very beginning of the game, there was not a moment in time when I felt that I was leading" (Metz, 2017).

"The final classification of machine learning algorithms is recommender systems. These usually involve both labeled data (supervised learning model) and unlabeled data (unsupervised learning model) but differ from the two learning models in that the learning is continuous."

The final classification of machine learning algorithms is recommender systems. These usually involve both labeled data (supervised learning model) and unlabeled data (unsupervised learning model) but differ from the two learning models in that the learning is continuous (Ertel & Black, 2018). For example, for Netflix's movie recommendation system, due to the fluidity of movie tastes (a user liking multiple disparate movie titles) and high variance ("similar" users on a metric having different tastes in other metrics), recommender systems cannot afford to use a predetermined model for recommendations. Instead, user preferences are continuously tracked and the recommendations are updated as frequently as needed. Models like k-means clustering are helpful in making predictions in a static temporal point, but the data changes with time so the model must change to maintain accuracy.

Machine Learning in Other Industries

Most aspects of modern life are, in some way, being changed by machine learning. With advancements in machine learning, Netflix can cater specific movies to its viewers, Google can suggest search terms based on user search histories, and Facebook can tune advertisements to specific users. In addition, machine learning holds the promise to truly revolutionize other industries beyond such mundane applications. The adoption of machine learning and artificial intelligence is not always as straightforward as people may think; any company needs to do a comprehensive analysis of its organizational, leadership structures, and workflows, and then they can plan ahead for how a machine learning system would fit in with the existing systems. This is usually a challenge for those companies not in technology-oriented industries, as they usually need to create entire departments, upgrade their systems, and hire specialists to support the new system. Such challenges make some companies and industries opposed to this new change (Char et al., 2018).

Healthcare is an example of an industry that has rapidly adopted machine learning in daily operations. Machine learning offers various advantages over traditional systems; first, it leads to improved efficiency by helping hospitals and health organizations manage data and make well-informed decisions from data trends. Recently, diagnostic algorithms have been found to be as good as, if not better than, human physicians at detecting and predicting the presence of cancerous cells (Beam & Kohane, 2018, p. 1317).

Education has also seen improvements from the use of machine learning. For example, Khan Academy, a popular online-learning website, employs machine learning to rate students on various "skills" based on their performance in periodic assessments. These ratings are then fed into a machine learning system, which determines what content users need to review and which content they have mastered, effectively creating a study roadmap for each individual student in the system. Through machine learning, Khan Academy is able to offer a level of personalized study that even real-world classes have not achieved (Alenezi & Faisal, 2020). Additionally, machine learning itself is a tool for learning. For instance, popular language translation websites such as

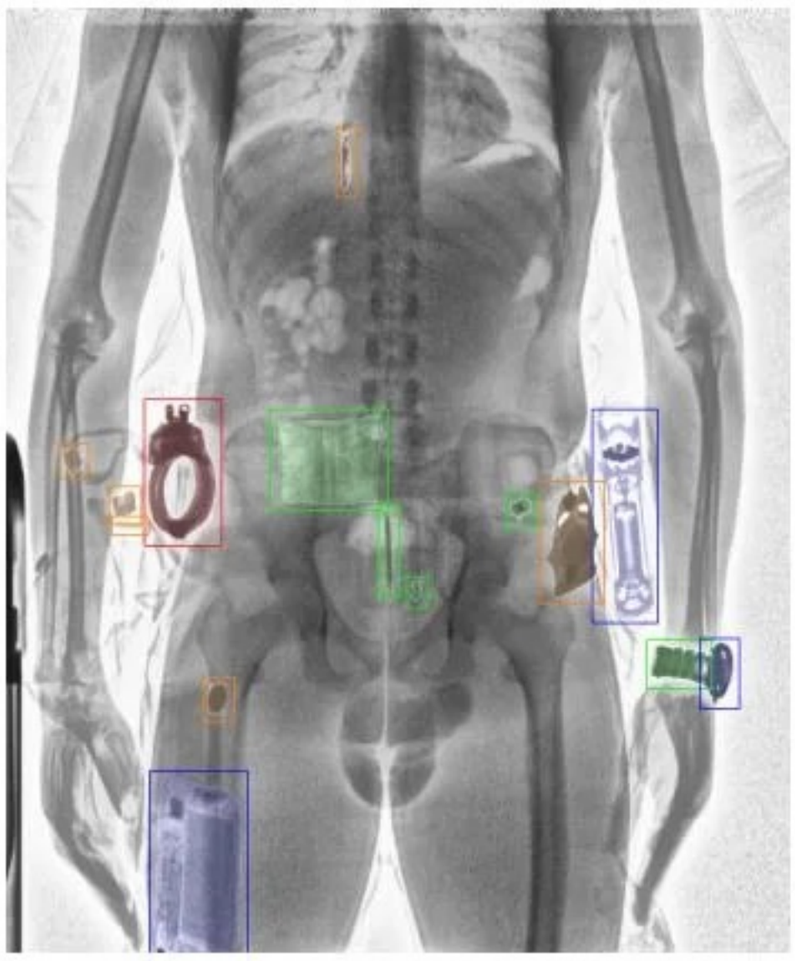


Figure 6: A machine learning model in use to interpret x-ray scans. Diagnostics are one of the emerging applications of machine learning systems in healthcare.

Source: Wikimedia Commons

Google Translate have a huge dataset on the structure and usage of different languages and can analyze a sentence from one language to make a context-aware translation to a different language. This might also allow people who speak different languages to have a conversation without the need for a translator. Indeed, machine translation has multiple advantages over human translators. Machine translation systems are readily available wherever and whenever needed, most are multilingual (in fact, Google translate knows a total of 109 different languages, a feat unattainable to human translators), and machine translation is faster than human translation (Niño, 2009, p. 255).

In agriculture, an entirely new field of research and practice has emerged due to the use of machine learning. Agriculture plays an important role in most economies. With increases in population, global warming, and volatile climate conditions, it is becoming harder for economies to sustain their domestic and export food needs. Termed "precision agriculture," an application of machine learning aspires to overcome this challenge by studying every variable in an area—rainfall, soil type, temperature, presence of pests, etc.—to help optimize the agricultural process and decisions in the area and ensure maximum produce (Sharma et al., 2021, p. 4868). Similarly, animal fertility, breeding patterns, growth patterns, and behavior can be tracked to calibrate the most optimal mating time, ensuring the best outcome for farmers. Every variable can and is tracked, making it easy to diagnose a problematic season and single out the environmental variable or the decision that led to the problem.

Governments, courts, and other administrative agencies are often faced with the need to analyze massive datasets and informed decisions that might affect millions or even billions of people. It becomes a challenge to comprehensively analyze all the data involved and identify every relevant feature and trend. Thus, many organizations turn to machine learning (Coglianese & Lehr, 2017). With machine learning, all the data does not have to be manually processed; instead, one only needs to feed it into the algorithm and tell it which kinds of features and trends to look for. This saves a lot of time, allowing offices to operate faster and more efficiently.

"Governments, courts, and other administrative agencies are often faced with the need to analyze massive datasets and informed decisions that might affect millions or even billions of people. It becomes a challenge to comprehensively analyze all the data involved and identify every relevant feature and trend. Thus, many organizations turn to machine learning."

