

# The Thought Algorithm

## A mathematical problem

October 3, 2016 · Technology

Chanon Olley

Olley C. The Thought Algorithm: A mathematical problem. Young Scientists Journal. 2016 Oct 3 [last modified: 2016 Oct 3].

Edition 1.

## Abstract

Can a computer think? This article explores the extent to which Artificial Intelligence is possible and how looking at our own brains may pave the way to more human-like data processing.

## Introduction

Image not found

<https://archive.ysjournal.com/wp-content/uploads/2016/08/deepmind1-300x169.jpg>

### Fig. 1: Visit to Google DeepMind in London

This photograph was taken inside Google DeepMind's headquarters, where I was invited personally for a visit with research scientist, Irina Higgins. It's an amazing place to work at the forefront of Artificial Intelligence and Machine Learning.

Artificial Intelligence has puzzled computer scientists for many years and has recently flourished with the advancements in Machine Learning. Google DeepMind's AlphaGo is a program that has beaten Lee Sedol – grandmaster of one of the most complex decision-based games, Go – 4 games to 11, an accomplishment previously thought to be impossible. Andrew Ng, Chief Scientist at Baidu Research and associate professor in Computer Science and Electrical Engineering at Stanford University, founded the Google Brain project and is pioneering the field of Deep Learning<sup>2</sup> – an approach to Machine Learning that utilises Artificial Neural Networks (ANNs) in an attempt to process data in a similar way to humans. Despite the incredible research going on in the field of Artificial Intelligence, you would be inclined to wonder whether this is really intelligence, whether a machine really can learn.

## Machine Learning

Machine Learning<sup>3</sup> – arguably a branch of Artificial Intelligence, alongside Natural Language Processing (NLP) – is made up of two types: supervised learning and unsupervised learning. Supervised learning is a form of data processing that is used to predict an output variable, such as the price of a house, given some input variables – house size, for example – and some example houses (a training set) to train with. This scenario is dealt with by what is called “regression”<sup>4</sup> analysis which outputs continuous data. Another case of supervised learning is “classification” or “logistic regression” analysis which outputs discontinuous data (or data with a restricted number of possibilities). This could be identifying whether a cancer is malignant or benign. The distinction between supervised and unsupervised learning is that in supervised learning the output variables (malignant/benign or house prices in these cases) are known for the previous data that is used to train the algorithm; whereas in unsupervised learning the output variables are unknown and the algorithm is used to predict trends or patterns in the training set as opposed to predicting future outcomes. I will introduce you to some important aspects of Machine Learning, however, the section of this article relies on a basic knowledge of the topic. To learn more about Machine Learning and the algorithms that are involved, visit the course by Coursera and Stanford University<sup>3</sup>.

The first piece of terminology that is important to almost all areas of Machine Learning is the “hypothesis”<sup>5</sup>,  $h$ . This is the function of the input variables  $x$  used to compute the output variables  $y$ . This function is parameterised by the parameter variables  $\theta$  (notation is conventional – nothing to do with angles). To explain what  $\theta$  is used for and how  $h$  works, I will take you through the process of “linear regression”, the simplest form of regression that is ultimately finding the line of best fit from a dataset and making a prediction similar to the house prices example.

Image not found

<https://archive.ysjournal.com/wp-content/uploads/2016/08/figure2-linearreg-300x226.png>

Fig. 2: Linear Regression in Octave

Linear regression is a process that attempts to minimise a “cost function” or “loss function” – this could be interpreted as the cost to be paid if a prediction is made: thus a correct prediction would have smaller consequences and a lower cost than an incorrect prediction. In particular, linear regression works with a “squared-error cost function” – the square of the vertical distance between the line given by  $h$  and a training example  $(x, y)$ . In Figure 2, you can see that the function  $h$  is a linear function. In linear regression, you can imagine the coefficients of  $x$  to be stored in the vector  $\theta$  and  $c$  to be a real constant  $\theta_0$  (intercept term), where  $x_0$  is conventionally equal to 1. The whole function is shown below (where  $m$  is equal to the number of inputs of  $x$ ).

