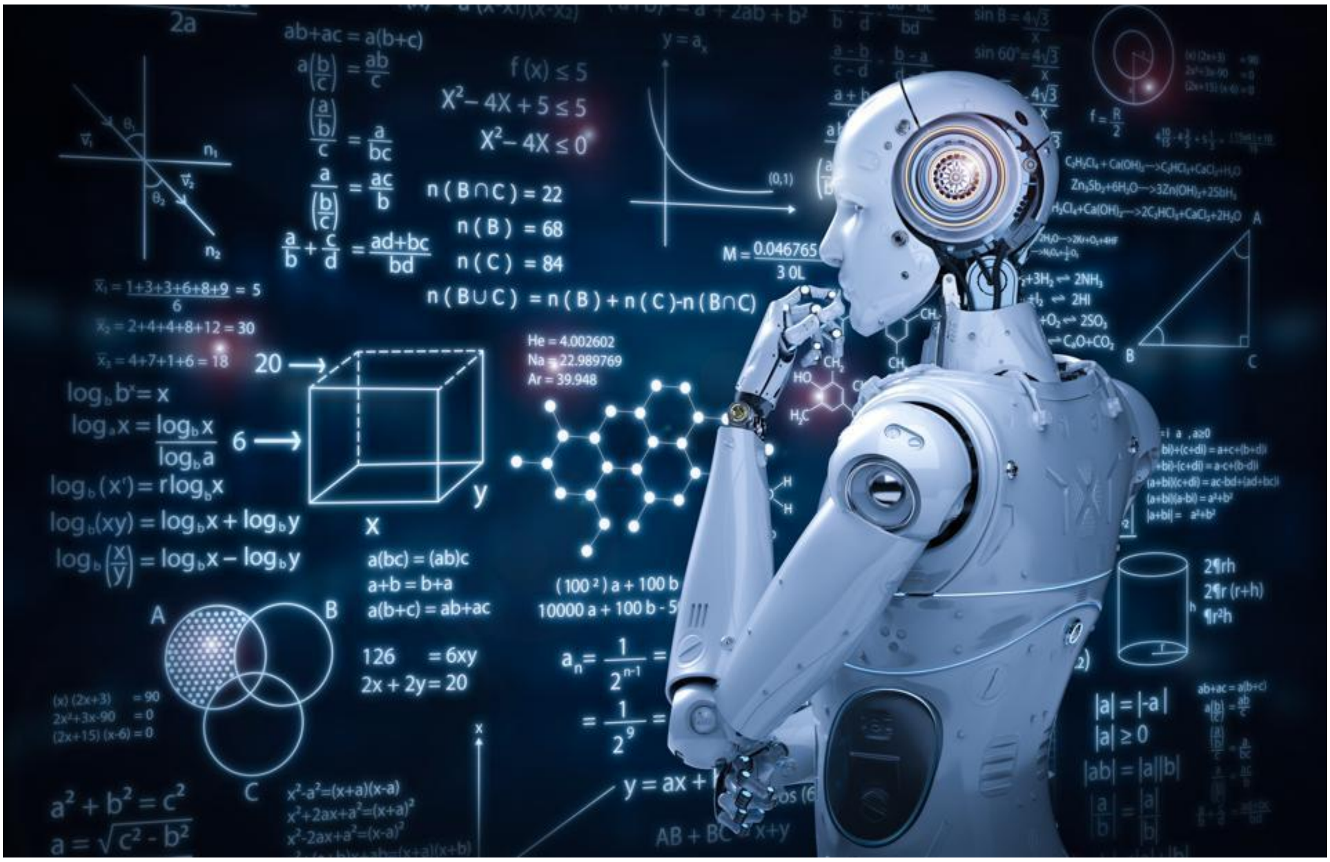
Contents

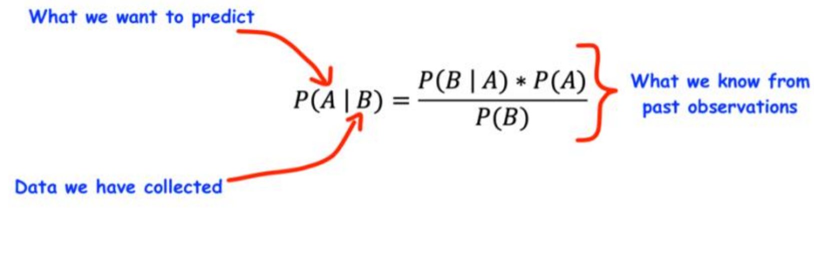

ab+ac = a(b+c)
f (x) ≤ 5
X²– 4X + 5 ≤ 5
X²– 4X ≤ 0
n (B ∩ C) = 22
n (B) = 68
n (C) = 84
n (B ∪ C) = n (B) + n (C)-n (B∩C)
log_b b^x = x
log_a x = log_b x / log_b a
log_b(xy) = log_b x + log_b y
a(bc) = (ab)c
a+b = b+a
a(b+c) = ab+ac
126 = 6xy
2x + 2y = 20
a² + b² = c²
y = ax +
|a| = |-a|
|a| ≥ 0
|ab| = |a||b|

Welchman Prize – Machine learning and AlphaZero

The technology of the present and the future: Machine Learning. The amount of power we have, the complexity, and depth of what we can ascertain using machine learning (ml) is what enthuses me. Ml is a concept under data science that involves optimization and statistical learning methods to predict patterns through its intricately crafted brain and learns from its mistakes and corrects them just like us, humans.

A typical machine learning algorithm consists of roughly three components: A decision process, an error function, and an updating or optimization process. The first component takes in the data and through its calculations via the algorithm it produces a guess. Error function, essentially, is how inaccurate the guess was. The machine compares its guess by looking at historic trends and decides how off its guess was if it was. The final function calculates how the guess can be improved given the historic data and makes a closer prediction. This is then repeated.

The most captivating part of ml to me is that once it has been programmed it doesn't require being regulated by the coder again and it gets more accurate over time through mistakes. There are four main types of machine learning: Supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.



Bayes' theorem

Figure 2.0

Supervised learning is when the dataset being used by the data scientist is labelled and paired with the desired output. For example, in the context of spam filtering a data scientist would provide the algorithm with a set of examples of spam messages and allow it to work out patterns. One of the most popular ml algorithms used for this type of machine learning is Bayes' theorem. The formula would look like in figure 2.1.

