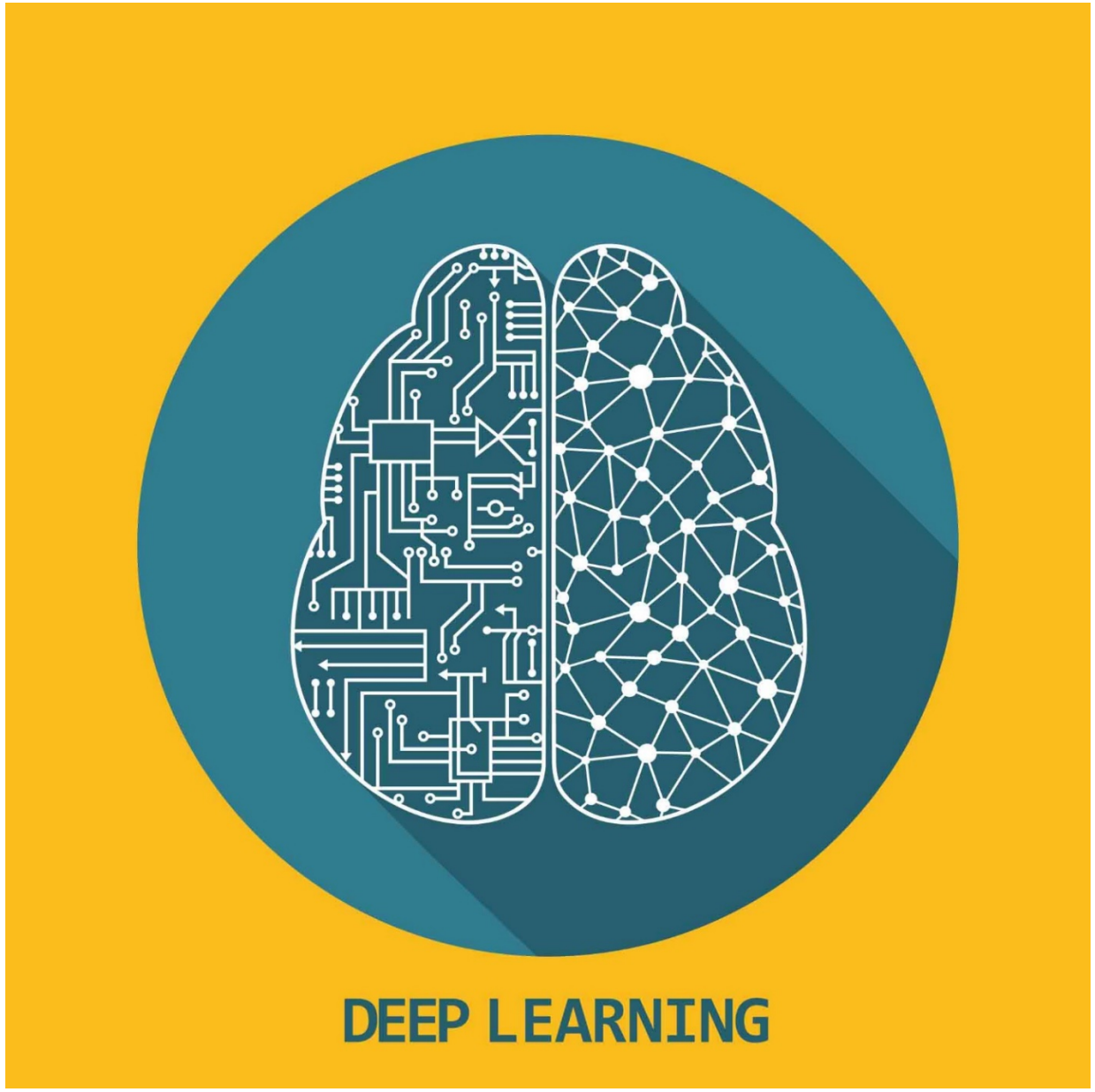
greatlearning
Power Ahead



Introduction to Deep Learning

greatlearning
Power Ahead

Agenda

01 What is Machine Learning?

02 What is Deep Learning?

03 Why Deep Learning?

04 Maths Prerequisites

05 Terms of Deep Learning

06 Frameworks for Deep Learning

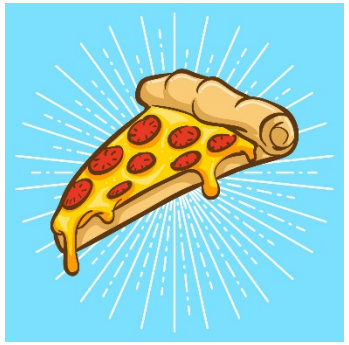
07 Demo

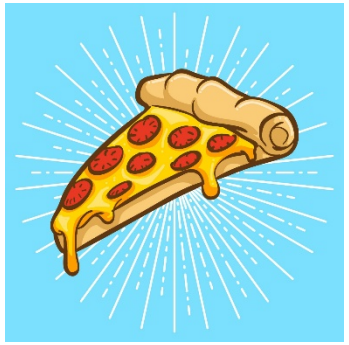
08 Applications of Deep Learning

Machine Learning

"A computer program is said to **learn** from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."