The transport industry has also seen

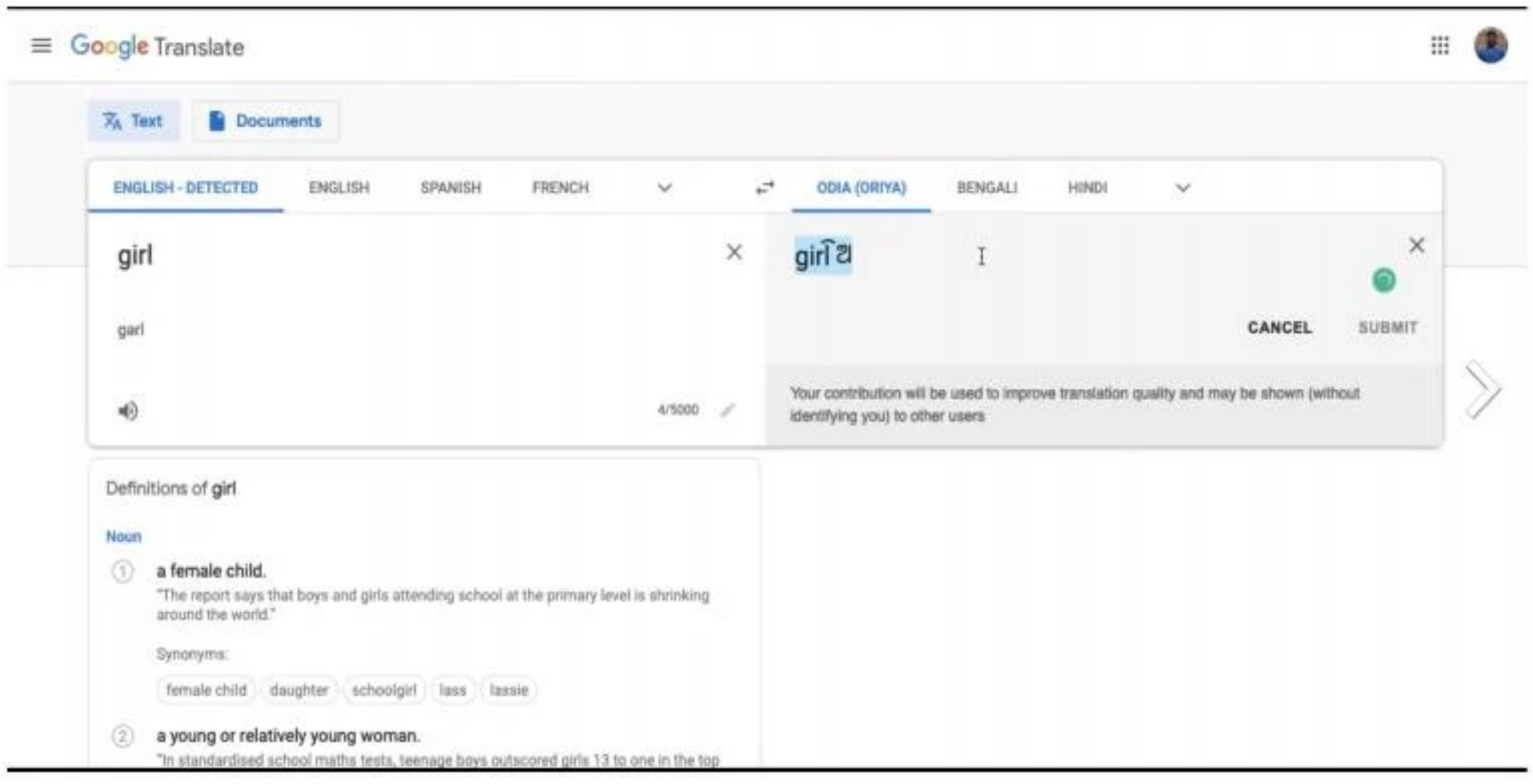


Figure 7: Google Translate, a popular machine translation engine that uses machine learning to detect a language and translate a phrase into any of 108 other languages.

Source: Wikimedia Commons

Figure 8: An autonomous car's "mind": it uses machine learning to detect different objects, lane markings on the road, pedestrians, and more.

Source: Wikimedia Commons

"Although machine learning offers advantages over traditional systems in many industries, it is never perfect. Ceding more control to algorithms should be done with the awareness that these systems offer no guarantees of fairness, equitability, or even veracity."

improvement in efficiency due to machine learning and AI. On the manufacturing end, machine learning is often used in optimizing fluid dynamics. Fluid dynamics, the study of the flow of liquids and gases, helps shape vehicles and airplanes to reduce air drag. Fluid dynamics is particularly important in race car and aircraft design, but it is also finding its way into the design of regular cars as electric car companies such as Porsche and Tesla try to maximize the mileage a user gets out of a single charge. On the user's end, machine learning is embedded in multiple products: navigation services such as Google Maps use machine learning to estimate traffic weight along different routes and suggest the fastest route from one place to another (Lau, 2020). This is particularly important in cities that experience traffic congestion, where the shortest route is not necessarily the fastest one. Additionally, autonomous cars, which heavily rely on machine learning and other AI algorithms, promise to be a better alternative to traditional cars in the future (Haydin, 2021).

Pitfalls of Machine Learning

Although machine learning offers advantages over traditional systems in many industries, it is never perfect. Ceding more control to algorithms should be done with the awareness that these systems offer no guarantees of fairness, equitability, or even veracity. Machine learning systems actually introduce a new set of challenges that most organizations are usually ill-equipped for. One such problem is AI bias, which can happen when a model is trained with very few parameters and ignores other important parameters. Models with high bias make consistent but inaccurate predictions. High bias might also appear where a model that was properly trained is tasked to make predictions on a similar but fundamentally different dataset, resulting in consistent but inaccurate extrapolations (Char et al., 2018). For example, a system designed to make health predictions on local residents in an area should not be used in a different area without retraining because residents in the second area may not share the physiological idiosyncrasies that residents in the first area had.