$$P(Spam \mid [term\ 1, term\ 2, \ldots]) = \frac{P([term\ 1, term\ 2, \ldots] \mid Spam) * P(Spam)}{P([term\ 1, term\ 2, \ldots])}$$

Figure 2.1

Having given the machine examples of spams, the data would need to be tokenized. This means that the data is split into smaller parts. For example, "Mathematic is very useful" would become "Mathematics, is, very, useful". Furthermore, each word has its weight this depends on how frequently it appears in spam e-mails. For instance, if ‟win big money prize" is one of your features and only appears in spam emails then it will be given a larger probability of being spam. If “important meeting” is only mentioned in ham emails (non-spam) then the probability of it being a ham email will increase. Each term in the message body is multiplied by its weight and the sum of the weight to determine the probability that the email is spam. For the simplicity of it if we take one term - “Win” - the following would happen: the probability of the word “Win” appears in an email is 25%, the probability of an email being spam is 35%, and the probability of a junk email has the word “Win” is 45%. Then, when an email that contains the word “Win” is received by a user, the algorithm will calculate the probability of this email is a spam according to the Bayes’ theorem is 63%. This probability isn’t too high however, a regular algorithm only filters when the probability of an email being junk is 99%. In more complex algorithms spam filtering is more complicated than just Bayes’ theorem.

Unsupervised learning does not require a data scientist to provide labels to inputs and to pair desired outcomes. Instead, the algorithm sifts through unlabelled input data and identifies patterns to group data. This kind of machine learning can be very useful in cybersecurity where the attacker constantly changes methods. The way in which unsupervised learning works is that it clusters data and analyses it. Clustering is a data mining technique which groups unlabelled data based in their similarities and differences. If we take figure 2.2 as an example, each colour has its own co-ordinate and the shades that are similar tend to be nearer each other and have similar features, hence why it is called “cluster”.