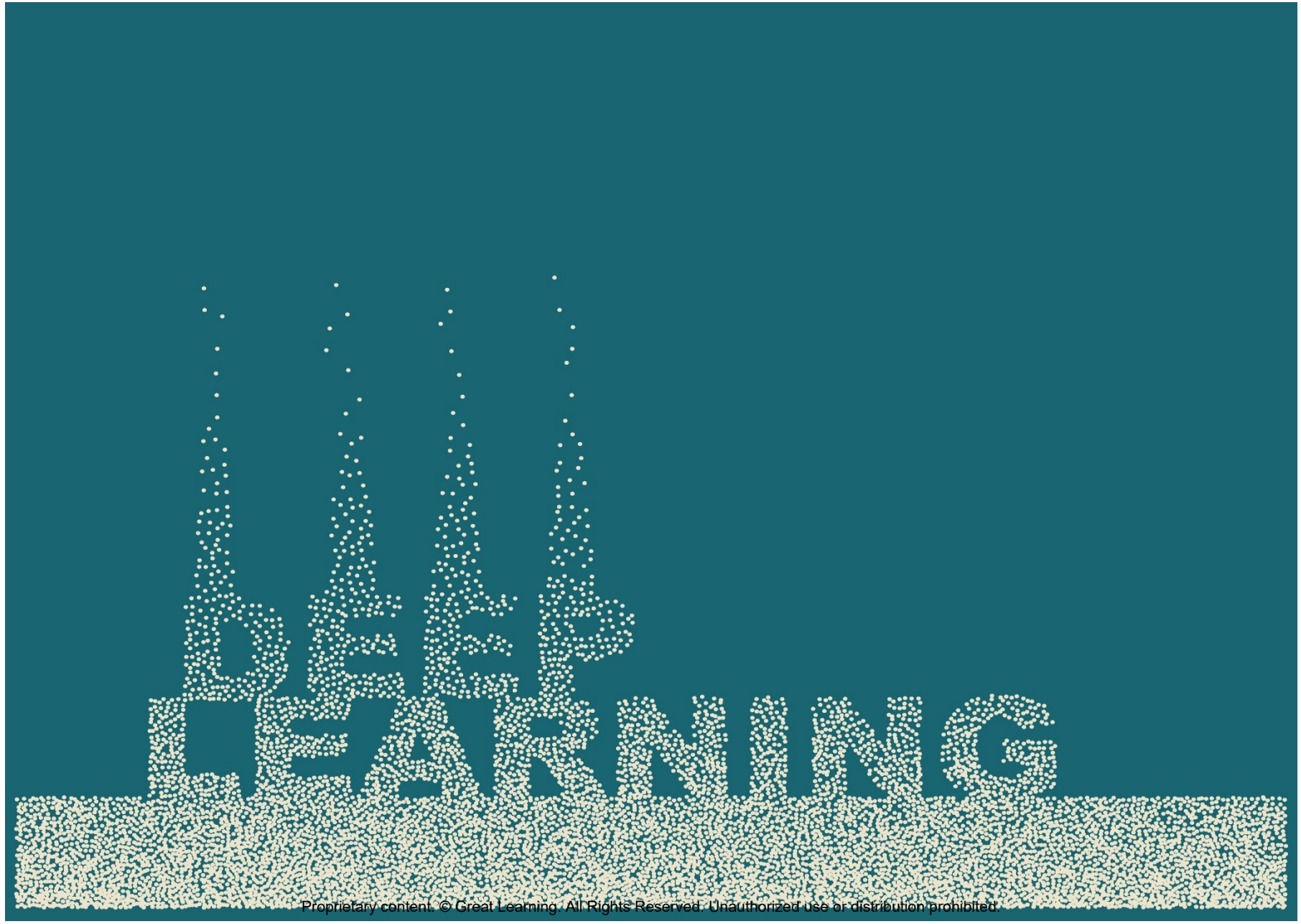
- Tom M. Mitchell





Limitations of Conventional Machine Learning

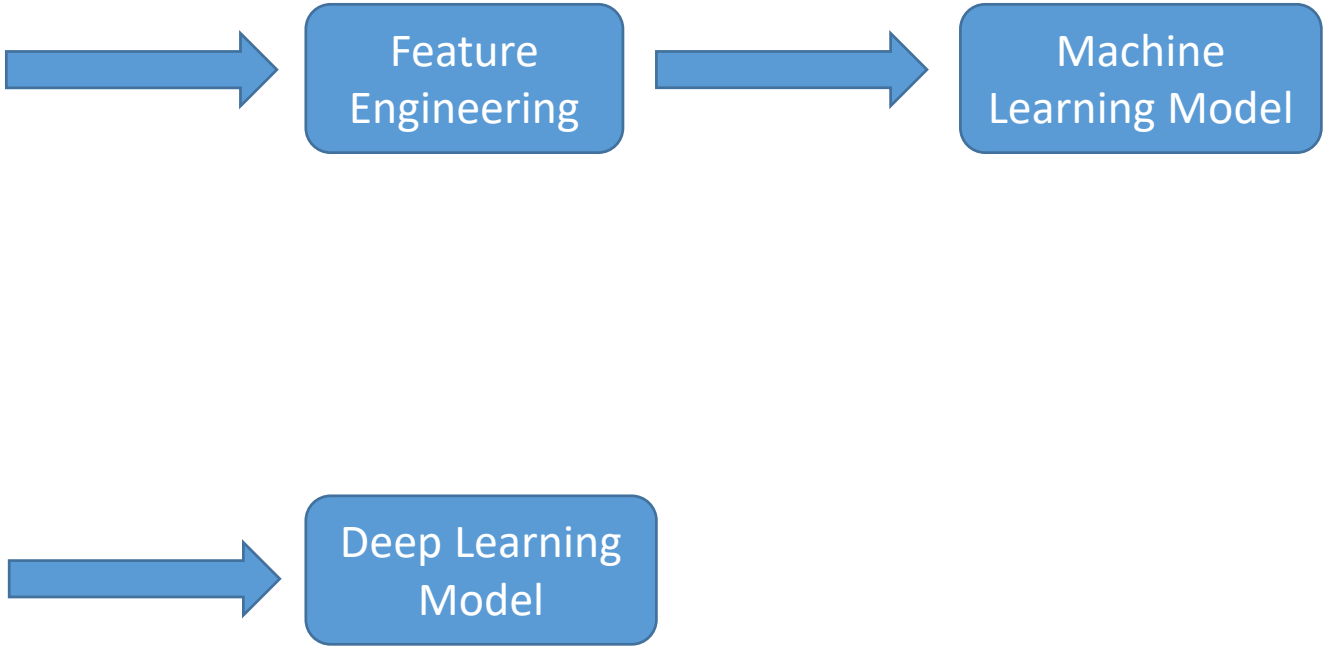
- Conventional Machine Learning requires data preprocessing and feature engineering.
- Conventional Machine Learning algorithms are probabilistic in nature, thus fails in use-cases such as Sequence prediction.