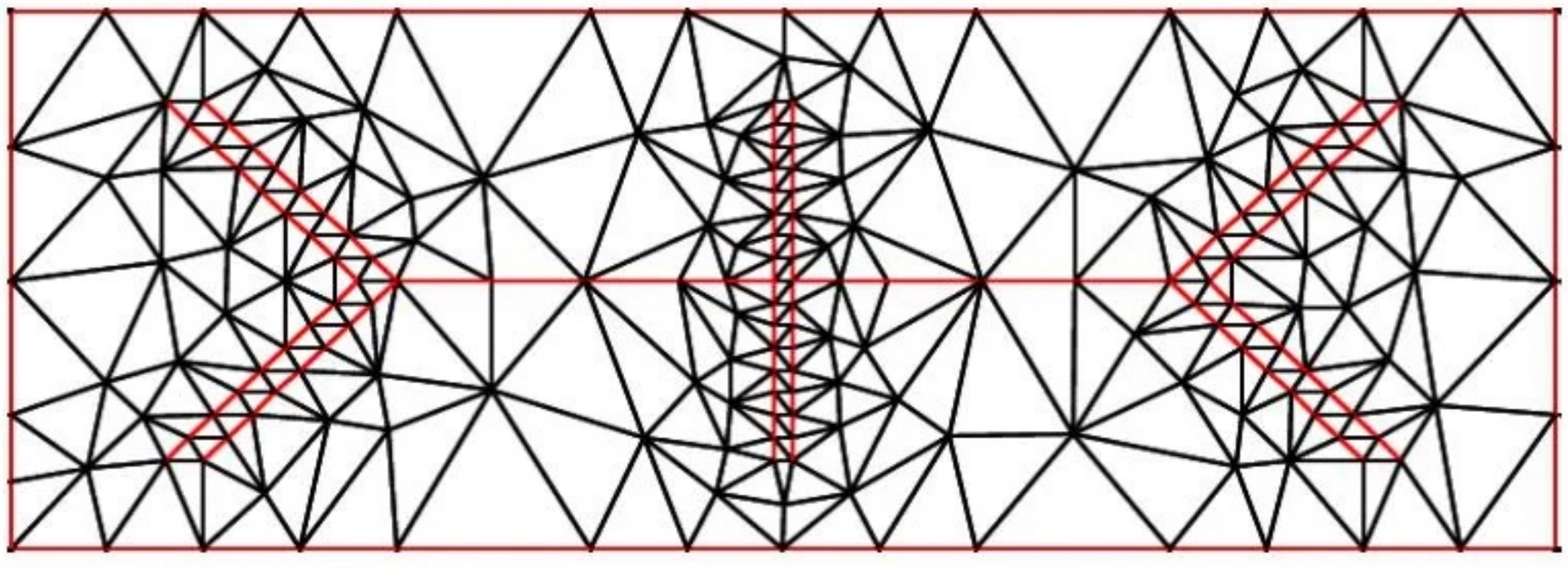


Figure 9: A visual imagination of an algorithm. The thought of complicated software algorithms deciding which people deserve which jobs and opportunities can be daunting, more so the seeming randomness of unexpected predictions.

Source: Wikimedia Commons

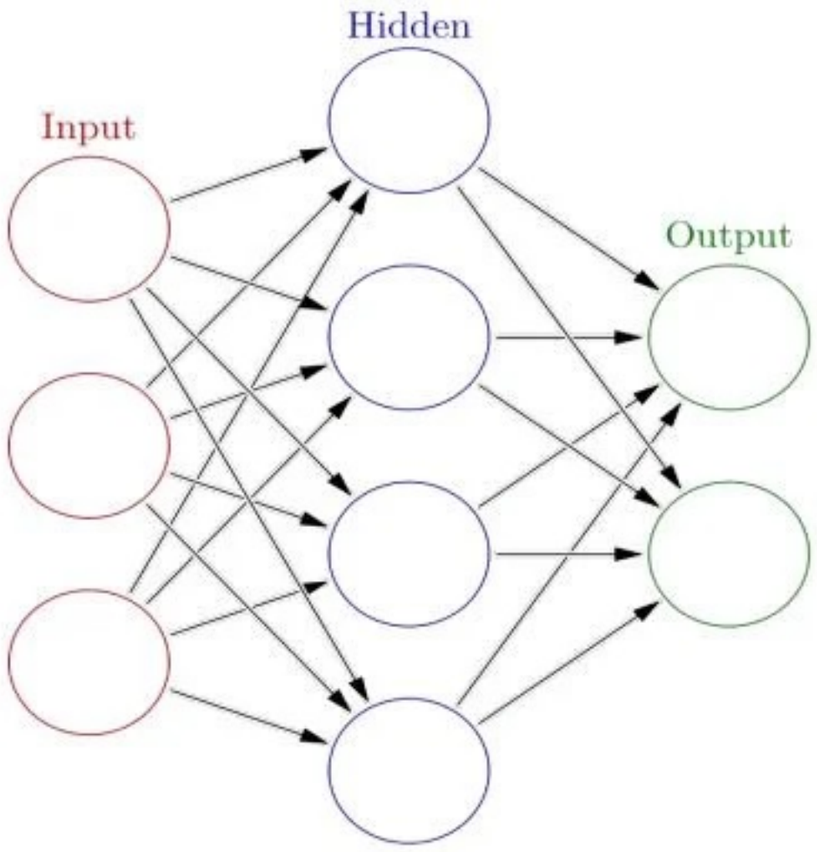


Figure 10: A visual representation of a neural network with an input layer, one hidden layer, and an output layer. Each node is referred to as a neuron, after the biological neuron.

Source: Wikimedia Commons

Why not just train the model on data from the whole world? For one, it is difficult to gather every last bit of the whole world's data on the healthcare industry or other industries. But the bigger problem lies in the fact that while training machine learning models on more "general" data reduces bias, it also creates a new problem: high variance. High variance occurs when a model uses too many parameters and is overly sensitive to small fluctuations in data, leading to drastic differences in predictions for the tiniest changes in data. For instance, multiple diseases are known to have similar symptoms, and many diseases have symptoms that do not show in every patient. A high variance model might make correct but inconsistent predictions (Ertel & Black, 2018).

A more interesting challenge arises when algorithms are trained on biased data. From elite college admissions to jobs, prison convictions, and even the most basic societal expectations, it is not difficult to find bias in society. These biases are captured when data is collected and carry on into machine learning models trained on the data. For instance, Facebook's advertising algorithm showed a prep-school teacher advert to an audience of 94% women and instead showed an advert for a truck driver to an audience of 87% men (ProPublica, 2020). In law, programs designed to aid judges in sentencing convicts have been proven to racially profile and discriminate against certain races which are historically discriminated against by the American justice system (Char et al., 2018). This issue calls for caution in the implementation of machine learning models and awareness that data might be biased. This also leads to ethical concerns over the appropriateness of machine learning in some fields, especially when the decisions being made have an impact on the lives of other people.

Others call into question the long-term effects of the extensive integration of machine learning into society. Elon Musk, the billionaire founder of Tesla, an industry-leading company in self-driving technology which uses machine learning, once likened the adoption of machine learning and artificial intelligence into society to "summoning the demon" and termed it the "biggest existential threat" (Coglianese & Lehr, 2017). This and other negative assumptions detract from the advantages of machine learning in society. For instance, automation of

"In law, programs designed to aid judges in sentencing convicts have been proven to racially profile and discriminate against certain races which are historically discriminated against by the American justice system."