$$h(\theta) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_m x_m$$

figure3 examplelrb

Image not found  
<https://archive.archive.ysjournal.com/wp-content/uploads/2016/08/figure3-examplelrb-300x116.png>

Fig. 3: Example table for linear regression

The above function is also used for multiple input types of  $x$  (features). In the previous example, the feature was population, however, it can also be used with two features or more, such as living area and the number of bedrooms (shown in Figure 3). To simplify our equation, we can use a summation term or a matrix operation with the matrix – rather than vector –  $x$  (shown below).

$$h_{\theta}(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x$$

The hypothesis is used to determine the cost  $J$  of the chosen parameters  $\theta$ . In the function shown below, you can see that it simply adds all the squared differences up – between the hypothesis and the true output variables  $y$  – and halves it.

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

The squared-error cost function is in fact a quadratic function and therefore is bowl-shaped with a single global minimum. This makes it fairly easy to minimise the cost function.

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

To do so, there are several possible ways, including the normal equation (solving by differentiation) or other advanced optimisation methods – conjugate gradient or BFGS<sup>7</sup>) that could be imported as a library in practice. We will look instead at “gradient descent”, an algorithm that can easily be implemented for datasets varying largely in size.

figure4 sqerrcost

Image not found  
<https://archive.archive.ysjournal.com/wp-content/uploads/2016/08/figure4-sqerrcost-300x219.png>

Fig. 4: Squared-error cost function

How gradient descent works is given in the name: a node  $(\theta, J(\theta))$  starts at a point on the squared-error cost function and works its way down to the minimum. The values of  $\theta$  may be randomly initialised or initialised at zero for all features of  $x$ . Another variable used in gradient descent is the step  $\alpha$  – how much the node moves each iteration. The step is only needed for setting the speed of the algorithm and ensuring that the node does not overstep the minimum and go up the other side of the bowl; the gradient of the cost function will also be used to determine the change in the position of the node. The algorithm is shown below.

Image not found  
<https://archive.archive.ysjournal.com/wp-content/uploads/2016/08/GradDesc-300x60.png>



$\frac{\delta J(\theta)}{\delta \theta_j}$

is equal to zero), you will have the correct values of  $\theta$  to draw a line of best fit to your dataset. To predict an outcome with some inputs  $x$ , substitute the inputs and the trained parameters into the hypothesis function, where  $y=h_{\theta}(x)$ .

I have introduced you to one of the most basic forms of Machine Learning. As you continue through the topic, you will be able to use more complex algorithms – such as logistic regression, Support Vector Machines (SVMs), ANNs, K-means – in many different situations. Machine Learning is implemented in scenarios where a cost function is literal: the cost of telling someone they have a malignant cancer when they do not; the cost of convicting an innocent person; the cost of crashing an autonomous vehicle. In these cases, you may find that the algorithm you use does more bad than good. You can learn to easily modify these algorithms and optimise them for your situation.

Machine Learning is vital for achieving Artificial Intelligence. By combining algorithms, you can write a program that can learn very similarly to how humans learn. However, do humans really think in algorithms? Does the brain just mathematically find patterns in data? Is there such thing as a thought algorithm?

## Achieving Artificial Intelligence

One of the most useful ways to check the progress of our approach to Artificial Intelligence is the Turing Test. This is based on the Victorian Imitation Game<sup>8</sup> – a game that involves a man and a woman in a hidden place, communicating with a third person who attempts to guess which is the man and which is the woman. The Turing Test is basically the same, except the man is replaced with a computer; the third person is required to guess which is the woman. This is useful because it is thought that being intelligent involves being able to make spontaneous decisions (intuition) and converse more naturally and creatively (ingenuity) than one less intelligent. Therefore, if the third person struggles to guess correctly, we can conclude that we have achieved something, if not Artificial Intelligence.

figure5 turingtest

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2016/08/figure5-turingtest-300x115.png>

**Fig. 5: The Imitation Game and Turing’s Test**

Alan Turing has been quoted<sup>9</sup> to say, “We may hope that machines will eventually compete with men in all purely intellectual fields.” However, he has also claimed that “electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner.” He struggled to explain how machines could be able to think; he thought the term “machine” to be contradictory by definition to “thinking” and “intelligence”.