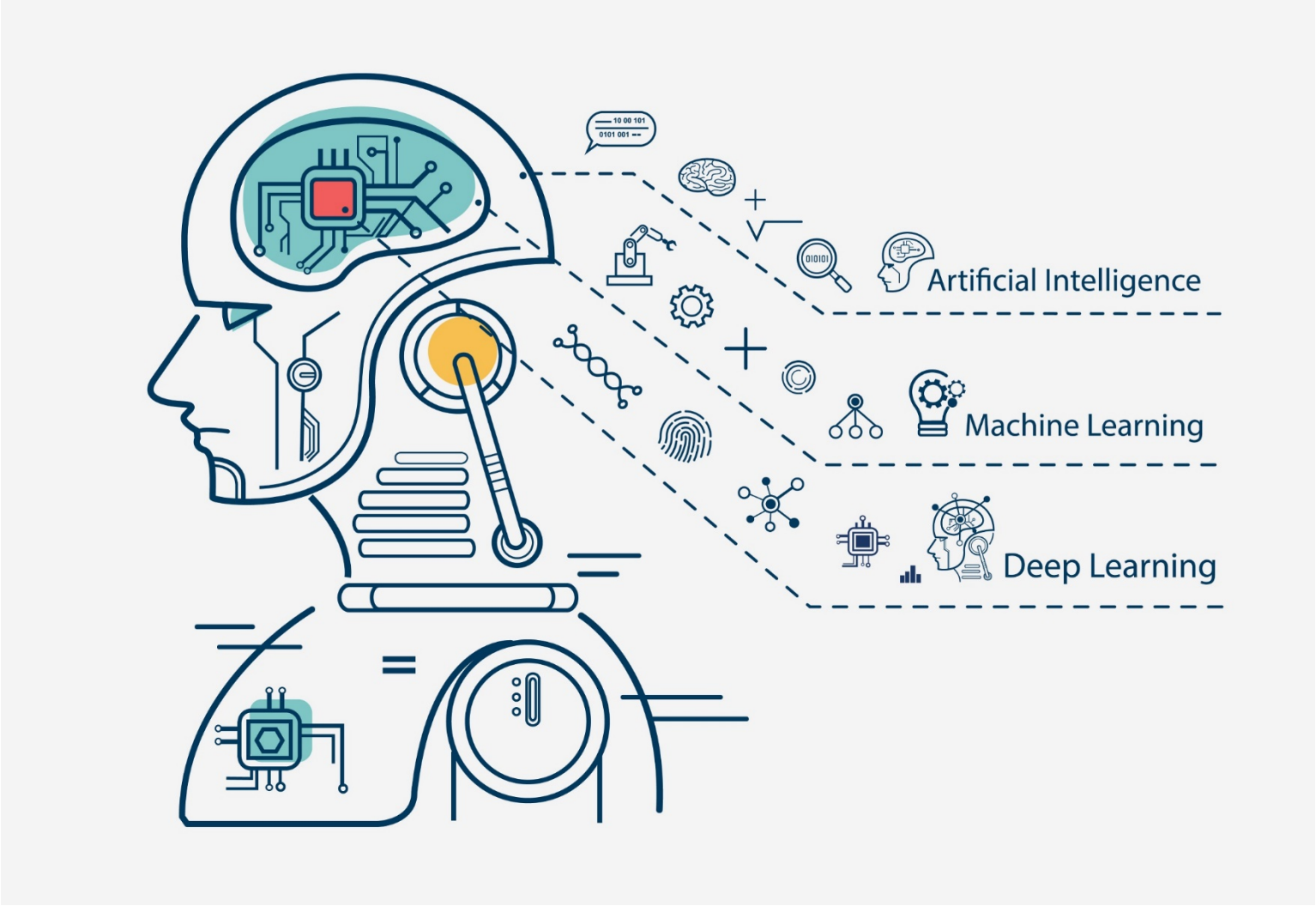
Deep Learning

- Subset of Machine Learning
- Trains the model using large datasets
- Training is done using multiple layers