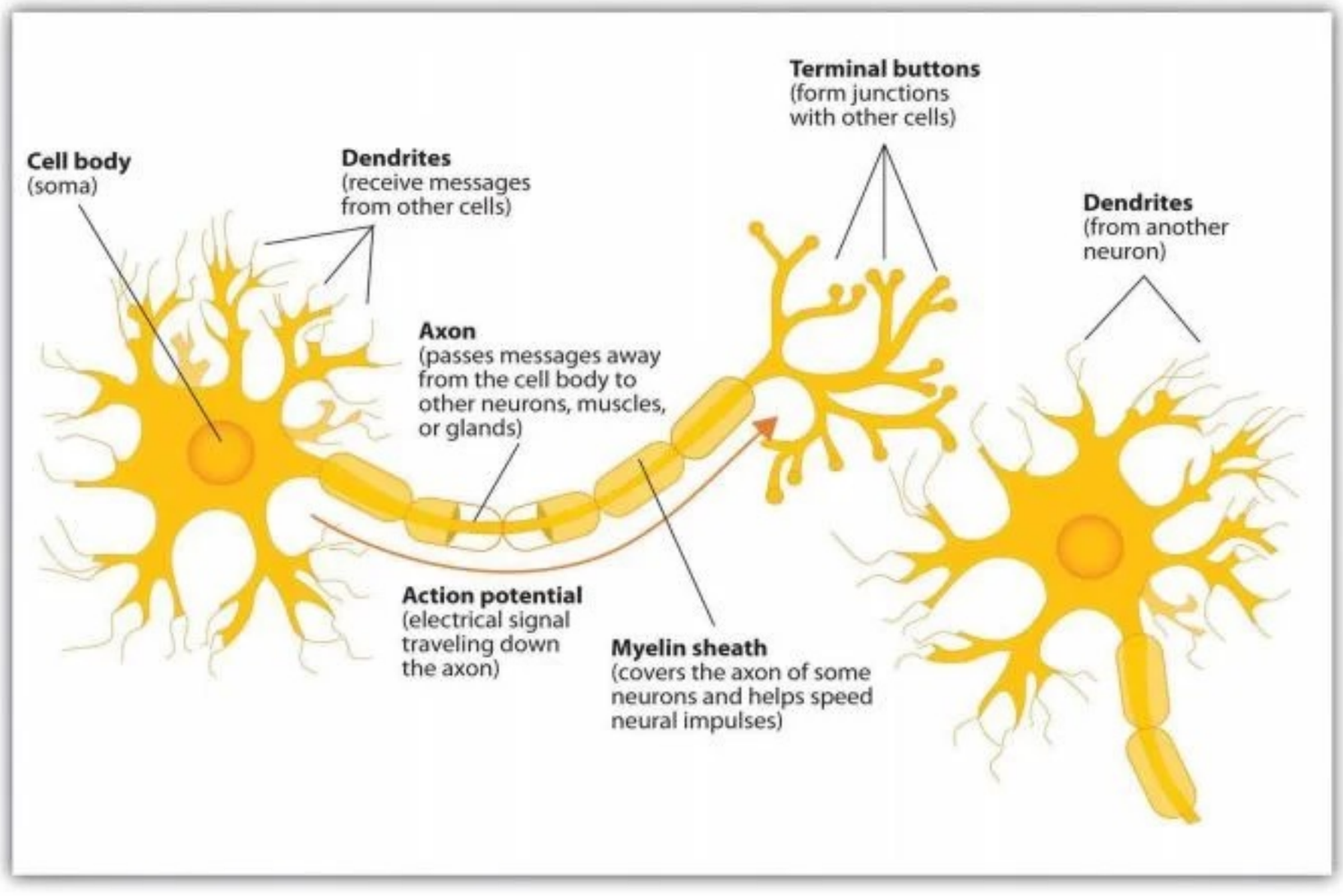


Figure 11: A neural network is similar to a network of biological neurons (pictured here): at each level, a computation is executed either turning on the activation function and transmitting a signal (analogous to a computational "1"), or not turning on the activation function (a computational "0").

Source: Wikipedia, Jennifer Walinga

manual labor in factories could potentially render jobs obsolete en masse (Paus, 2018). Even when it creates new positions, the previous workers will not fit in the created positions, which begs a bigger question: in an employment ecosystem dominated by machine learning, what place do those lacking technical skills have? This must be addressed before the widespread adoption of machine learning and artificial intelligence.

Building a Neural Network in Julia

"A neural network, an important computational structure in machine learning and the foundation of deep learning, is a set of algorithms that seeks patterns in a dataset by mimicking the functionality of biological neural systems."

A neural network, an important computational structure in machine learning and the foundation of deep learning, is a set of algorithms that seeks patterns in a dataset by mimicking the functionality of biological neural systems (Judd, 1990).

Julia was used because of its speed over other numerical computing languages like Python (Julia is compiled while Python is interpreted), its native support for important operations like the dot operator (which has to be done with a library in Python), and support for metaprogramming—the ability of a program to modify itself and other programs as it runs, as opposed to only modifying data (Bezanson, 2012).

Section 1.1: Relevant functions used in a neural network

1. Linear regression: This is of the form y = m*x + b . To make it more intuitive, the equation is usually adjusted, changing "m" to "W" to represent weights.

$$z = W * a + b$$

2. The sigmoid function: This is relevant where it is necessary to map an arbitrary numerical input to an output between 0 and 1.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

3. The sigmoid derivative: during back-propagation, the gradient of the function at given points for gradient-descent optimization must be found. Since the sigmoid function will have been computed before needing its derivative, it is most efficient to derive a differential equation (an equation for the derivative involving the original function itself) instead of explicitly computing the derivative.

$$\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left[\frac{1}{1 + e^{-x}}\right] \\
&- \frac{d}{dx}\left(1 + e^{-x}\right)^{-1} \\
&= -(1 + e^{-x})^{-2}(-e^{-x}) \\
&- \frac{e^{-x}}{(1 + e^{-x})^2} \\
&- \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} \\
&- \frac{1}{1 + e^{-x}} \cdot \frac{(1 + e^{-x}) - 1}{1 + e^{-x}} \\
&- \frac{1}{1 + e^{-x}} \cdot \left(\frac{1 + e^{-x}}{1 + e^{-x}} - \frac{1}{1 + e^{-x}}\right) \\
&- \frac{1}{1 + e^{-x}} \cdot \left(1 - \frac{1}{1 + e^{-x}}\right) \\
&- \sigma(x) \cdot (1 - \sigma(x))
\end{aligned}$$

4. The tanh function: This is a hyperbolic function that is particularly useful when we need to map arbitrary numerical inputs to the range -1 to 1.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

5. The tanh derivative: the tanh derivative is also needed for gradient descent. It is computed as follows.

$$\frac{d}{dx}sinh(x) = \frac{d}{dx}(\frac{e^x - e^{-x}}{2}) = \frac{e^x + e^{-x}}{2} = cosh(x)$$

$$\frac{d}{dx}cosh(x) = \frac{d}{dx}(\frac{e^x + e^{-x}}{2}) = \frac{e^x - e^{-x}}{2} = sinh(x)$$

$$\frac{d(f(x)/g(x))}{dx} = \frac{f'g - fg'}{g^2}$$

$$\frac{d}{dx}tanh(x) = \frac{d}{dx}(\frac{sinh(x)}{cosh(x)}) = \frac{sinh'(x).cosh(x) - sinh(x)cosh'(x)}{cosh^2(x)}$$

$$\frac{d}{dx}tanh(x) = \frac{cosh^2(x) - sinh^2(x)}{cosh^2(x)} = 1 - \frac{sinh^2(x)}{cosh^2(x)}$$

$$\frac{d}{dx}tanh(x) = 1 - tanh^2(x)$$

Section 1.2: Software implementation of relevant functions

View project code in the appendix at the end of this article. There is a code block for each paragraph from section 1.2 to section 6.

