The Maths Behind Artificial Intelligence, by Antonia Tavares

Have you ever wondered how AI knows what show you should watch next? Or, how it's able to predict what you are going to say next, or how it is able to write essays? Maybe not, maybe you haven't even thought of it and are just grateful that AI can help you with your homework, or you simply enjoy giving it random prompts to see what it can come up with. But perhaps this essay can give you an insight into an AI's brain and the maths behind how it works.

Step 1: Linear Algebra

Data is stored as scalars, vectors, matrices, and tensors. Scalars are simply a single value. Vectors are a 1D array that represents a data point. Matrices are a 2D array that represents multiple data points. And tensors are a generalisation of vectors and matrices that are of higher dimensions.

$$Scalar: [7]$$

$$Vector: \begin{bmatrix} 5 & 1 & 0 \end{bmatrix} \; or \; \begin{bmatrix} 3 \\ 2 \\ 9 \end{bmatrix}$$

$$Matrix: \begin{bmatrix} 5 & 4 \\ 1 & 8 \end{bmatrix}$$

$$Tensor: \left[\begin{bmatrix} 5 & 9 \\ 6 & 0 \end{bmatrix} \quad \begin{bmatrix} 4 & 2 \\ 1 & 8 \end{bmatrix} \right]$$

Data is stored this way because it means that calculations can easily be done to manipulate the data into outputs quickly, especially with the use of tensors, where large datasets can be manipulated and processed at once.

So, what do we do with the data?

The Dot Product

The dot product is the scalar value produced when you dot two vectors of the same size.

A vector's size is expressed as the number of rows (m) by the number of columns (n) in the form $m \times n$. So, based off of the example vectors above, the first vector is a 1x3 vector and the second is a 3x1 vector.

When you dot two vectors, you are simply multiplying corresponding values and adding everything up:

$$\begin{bmatrix}7\\1\\0\end{bmatrix} \cdot \begin{bmatrix}2\\1\\8\end{bmatrix} = 7 \times 2 + 1 \times 1 + 0 \times 8$$

$$= 15$$

Or, as a general formula:

$$\begin{bmatrix}a\\b\\c\end{bmatrix} \cdot \begin{bmatrix}x\\y\\z\end{bmatrix} = ax + \ by + \ cz$$

The dot product tells us how similar two vectors are (how closely they line up if plotted on a graph), where a larger dot product means the vectors or more similar. This can be explained with the formula:

$$a \cdot b = ||a||b||cos\theta$$

Where a and b are vectors and θ is the angle between them. The more similar the two vectors are, the smaller θ is, and so the closer $cos\theta$ is to 1, making the dot product is larger. And when the dot product is 0, the vectors are not similar at all, are are actually perpendicular to each other on a graph.

Matrix Multiplication

Matrix multiplication is when you multiply matrices together (as the name would suggest). However, you cannot simply multiply any two matrices together. If you take the size of one matrix as $m_1 \times n_1$ and the size of another matrix as $m_2 \times n_2$, then $n_1$and m_2 must be the same. In other words, the number of columns of the first matrix must be equal to the number of rows in the second matrix. Then, when multiplying matrices, you are simply doing a bunch of dot products to get a resulting matrix that is of the size $m_1 \times n_2$.

For example:

You can multiply a 2x3 matrix (A) and a 3x3 matrix (B)to get a 2x3 matrix:

$$\begin{bmatrix}2 & 0 & 9\\4 & 3 & 3\end{bmatrix} \times \begin{bmatrix}4 & 0 & 9\\3 & 8 & 4\\0 & 1 & 8\end{bmatrix} = \begin{bmatrix}? & ? & ?\\? & ? & ?\end{bmatrix}$$

The first value in the resulting matrix (top left) would be the dot product of the first row in A and the first column of B:

$$\begin{bmatrix} 2 \\ 0 \\ 9 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 3 \\ 0 \end{bmatrix} = 2 \times 4 + 0 \times 3 + 9 \ \times 0 = 8$$

The next value (top middle) will be the dot product of the first row of A and the second column of B.

And the next value (top right) will be the dot product of the first row of A and the third column of B.