Deep Learning

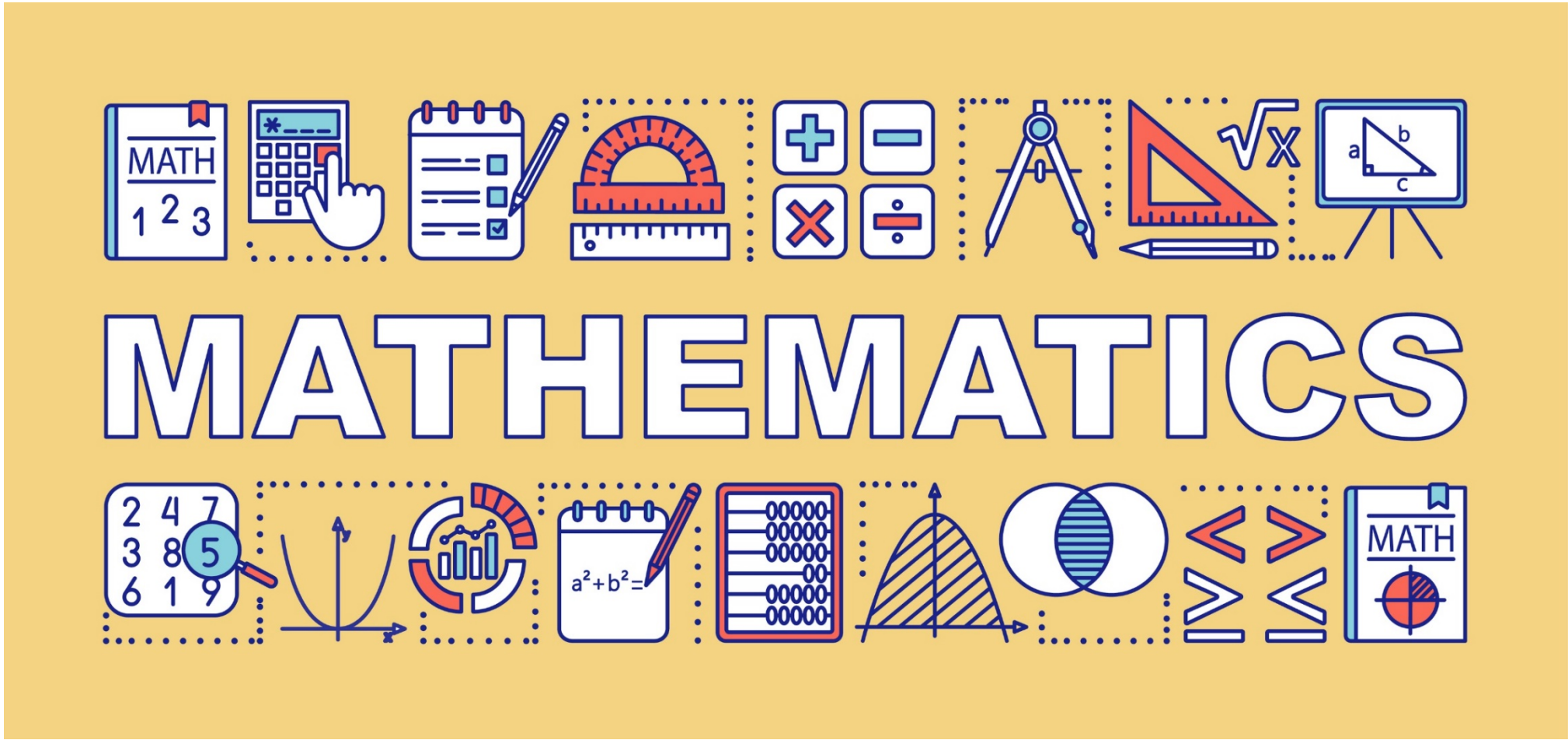


ML vs DL vs AI



Why Deep Learning?

- Deep Learning does feature engineering on its own.
- They perform better when we are dealing with huge datasets.
- Deep Learning performs better with image data, text data.