Section 2.1: Neural network abstraction

In computer science, data abstraction is the process of masking the details and data in a program by creating an outer shell as a single representation of the collective data and logic. Rather than passing multiple lists across different functions, it is easier to create an outer shell to represent the neural network, then store all the relevant data inside the neural network abstraction. This outer shell (called a "class" in some languages such as Python and Java, or a "struct" in Julia and C) shall then be passed as a single object into

Section 2.2: Initializing the Neural Network

An abstraction is not active until an instance of is created. To do this, a function is created to handle the instantiation request with appropriate instance variables or parameters and return a Neural network tuned to the requirements.

Section 3: Forward propagation

In a feed-forward neural network, forward propagation refers to processing inputs "forward" through the network of neurons to generate predictions.

Section 4: Back-propagation

Backward propagation goes in the reverse direction to forward propagation. While training a neural network, backward propagation is used to modify the parameters through a process known as gradient descent. Backward propagation is done after each iteration of forward propagation (Sejnowski, 2018).

Section 5: Training routine

The structure of the neural network is complete, but a routine for training the neural network must be created. This is completed by forward-propagating then back-propagating the neural network for a predetermined number of iterations (Judd, 1990). The weights in the neural network are updated in each iteration, and tend to stabilize after a number of iterations. A thousand iterations is a good compromise between ensuring the weights are stabilized and limiting redundant iterations. enough to stabilize the neural network weights and not computationally

Section 6: Prediction routine

After building and training this neural network, the next step is to analyze predictions on an unseen dataset. This is similar to the training routine, but backward propagation and iteration is not needed since parameters have already been optimized.

Section 7: Predictions

A neural network can be used for most pattern-recognition problems, including decision boundary problems, linear, quadratic, and exponential regression, even image recognition and natural language processing although those would require more sophisticated types of neural networks.

Here are some predictions on data:

1. Exponential data sequence: Where needed, a neural network can be used for sequential linearization of higher-order functions or transcendental functions. This is useful where getting faster, relatively accurate estimates is better than getting exact values at a higher computational cost (Sejnowski, 2018).

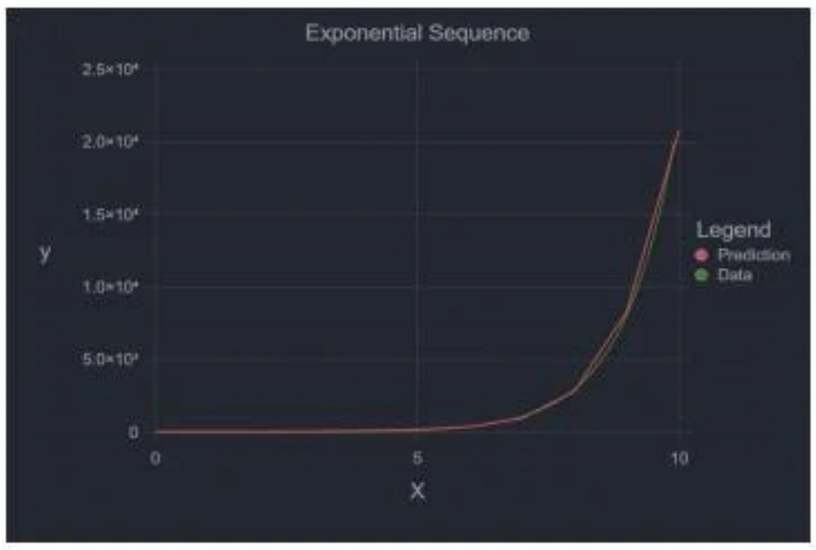


Figure 12: Sequential linearization of an exponential data sequence.

(Source: Image generated by the author)

2. Decision boundary problems: The neural network determines a boundary to separate data into two distinct groups. Neural networks try to learn the decision boundary which minimizes empirical error in the division of data. The type and accuracy of generated decision boundaries is dependent on the number of hidden layers in a neural network. A neural network without hidden layers can only predict linear decision boundaries (Sejnowski, 2018).

"In computer science, data abstraction is the process of masking the details and data in a program by creating an outer shell as a single representation of the collective data and logic."

Figure 13: A decision boundary generated by a neural network with 4 hidden layers.

(Source: Image generated by the author)

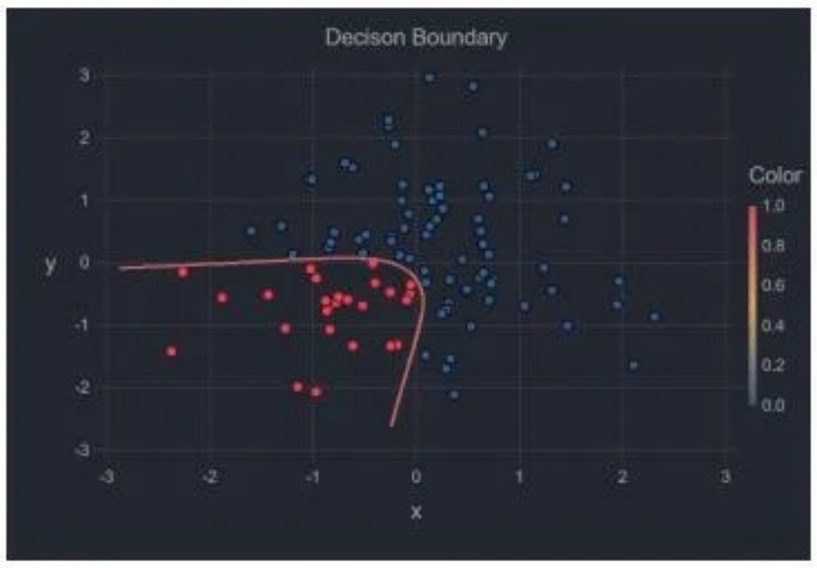
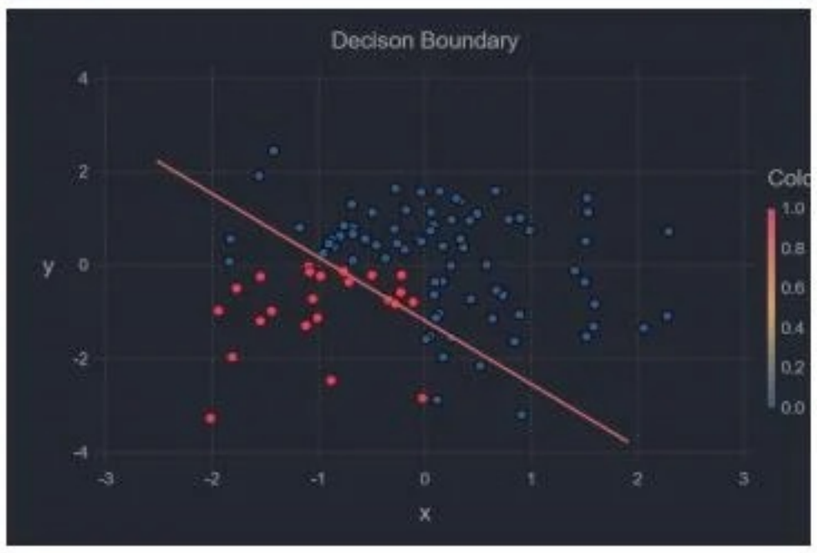
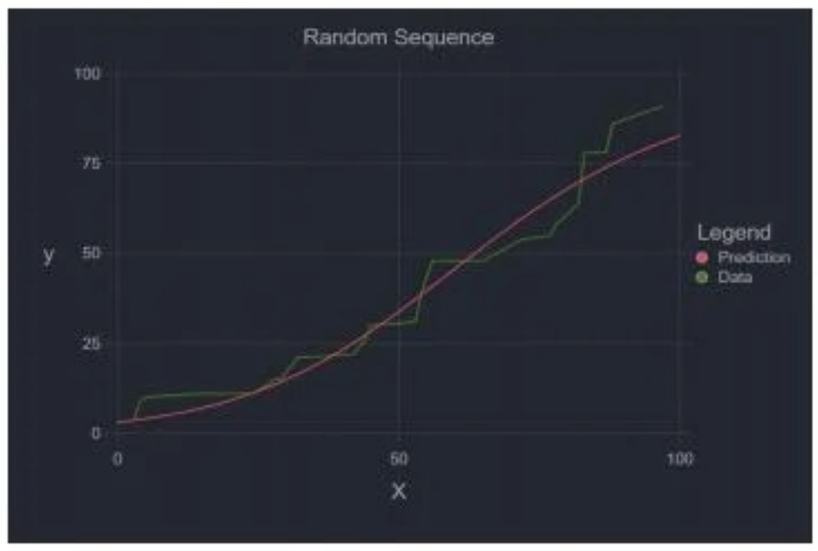