It is possible to argue that any task done by a human brain could be represented mathematically as a function thus, we will discuss the concept of a thought algorithm and what clues our brains give us for solving the mystery of Artificial Intelligence.

## The greatest machine on Earth

*We ourselves are a part of the mystery that we are trying to solve.*

— Max Planck

The human brain is complicated – it can move from one task to another seamlessly, invent number systems and explore the nature of the universe piece by piece, equation by equation, particle by particle. How does the brain process thoughts as words and pictures without needing to listen to or speak those words, or draw the pictures? How can humans play a piece on the piano without reading the music, purely by memory of how it feels, without further thought? These questions will not be answered in this article, however, I will explore the basic connections between Artificial Intelligence and Neuroscience.

## Neural Networks

It is thought that one of the best approaches to Artificial Intelligence is by taking the human brain as an example to follow – or even to compete with. As I mentioned in the previous chapter, some believe that the calculations made by the brain can be represented as mathematical functions which can be computed by a machine.

figure6 neurondiag

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2016/08/figure6-neurondiag-300x225.jpg>

**Fig. 6: Neuron diagram**

The human brain is made up of approximately 86 billion neurons (diagram shown in Figure 6<sup>10</sup>). In comparison to computers, the latest microprocessors sport about 1.5 billion transistors.

Neurons communicate with each other across synapses<sup>11</sup> – the conjunctions between neurons that allow the neurons to transmit signals. Synapses can be either chemical or electrical. Chemical synapses are connections between two neurons by the release of a neurotransmitter, resulting in the second cell (postsynaptic neuron) to be excited or inhibited by an electrical response or a secondary messenger. Electrical synapses, on the other hand, are connections from a presynaptic cell to a postsynaptic cell by gap junctions or synaptic cleft that are capable of passing an electrical current causing voltage changes in the neurons. Electrical synapses are advantageous for rapid transfers of signals, however, chemical synapses can have more complex effects on the postsynaptic cell.

Information is passed around between neurons in a neural network in order to complete a task or thought process. It is thought that “different cognitive processes can be bound together to give rise to perception, for example.”<sup>12</sup> This idea has inspired the field of Deep Learning and the development of Artificial Neural Networks.

*For the first time in my life, it made me feel like it might be possible to make some progress on a small part of the AI dream within our lifetime.*

— Andrew Ng

ANNs are used as a method of Machine Learning – an input is passed to a function in a neural model which computes an output. Combining these neural models together is useful for creating more complex algorithms. This concept is similar to human neural networks: inputs are passed via dendrites to the cell body, accumulating until an action potential fires, sending the electrical signals down the axon to be passed to another cell. It is thought that it may be possible to create an ANN to compute the same processes as a human brain does: thought, perception, creativity.

## Human Creativity

Humans are unbelievable when it comes to creativity: our ability to create unexpected twists in music and art, or behave unpredictably to confuse our enemies is incredible. It is perhaps the latter example that distinguishes us from other animals and has made randomness our evolutionary signature.

Randomness is a major key to survival. Deceive your opponent by acting unpredictably and you could win against your predator. It just happens that humans are extremely good at it, to an extent that they compete against other humans with randomness, such as in international wars. Due to evolution and natural selection, it has been thought that the randomness in our deception has also contributed to our creativity and even our mood swings.