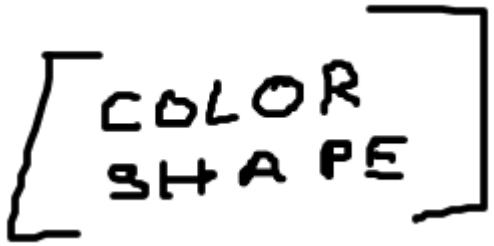


Linear Algebra

- Tensors
- Scalars

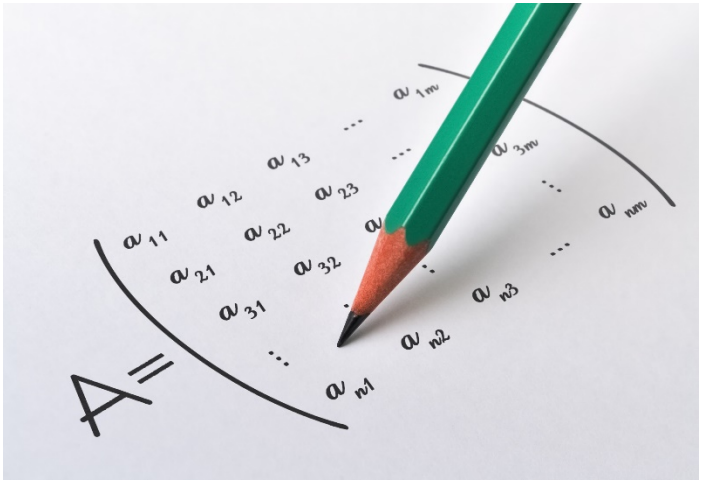
Linear Algebra

- Vectors



Linear Algebra

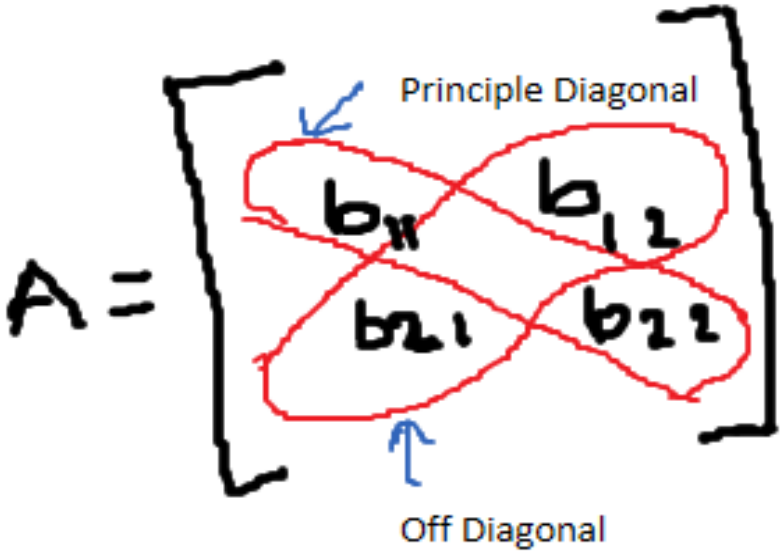
- Matrix



$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Linear Algebra

- Determinant of a Matrix



Linear Algebra

$$|A| = |b_{11} \times b_{22} - b_{12} \times b_{21}|$$

$$M = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

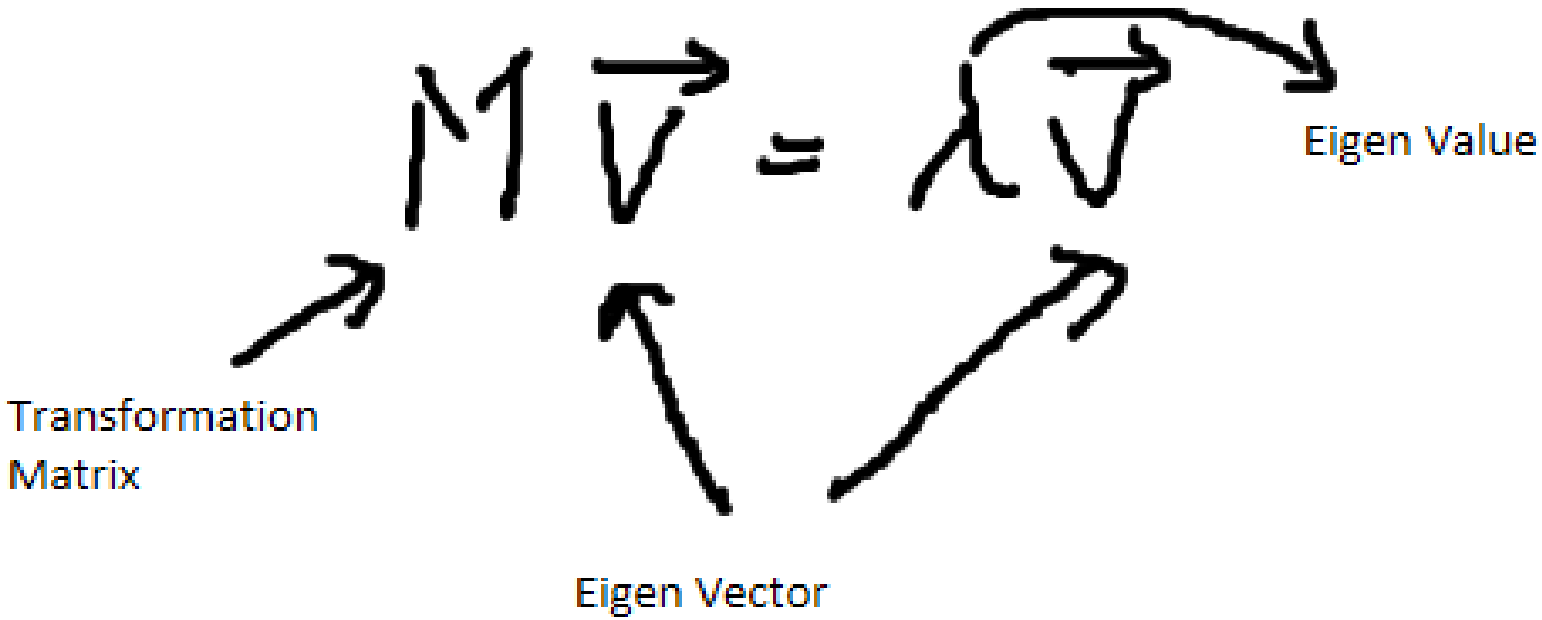
$$|M| = |2 \times 5 - 4 \times 3|$$

$$= |10 - 12|$$

$$= -2$$

Linear Algebra

- Eigen Value and Eigen Vector



Linear Algebra

- Eigen Value and Eigen Vector

$$M\vec{v} = \lambda\vec{v}$$

$$M\vec{v} = \lambda I\vec{v}$$

$$M\vec{v} - \lambda I\vec{v} = 0$$

$$(\underbrace{M - \lambda I}_{A})\vec{v} = 0$$

$$\det(M - \lambda I) = 0$$

Linear Algebra

$$M = \begin{pmatrix} 1 & 2 \\ 3 & 0 \end{pmatrix} \quad \vec{v} \neq 0$$

$$\det\left(\begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\lambda\right) = 0$$

$$\det\begin{pmatrix} 1-\lambda & 2 \\ 3 & -\lambda \end{pmatrix} = 0$$

$$\lambda^2 - \lambda - 6 = 0$$

$$\lambda^2 - 3\lambda + 2\lambda - 6 = 0$$

$$\lambda(\lambda - 3) + 2(\lambda - 3) = 0$$

$$(\lambda - 3)(\lambda + 2) = 0$$

$$\lambda = 3 \quad \text{or} \quad \lambda = -2$$

Linear Algebra

$$\begin{pmatrix} 1 & 2 \\ 3 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3x_1 \\ 3x_2 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ or } \begin{pmatrix} -2 \\ -2 \end{pmatrix} \text{ etc}$$