Figure 14: A linear decision boundary generated by a neural network with zero hidden layers. The presence of hidden layers is critical to a neural network's ability to develop a sophisticated understanding of data and make useful decisions. (Source: Image generated by the author)



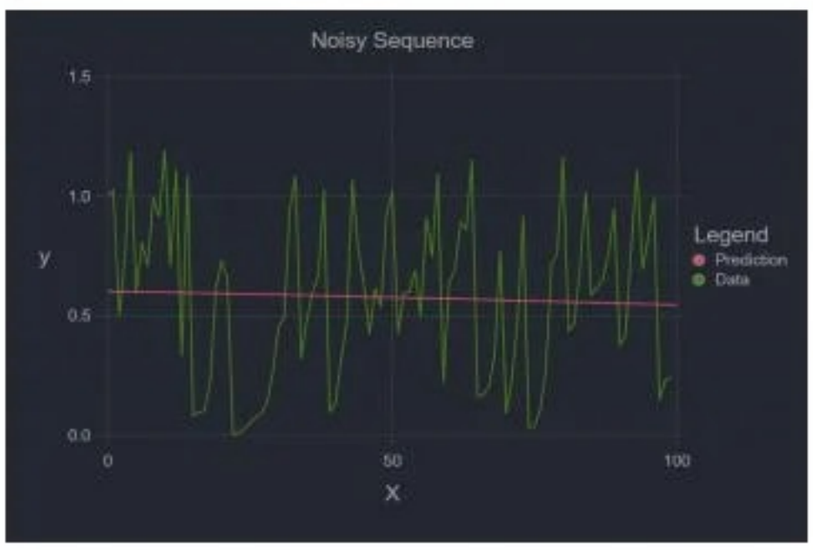
3. Higher order regression: Using a neural network generates a higher degree model that better fits the data. While this is good, it might lead to overfitting when a neural network is used on an overly simple dataset.

Figure 15: A higher order regression model (red line) generated to fit the trend in the data (green line). (Source: Image generated by the author)



4. Noisy data: Neural networks are not useful on all kinds of data. It is important to process data before feeding it into a neural network (or any other prediction model) and try to bring out the important features or suppress the less important where necessary. Unprocessed data is often noisy, which results in nonsensical predictions.

Figure 16: An attempted prediction on a noisy dataset. While neural networks optimize to minimize the error in predictions, there is no guarantee of accurate predictions if the data is noisy. (Source: Image generated by the author)



Conclusion

There is a lot to be expected from machine learning and AI in the future. Firstly, integration of AI into other industries is set to increase as the computational methods in AI become more powerful. While machine learning may eventually lead to the loss of certain work positions, it also creates more efficient workflows and allows professionals to focus their efforts elsewhere by removing the need to manually track and maintain processes (Pai, 2020).

Another trend is the rise of deep learning. Termed "the deep learning revolution," its rise is driven by the ability of deep neural networks to detect even the slightest trends from vast datasets (Sejnowski, 2018). Deep learning is itself a more sophisticated application of neural networks in data analysis. For instance, a deep neural network can have hundreds or thousands of hidden layers (the neural networks used in this article only had four hidden layers). Besides the feed-forward neural networks in machine learning, deep learning employs recurrent neural networks (RNNs) and convolutional neural networks (CNNs). RNNs involve iteration at specific layers in the neural network, which helps capture dependencies in the data and is ideal for predictions on data such as context-specific grammatical sentences, while CNNs can identify the features in data in the process of learning and predicting, making them ideal for problems such as image processing where it is harder for a human to identify and label specific data points in the numeric representation of an image (Pai, 2020).

With the rise of deep learning, GPU (Graphical Processing Unit) and TPU (Tensor Processing Unit) computing is also gaining attention because of its use in large-scale deep learning systems. GPUs are designed for processing and rendering graphics, which usually involves short bursts of intense workloads. As a result, GPUs are fundamentally optimized for parallel computing unlike ordinary CPUs which are optimized for sequential computing. This feature positions GPUs to better handle hundreds or thousands of concurrent, short-time computation tasks from multiple neurons, speeding up the performance of neural networks by up to 20 times. In contrast, an ordinary CPU would have to queue each task into an instruction pipeline and execute each individually or only a few at a time and await execution. TPUs, pioneered by Google, are specifically optimized for AI acceleration and