And the process is repeated for the final row of values, except the calculations are the dot product of the second row of A and each column of B to get the final matrix of:

$$\begin{bmatrix} 2 & 0 & 9 \\ 4 & 3 & 3 \end{bmatrix} \times \begin{bmatrix} 4 & 0 & 9 \\ 3 & 8 & 4 \\ 0 & 1 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 9 & 90 \\ 25 & 28 & 72 \end{bmatrix}$$

Dot products and matrix multiplication are the foundation of the maths behind AI. Within neural networks of an AI's brain, AI processes information by multiplying matrices of data and multiplying them by weight matrices to produce matrices that represent new information based on the original inputs. Weight matrices store values that determine how much a piece of data contributes to the output, since in reality, not all data have the same value and importance. At some neurons, biases can also be added in order to make outputs more accurate. These weights and biases never stay constant, and change as the AI gains new information and learns from past mistakes. Weights and biases are determined using calculus.

Step 2: Calculus

Calculus is essential for allowing AI to learn and improve. Derivatives and gradients allow for optimisation and reducing errors.

Derivatives

In a curve, the gradient is always changing. So, to find the gradient at any given point, you must differentiate the equation of the curve and find its derivative.

If you take any function of x, $f(x)$, you can approximate the gradient, $f'(x)$, between two points on the curve $y = f(x)$ where the two points are: $\left(x, f(x)\right)$ and $\left(x+h, f(x+h)\right)$ using gradient = change in y divided by change in x

$$f'(x) = \frac{f(x+h) - f(x)}{x+h-x}$$

$$= \frac{f(x+h) - f(x)}{h}$$

So, if $f(x) = x^2 + 2x$, the gradient would be:

$$f'(x) = \frac{(x+h)^2 + 2(x+h) - x^2 - 2x}{h}$$

$$= \frac{x^2 + 2xh + h^2 + 2x + 2h - x^2 - 2x}{h}$$

$$= 2x + h + 2$$

And as h gets smaller and smaller, the approximation of the gradient becomes closer and closer to the actual gradient of the curve, therefore the gradient of $f(x)$ can be found as h approaches 0.

$$f'(x) = \lim_{h \to 0} 2x + h + 2$$

Therefore, $f'(x)$ is simply $2x + 2$.

However, this calculation can get quite long for more complex equations, so a simple trick is that for each function of x, $f'(x) = ax^{a-1}$

So, if you look at $f(x) = x^2 + 2x$ as two separate functions of x: x^2 and $2x$, their derivatives would be $2x$ and 2, so you can add them together to get:

$$f'(x) = 2x + 2$$

<u>Chain Rule</u>

The chain rule is used to help find the derivative of an equation where there are nested functions. The formula for this rule is:

$$f'\left(g(x)\right) = f'\left(g(x)\right) + g'(x)$$

For example:

$$f(x) = (x^3 + 4x)^4$$

You can think of the bracket as just a single variable and differentiate normally:

$$u = x^3 + 4x$$

$$u^4 \to 4u^3$$

$$(x^3 + 4x)^4 \to 4(x^3 + 4x)^3$$

And then you differentiate what is inside the bracket (the nested function) and add it on:

$$x^3 + 4x \to 3x + 4$$

So:

$$f'(x) = 4(x^3 + 4x)^3 + 3x + 4$$

Partial Derivatives

When an equation uses multiple variables, partial derivatives find the rate of change when only one variable changes and the other(s) stay constant.

Therefore, you can differentiate normally and treat the other variables as constants.

For example:

$$f(x, y) = x^2y + xy^2$$

You can create two partial derivatives:

$$\frac{\partial f}{\partial x} = 2xy + y^2$$

$$\frac{\partial f}{\partial y} = x^2 + 2xy$$

The notation means the derivative of f with respect to x/y and the ∂ shows that it is a partial derivative.

So, how is this all used in AI?

Well, AI needs a way of knowing how to change its weights and biases to minimise its error. The first step is by finding out a quantitative value for how much it is wrong when it forms a predicted output. This is done by using a loss function.

There are different loss functions specific to different scenarios, but they are all formulas that use the predicted answer and the actual answer to give a value of how far off the AI is.