$$x_1 - 2x_2 = 3x_1$$

$$-2x_1 = -2x_2$$

$$x_1 = x_2$$

$$3x_1 = 3x_2$$

$$x_1 = x_2$$

Calculus

- Derivatives

$$f(x) \longrightarrow f'(x)$$

$$f(x) = 2x$$

$$f'(x) = \frac{d}{dx}(2x)$$

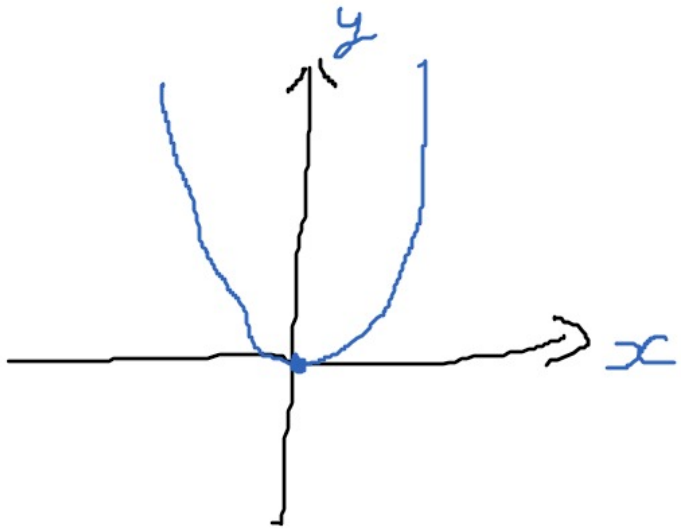
$$f'(x) = 2$$

- Derivative of tensor is called Gradient.