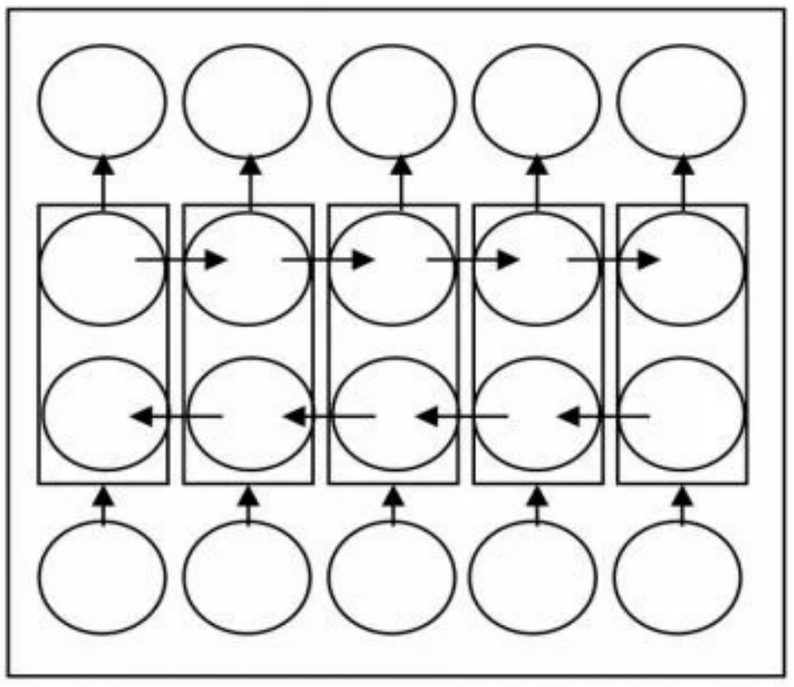


Figure 17: A bi-directional recurrent neural network (BI-RNN). BI-RNNs are particularly important in natural language processing, where words before and after each word provide important context for the usage of the word being classified. In a BI-RNN, layers computing in opposite directions are stacked together such that one generates predictions in the natural ordering of words in a sentence and the other computes predictions in the reverse ordering of words. Both predictions are then considered for the final interpretation of each word. Thus, the model can look as far ahead and as far behind as needed to make the best predictions (Goldberg & Hirst, 2017). (Source: Wikimedia Commons)

share much of the advantages of GPUs (Huang, 2018).

As new technologies such as GPUs, TPUs, and improved CPUs optimize the computational capability of modern computers for AI, specialists are able design and implement more powerful machine learning and deep learning models to solve challenging problems. This is, in part, a contributor to the rising adoption of machine learning and AI in other industries. More powerful applications of AI are emerging: NASA is using AI to hunt for exoplanets (Italiana, 2018), CERN is using AI to extract useful information from terabytes of data collected from its particle accelerator (Nature Editorial, 2018), meteorologists use AI to predict tropical cyclones with up to 99% accuracy (Italiana, 2018), and Eve, a robot "scientist" designed at Cambridge University and The University of Manchester, proved instrumental in the discovery that triclosan, a toothpaste ingredient, can be used to fight drug-resistant malaria (Italiana, 2018). Future advancements in processing and in AI will lead to more powerful applications and wider adoption of AI. can be expected to lead to increased adoption of AI subfields in other industries. In fact, a 2017 PwC study estimates that AI could replace 38% of all current jobs in the United States in the next 15 years (Barrows, 2017). However, a second PwC study done in the UK in 2018 determined that AI creates about as many technical jobs as the manual jobs it displaces (PricewaterhouseCoopers, 2018). For industry workers, there is but one tough decision: keep up or be rendered obsolete.

References

Alenezi, H. S., & Faisal, M. H. (2020). Utilizing crowdsourcing and machine learning in education: Literature review. Education and Information Technologies, 25(4), 2971–2986. https://doi.org/10.1007/s10639-020-10102-w

Barrows, J. (2017, December 21). Artificial Intelligence Will Change The Job Landscape Forever. Here's How To Prepare. Forbes. https://www.forbes.com/sites/quora/2017/12/18/artificial-intelligence-will-change-the-job-landscape-forever-heres-how-to-prepare/?sh=a13b81e27f40

Beam, A. L., & Kohane, I. S. (2018). Big Data and Machine Learning in Health Care. JAMA, 319(13), 1317. https://doi.org/10.1001/jama.2017.18391

Bezanson, J. S. K. (2012, February 14). Why We Created Julia. Julia Lang. https://julialang.org/blog/2012/02/why-we-created-julia/

Char, D. S., Shah, N. H., & Magnus, D. (2018). Implementing Machine Learning in Health Care — Addressing Ethical Challenges. New England Journal of Medicine, 378(11), 981–983. https://doi.org/10.1056/nejmp1714229
Coglianese, C., & Lehr, D. (2017). Regulating by robot: administrative decision making in the machine-learning era. The Georgetown Law Journal, 105(5), 1147–. (Coglianese & Lehr, 2017, p. 4)

Ertel, W., & Black, N. T. (2018). Introduction to Artificial Intelligence (Undergraduate Topics in Computer Science) (2nd ed. 2017 ed.). Springer.

Goldberg, Y., & Hirst, G. (2017). Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies). Morgan & Claypool Publishers.
Grossfeld, B. (2020, October 12). Deep learning vs machine learning: a simple way to understand the difference. Zendesk. https://www.zendesk.com/blog/machine-learning-and-deep-learning/#:%7E:text=Machine%20learning%20uses%20algorithms%20to,on%20what%20it%20has%20learned&text=Deep%20learning%20is%20a%20subfield,most%20human%2Dlike%20artificial%20intelligence

Haydin, V. (2021, March 5). How Machine Learning Algorithms Make Self-Driving Cars a Reality. Intellias. https://www.intellias.com/how-machine-learning-algorithms-make-self-driving-cars-a-reality/

Huang, J. (2018, December 21). Accelerating AI with GPUs: A New Computing Model | NVIDIA Blog. The Official NVIDIA Blog. https://blogs.nvidia.com/blog/2016/01/12/accelerating-ai-artificial-intelligence-gpus/
Italiana, U. D. S. (2018, October 15). AI: A limitless source of potential. Study International. https://www.studyinternational.com/news/ai-a-limitless-source-of-potential/

Judd, S. J. (1990). Neural Network Design and the Complexity of Learning (Neural Network Modeling and Connectionism) (Illustrated ed., Vol. 1). MIT Press.

Kasparov, G. (1998). My 1997 Experience with DEEP BLUE. ICGA Journal, 21(1), 45–51. https://doi.org/10.3233/icg-1998-21107

Koch, C., & Koch, C. (2016, March 19). How the Computer Beat the Go Master. Scientific American. https://www.scientificamerican.com/article/how-the-computer-beat-the-go-master/#:%7E:text=With%20its%20breadth%20of%20250,or%2010360%20possible%20moves.

Lau, J. (2020, September 3). Google Maps 101: How AI helps predict traffic and determine routes. Google. https://blog.google/products/maps/google-maps-101-how-ai-helps-predict-traffic-and-determine-routes/

López-Cajún, C., & Ceccarelli, M. (2018). Explorations in the History of Machines and Mechanisms: Proceedings of the Fifth IFToMM Symposium on the History of Machines and Mechanisms (History of Mechanism and Machine Science (32)) (Softcover reprint of the original 1st ed. 2016 ed.). Springer.