The next step would be to find the derivative of the loss with respect to the different weights and biases (how much the loss changes by when the weights and biases are adjusted). To efficiently calculate all of these derivatives, back propagation is used, where the AI will start with the output and work backwards through the loss function, finding its derivatives, until it reaches the inputs. However loss functions are complex and have many nested functions within them, and there are many weights and biases that contribute to the predicted output that will also affect the loss. This is where the chain rule and partial derivatives come in. The chain rule is used to find the derivative of the loss function and break the process down into manageable steps. And partial derivatives allow AI to work out the effect of only changing one weight/bias and keeping the rest constant so it can work out which weights and biases should be adjusted.

With this information, the AI repeatedly makes adjustments in the direction that reduces its loss at its learning rate. Its learning rate is simply how much it decides to adjust the weight and bias by. However, a learning rate too high causes it to overshoot the optimum point and constantly change directions, but one too low means that computation is wasted, since it takes longer to reach the solution. This means that an AI's learning rate also needs to be optimised, and a balance must be found between speed and accuracy.

This entire process is called gradient descent, and it is an iterative process that constantly gives feedback to the AI and allows it to learn.

Step 3: Probability and Statistics

But what happens when multiple possible solutions are produced? What if there is uncertainty and the AI is not sure which solution is best? In that case, probability distributions, Baye’s theorem and statistics are used to reach a conclusion.

<u>Bayes’ Theorem</u>

Bayes’ Theorem is a formula that is used to calculate conditional probabilities; the probability of an event occurring given that another event is already true. Therefore, it allows probabilities to be updated when new data is discovered.

The formula is:

$$P(A|B) = \frac{P(B|A) \times \ P(A)}{P(B)}$$

So, an AI can form a hypothesis (a statement that is being tested) and calculate the probability of the hypothesis being true based on a condition being true.

For example, the hypothesis could be that an email is spam. The AI may decide to calculate a new probability given the presence of the word “free”. Therefore, $P(A)$ would be the probability any email is spam, $P(B)$ is the probability that any email contains “free”, and $P(B|A)$ is the probability that “free” is in a spam email. If the probability is high enough, the AI may decide to label the email as spam. This is an example of the Naive Bayes classifier. Previously, the AI would have had a general probability for the email being spam, but it was able to update the probability that the email was spam when new data (the email contained “free”) was introduced to the AI.

<u>Probability Distributions</u>

An AI can create and use probability distributions to make decisions and determine how confident it is in its answer. When it processes data, it may produce multiple solutions but will often go with the solution that has the highest probability of being correct. However, if the highest probability is still quite small, then it’s less confident in its answer because there are other answers with similar probabilities of being correct.

There are many different types of probability distributions that an AI may use, depending on the type of data and its nature. When looking at temperatures, it handles continuous data, since temperature values are given within a range, and data points often centre around a mean, so it would use a normal distribution.

Whereas for determining the number of times an event will happen in a given time period, the data is discrete, since you can only have an integer number of events, so the poisson distribution would be used if the events occur at an average rate.

This is also why an AI will give a probability in its answers. Like, instead of telling you it will rain tomorrow, it says there is a 50% chance it will rain tomorrow.

Statistics

AI uses statistics in many ways to help its decision making. It can analyse data, by looking at the expectation of the data (the average), and the variance (how far data values are from the average), as well as the mean, mode, median and standard deviation (the square root of the variance). This helps the AI find trends in data, analyse new data, and make predictions, and create probabilities and probability distributions to aid its decision making.

Conclusion

Overall, AI is just linear algebra, calculus, probabilities, statistics, and *a lot* of functions. Except, AI can do thousands of millions of calculations per second. There are other functions and ideas that go into making an AI (you definitely cannot fit everything in to just 2000 words) but hopefully you enjoyed reading this essay and learnt something new!

References:

1. https://youtu.be/_DSXk3MePVw?si=WPXvZAHKpoGRjgCW
2. https://www.gilesthomas.com/2025/02/basic-neural-network-matrix-maths-part-1
3. https://www.ibm.com/think/topics/linear-algebra-for-machine-learning
4. https://medium.com/@rishabalagharu/derivatives-and-their-use-in-ai-e65ae5ba80f3
5. https://www.ibm.com/think/topics/backpropagation
6. https://www.ibm.com/think/topics/gradient-descent
7. https://medium.com/@sujathamudadla1213/common-probability-distributions-used-in-machine-learning-and-their-applications-64eed2784dfd