This evolutionary signature may be a part of the human brain. It has raised the question: is the brain a random generator? When put to the test, it seems that it is not, as we are terrible at generating a random sequence of numbers when asked to – we hardly ever allow ourselves to choose the same number more than once in a row, or choose a distinct, but temporary, pattern. However, a different test is called matching pennies<sup>12</sup> – a game that involves two people competing face-to-face with the same number of coins. Each round the two players place two coins on a table and player A keeps both coins if they match (both heads or both tails), whereas B keeps them otherwise. This competitiveness has resulted in both players making their strategy as choosing as randomly as possible and test results by David Budescu, Fordham University, and Amnon Rapoport, University of California, have reported that the sequences of heads and tails generated by players came very close to “true mathematical randomness”.

Computers are much better random generators – or to be more precise, pseudorandom generators. What is meant by “pseudorandom” is that it is almost random or seems random to humans and cannot be easily predicted; computers, not unlike humans, cannot easily generate mathematically pure random numbers. Pseudorandom numbers are generated from choosing a

set digit in an already seemingly random number, such as pi, or in an input; for instance, the slight fluctuations in current are caused by random changes in the movement of particles, supposed by quantum mechanics, as well as general noise and stochastic resonance. The pseudorandomness in computers could be used as inputs into ANNs, or to alter existing inputs to change the behaviour and add slight creativity, just like the changes in the excitation of neurons caused by ephaptic coupling<sup>13</sup>.

## The Singularity

Could the Singularity – the point in time when Artificial Intelligence surpasses human intelligence – ever happen? Scientists from many fields have a view on Artificial Intelligence: there are those who accept a possibility of the technological Singularity<sup>14</sup>; others are either sceptical of its feasibility or in fear of a disastrous future we are building.

The fear of autonomous automation is a huge problem and there will need to be major regulations if we were ever to get close to the Singularity. Stephen Hawking told the BBC that “the development of full artificial intelligence (AI) could spell the end of the human race”.<sup>15</sup> It only takes a few minutes to think up many different uses of Artificial Intelligence that would make you shudder; imagine tiny autonomous bugs like the one shown in Figure 7<sup>16</sup>.

figure7 scariestml

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2016/08/figure7-scariestml-300x237.png>

**Fig. 7: The scariest use of machine learning**

Whether we are aiming to achieve the Singularity or simply approaching it, we need to take caution on what programs we write and how we share them. Not unlike our private data – such as bank details, passwords etc. – technology is confidential. Furthermore, the programs themselves need restrictions to only allow them to learn in certain cases for specific purposes.

To return to the abstract of this article, the question still remains. Could a computer think? That is a matter of your perspective. The brain is unbelievably complicated, yet may be able to be broken down into mathematical functions that could be run by a machine. Computer scientists are developing advanced algorithms, such as ANNs, in an attempt to improve a machine’s ability to learn, but how does running a program make a non-learning desktop computer suddenly a human-like genius? Is the Singularity realistic, or is it a far-flung dream in a parallel universe, too dangerous for us to even approach?

## References

1. Google DeepMind’s “AlphaGo”

[REFERENCE LINK](#)

2. Andrew Ng, “The Man Behind the Google Brain”

[REFERENCE LINK](#)

3. “Machine Learning” course by Coursera

[REFERENCE LINK](#)

4. “Supervised Learning”

[REFERENCE LINK](#)

5. “CS229 Lecture Notes”

[REFERENCE LINK](#)

6. “Loss function”

[REFERENCE LINK](#)

7. Quasi-Newton methods

[REFERENCE LINK](#)

8. BBC News: “Alan Turing”

[REFERENCE LINK](#)

9. Alan Turing and “Machine Intelligence”

[REFERENCE LINK](#)

10. “Introductory Psychology Image Bank”

[REFERENCE LINK](#)

11. "Synapse"  
REFERENCE LINK

12. "Chance" by New Scientist and Michael Brooks  
REFERENCE LINK

13. "Ephaptic coupling"  
REFERENCE LINK

14. "The Singularity is upon us? Not so fast"  
REFERENCE LINK

15. "Stephen Hawking warns artificial intelligence could end mankind"  
REFERENCE LINK

16. "The scariest use of machine learning"  
REFERENCE LINK