McCarthy, J. (2006, October 30). The Dartmouth Workshop--as planned and as it happened. Stanford University. http://www-formal.stanford.edu/jmc/slides/dartmouth/dartmouth/node1.html

Metz, C. (2017, June 3). In Two Moves, AlphaGo and Lee Sedol Redefined the Future. Wired. https://www.wired.com/2016/03/two-moves-alphago-lee-sedol-redefined-future/
Nature Editorial. (2018, May 4). Particle physicists turn to AI to cope with CERN's collision deluge. Nature. https://www.nature.com/articles/d41586-018-05084-2?error=cookies_not_supported&code=a9d46c28-e422-40fd-bc43-a2ce6caab27e

Niño, A. (2009). Machine translation in foreign language learning: language learners' and tutors' perceptions of its advantages and disadvantages. ReCALL, 21(2), 241–258. https://doi.org/10.1017/s0958344009000172

Pai, A. (2020, October 19). CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/

Paus, E. (2018). Confronting Dystopia: The New Technological Revolution and the Future of Work (Illustrated ed.). ILR Press.
PricewaterhouseCoopers. (2018, July 17). AI will create as many jobs as it displaces by boosting economic growth. PwC. https://www.pwc.co.uk/press-room/press-releases/AI-will-create-as-many-jobs-as-it-displaces-by-boosting-economic-growth.html

ProPublica. (2020, March 2). Facebook Ads Can Still Discriminate Against Women and Older Workers, Despite a Civil Rights Settlement. https://www.propublica.org/article/facebook-ads-can-still-discriminate-against-women-and-older-workers-despite-a-civil-rights-settlement

Rada, R. (1986). Artificial intelligence. Artificial Intelligence, 28(1), 119–121. https://doi.org/10.1016/0004-3702(86)90034-2

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., & Silver, D. (2020, December 23). Mastering Atari, Go, chess and shogi by planning with a learned model. Nature. https://www.nature.com/articles/s41586-020-03051-4?error=cookies_not_supported&code=6a225d41-20f3-485d-8628-56e5761fbf32

Sejnowski, T. J. (2018). The Deep Learning Revolution (1st ed.). The MIT Press.

Shannon, C. E., & McCarthy, J. (1956). Automata Studies. (AM-34), Volume 34 (Annals of Mathematics Studies). Princeton University Press.

Sharma, A., Jain, A., Gupta, P., & Chowdary, V. (2021). Machine Learning Applications for Precision Agriculture: A Comprehensive Review. IEEE Access, 9, 4843–4873. https://doi.org/10.1109/access.2020.3048415

Tesauro, G. (1995, March 1). Temporal Difference Learning and TD-Gammon. BackGammon. https://bkgm.com/articles/tesauro/tdl.html

Appendix: Project Code

Section 1.2: Software implementation of relevant functions

```
18
19  # TODO: Important functions we'll need in out Neural Network
20
21  #1 sigmoid function
22  function sigmoid(x)
23      sig_x = 1 ./ (1 .+ exp.(-x))
24  end
25
26  # sigmoid derivative
27  function sigmoid_derivative(sig_x)
28      sigmoid_derivative = sig_x .* (1 .- sig_x)
29  end
30
31  # tanh function
32  function tanh(x)
33      tanh_x = (exp.(x) - exp.(-x)) / (exp.(x) + exp.(-x))
34  end
35
36  # tanh derivative
37  function tanh_derivative(tanh_x)
38      tanh_derivative = 1 - tanh_x.^2
39  end
40
```

Section 2.1: Neural network abstraction

```
7
8   """
9   Neural network abstraction
10  """
11  mutable struct Network
12      a::Array{Any,1}     # Inputs: 1-dimensional array of elements of any (numerical) type
13      W::Array{Any,1}     # Weights: 1-dimensional array of elements of any type
14      b::Array{Any,1}     # Error correction: 1-dimensional array of elements of any type
15      epsilon::Float64    # Step size; hyperparameter for gradient descent
16      result::Array       # prediction results
17  end
18
```

Section 2.2: Initializing the Neural Network

```
100
101 # backward propagation --> Process of modifying model parameters
102 """
103 ---
104     Backward propagation routine --> modifying parameters
105     backward!(network::Network, X, y)
106
107     # Argunemts
108     network::Network : Neural Network to be propagated backward
109     X : input data values
110     y : accurate input labels on the data
111 ---
112 """
113 function backward!(network::Network, X, y)
114     # Compute error in network's predictions
115     error = y - network.result
116     # Find step amount by multilying error with sigmoid derivative of results
117     delta = error .* sigmoid_derivative(network.result)
118
119     # Compute new weights and insert into Array of weights
120     network.W[end] += network.epsilon * transpose(network.a[end-1]) * delta
121
122     # Compute new error corrections (average of step amounts)
123     network.b[end] .+= network.epsilon * mean(delta)
124
125     # Get number of nodes in network and back-propagate,
126     # excluding final layer which has already been handled above
127     W_length = length(network.W)
128     for i = 1:W_length-1
129         j = W_length - i
130
131         error = delta * transpose(network.W[j+1])
132         delta = error .* sigmoid_derivative(network.a[j+1])
133         network.W[j] += network.epsilon * transpose(network.a[j]) * delta
134         network.b[j] .+= network.epsilon * mean(delta)
135     end
136
137     return network
138 end
139
```

Section 3: Forward propagation

```
"""
...
    Forward-propagation routine --> generating predictions
    forward!(network::Network, X)

    # Arguments
    network : Neural network
    X : input values
...
"""
function forward!(network::Network, X)
    network.a = [X]     # Plug inputs into Network

    # TODO Step 1: For each weight,multiply with corresponding input,
    # and add corresponding error correction
    for i = 1:length(network.W)
        # z = W*a + b --> modelled after y = mx + b
        z = network.a[i] * network.W[i] .+ network.b[i]

        # TODO: Step 2: Compute sigmoid of value computed by linear regression
        # and re-insert into Array of inputs for next iteration
        push!(network.a, sigmoid(z))
    end
    # Results are last N values computed by final layer
    network.result = network.a[end]

    return network
end
```

Section 4: Back-propagation

```
# backward propagation --> Process of modifying model parameters
"""
...
    Backward propagation routine --> modifying parameters
    backward!(network::Network, X, y)

    # Argunemts
    network::Network : Neural Network to be propagated backward
    X : input data values
    y : accurate input labels on the data
...
"""
function backward!(network::Network, X, y)
    # Compute error in network's predictions
    error = y - network.result
    # Find step amount by multilying error with sigmoid derivative of results
    delta = error .* sigmoid_derivative(network.result)

    # Compute new weights and insert into Array of weights
    network.W[end] += network.epsilon * transpose(network.a[end-1]) * delta

    # Compute new error corrections (average of step amounts)
    network.b[end] .+= network.epsilon * mean(delta)

    # Get number of nodes in network and back-propagate,
    # excluding final layer which has already been handled above
    W_length = length(network.W)
    for i = 1:W_length-1
        j = W_length - i

        error = delta * transpose(network.W[j+1])
        delta = error .* sigmoid_derivative(network.a[j+1])
        network.W[j] += network.epsilon * transpose(network.a[j]) * delta
        network.b[j] .+= network.epsilon * mean(delta)
    end

    return network
end
```

Section 5: Training Routine

```
"""
...
    Training routine
    train(network::Network), X, y, n = 1000

    # Arguments
    network : Neural network to be trained
    X : Array of input data values
    y : Array of pre-determined (and accurate!) data labels
...
"""
function train!(network::Network, X, y, n=1000)
    # For each data iteration,
    for i = 1:n
        # Step 1: forward-propagate to get predictions
        forward!(network, X)

        # Step 2: backpropagate to update model
        backward!(network, X, y)
    end
end
```

Section 6: Prediction routine

```
"""
...
    Prediction routine
    predict!(network::Network, XP)

    # Arguments
    network : Neural network to be used to generate predictions
    XP : data to be analyzed for patterns
...
"""
function predict!(network::Network, XP)
    # Forward propagate through network to get predictions
    forward!(network, XP)
end
```