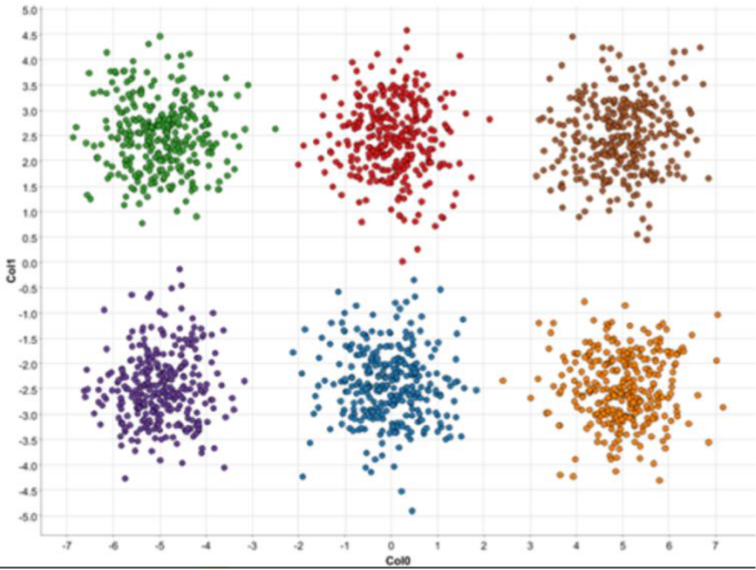


Figure 2.2

Semi-supervised machine learning is a mixture of supervised and unsupervised learning. Essentially, a small portion of labelled data is combined with a large portion of unlabelled data. The beauty of this type of ml is that the machine can be trained to label unlabelled data without using much labelled data. Suppose one would like to classify any given webpage into one of several categories like “education”, “health”, “entertainment” etc. There is a major issue with this though: it will be expensive to go through tens of thousands of webpages and labelling them. Therefore, only a small portion of them would be labelled. However, it can be said that webpages are abundant meaning they can be accessed easily. An algorithm in Python for example could collect millions of pages in a couple of hours which would make up the unlabelled sector of the dataset. The algorithm is perfected using semi-supervised learning.

Reinforcement learning is when a dataset uses “rewards/punishment” system, offering feedback to the agent and learns from its own experience through trial and error. An agent is an entity that can perceive/explore the environment and act upon it. [Agent performs an action → change state/remain in the same state → sees the results of its action] This runs in a loop. “Rewards” means positive feedback i.e. the agent has made the correct decision. “Penalty” also known as “punishment” is when the agent receives negative feedback i.e. it has made a mistake. Reinforcement learning (RL) is a core part of artificial intelligence, and all AI agents work on the concept of RL. For example, autonomous cars. In driving there are various factors to consider such as speed limits, drivable zones, car collisions – just to mention a few. RL is used for some tasks such as trajectory optimisation, motion planning, dynamic pathing to deal with these factors mentioned above.

In the forthcoming paragraphs I will explore a specific machine that I find incredibly intriguing. The sensation I feel towards this algorithm was sparked by a book I read. This machine is no regular machine. It is the one that beat the world’s best Chess algorithm of the time also known as Stockfish 8. This supernatural algorithm that left Stockfish 8 powerless is called AlphaZero. AlphaZero was created by Google on December 5th, 2017. Having only played Chess for four hours and was limited by the number of calculations it could do (per minute) – 80,000 to 70 million - AlphaZero overthrew Stockfish 28 times and drew 72 times out of 100 matches. It is therefore striking that the efficiency of AlphaZero proved to be more significant in overall levels of performance, as opposed to rate of calculation per minute. In addition to these, AlphaZero was never taught any chess strategies – not even standard openings. Instead, it used the most recent ml principles which was to self-learn by playing against itself.

There were two main machine learning concepts behind it but, I will only focus on one: deep neural networks. Deep neural networks are the backbone of deep learning. Neural networks with three or more hidden layers count as Deep learning. Hidden layers are layers between the input layers and the output layers that use artificial neurons to take in a set of weighted inputs and produce an output through an activation function. An activation function is a function which is applied to an inactivated layer of neurones to make an algorithm’s operations non-linear which activates the neurone. The major activation functions are Rectified linear unit (ReLu), sigmoid and

Figure 2.3

tanh. AlphaZero uses several activation functions but, mostly it consists of the ReLu function as far as I know.

Traditional neural networks only contain 2 to 3 hidden layers while deep networks can have as many as 150. For example, the algorithm that keeps an autonomous car in the lane uses only three hidden layers. However, AlphaZero incredibly has 80 hidden layers and hundreds of thousands of neurons which makes it very complex. These neural networks are personified to help people understand what they are. So, they have artificial neurons (coded), not literally. Deep learning is when computers learn automatically from datasets without introducing rules, but this necessitates a lot of raw data otherwise it would be too difficult for the machine to predict and learn.

Deep learning uses a series of intertwined layers which swiftly learn complex models. A neural network with a single layer can still make approximate predictions but additional layers will help with optimization. So, if we take a machine with multiple layers, it will start by deducing a prediction using the first layer, then building upon the previous layer, it will refine and optimise the prediction. This process of working through the network is known as forward propagation. Another process is called backpropagation. This is when the machine uses algorithms such that errors in predictions are calculated and weights and biases are adjusted by moving backwards through the layers. Forward propagation and back propagation work in sync to allow neural networks to make predictions and refine accordingly. Weights control the signal (or the strength) of the connection between two neurons. In other words, a weight decides how much influence the input will have on the output. Biases, which are constant, are an additional input into the next layer that help fitting the prediction better with the data. Take $\boldsymbol{y = mx + c}$ as an example, The biases are like the **c** value here. If the constant c is absent then the line will pass through the origin (0, 0) and you will get a poorer fit.

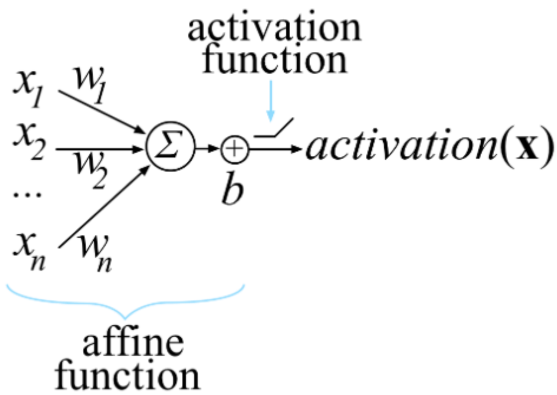


Figure 2.4

The activation of a single computation unit in a neural network (artificial neurone) is typically calculated using the dot product of a weight vector (*w*) with an input vector (*x*) plus a scalar bias (*b*). Then an activation function is used to activate that neuron. Hundreds of these neurons make up one hidden layer. The function *Z(X)* shows one inactivated neuron. *Z(X)* is called the unit's affine function. The activation of one layer's neurones become the input of the next layer's units. Training this neuron means

$$z(\mathbf{x}) = \sum_i^n w_i x_i + b = \mathbf{w} \cdot \mathbf{x} + b\,.$$

Figure 2.5

choosing weights (**w**) and bias (*b*) so that we get the desired output for all *N* inputs (**x**). To manage this we try to minimise the cost function. The cost function is how much every operation in the neural network contributes to the inaccuracy. As the cost function decreases inaccuracies decrease, therefore the predictions the algorithm makes improves.

$$\frac{1}{N}\sum_{\mathbf{x}}(target(\mathbf{x}) - activation(\mathbf{x}))^2 = \frac{1}{N}\sum_{\mathbf{x}}(target(\mathbf{x}) - max(0, \sum_{i}^{|x|} w_i x_i + b))^2$$

Figure 2.6

Figure 2.6 shows a cost (loss) function. *Target(X)* is the desired number of activated neurones. *activation(x)* is the number of neurones activated and the summation of the "net activation" of all neurones and the mean is worked out by dividing it all by the number of neurones. The *activation(x)* however can be greater than the *target(x)* which is why the brackets are squared: to make it positive because even if the activated number of neurones exceeds the target it is still an accuracy which is added onto the total inaccuracy. The right-hand side is the same thing but the part that is different uses the ReLu activation function. ReLu outputs the input directly if it is positive, otherwise, it will output zero: *max(0, Z(X)).* Since this is just one-unit, neural networks need to train the weights and biases of all neurons in all layers at the same time.

As you can see machine learning and AI has already started forming the present and there is no doubt that it will be paramount in the future. All four versions of machine learning are getting more common each day. AlphaZero is a machine learning miracle that enchanted the world, and it will be influential to many more companies and will incentivise companies to lean towards this topic. The world is developing at the speed of light and machine learning is right at the core of this revolution. Be a part of this change.

Bibliography:

- https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/
- 21 lessons of the 21^{st} century
- Explained.ai
- livebook.manning.com
- blogs.cornell.edu
- knime.com