Calculus

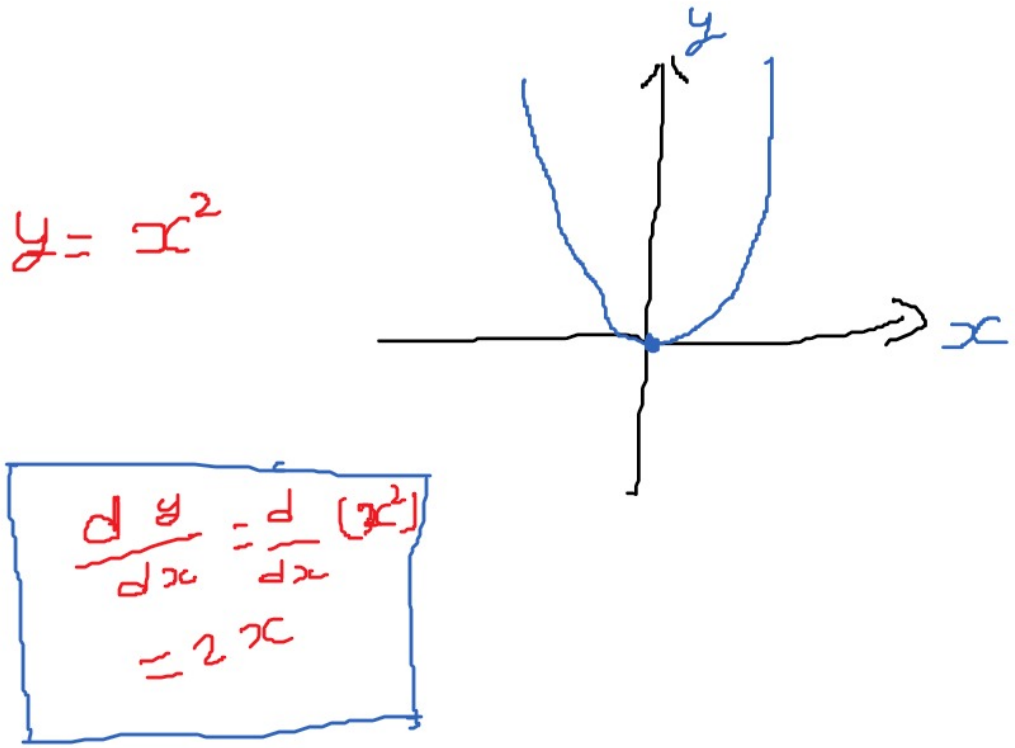
- Gradient Decent

$y = x^2$



Calculus

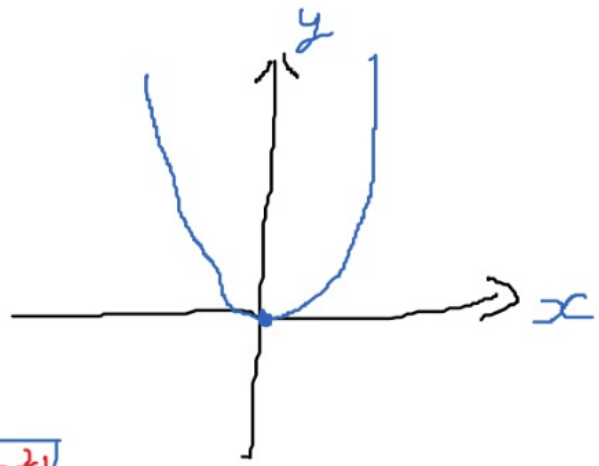
- Gradient Decent



Calculus

- Gradient Decent

$y = x^2$



$$\frac{dy}{dx} = \frac{d}{dx}(x^2)$$
$$= 2x$$

$x = 6, \quad h = 0.1$

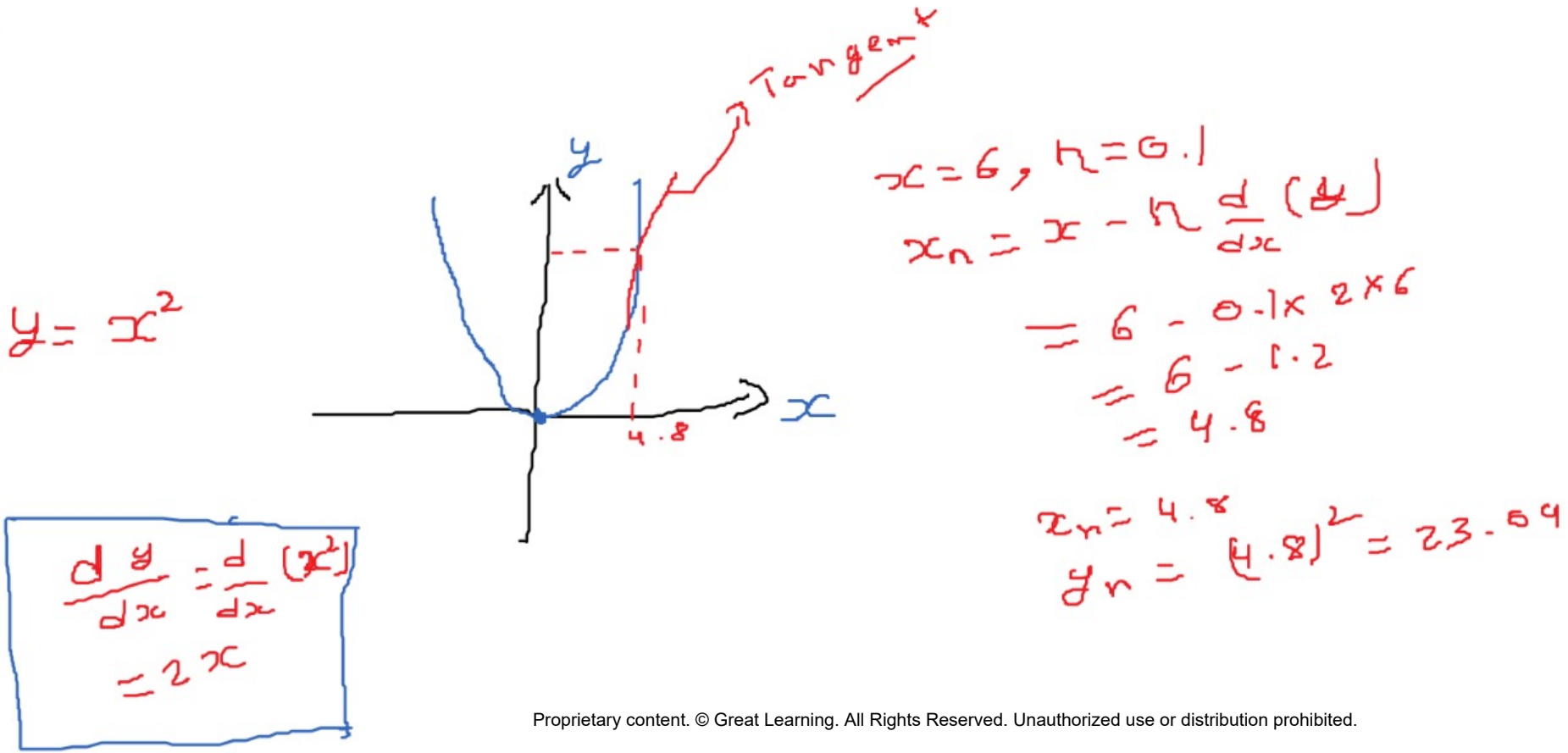
$$x_n = x - h\frac{d}{dx}(y)$$
$$= 6 - 0.1 \times 2 \times 6$$
$$= 6 - 1.2$$
$$= 4.8$$

$x_n = 4.8$

$y_n = (4.8)^2 = 23.04$

Calculus

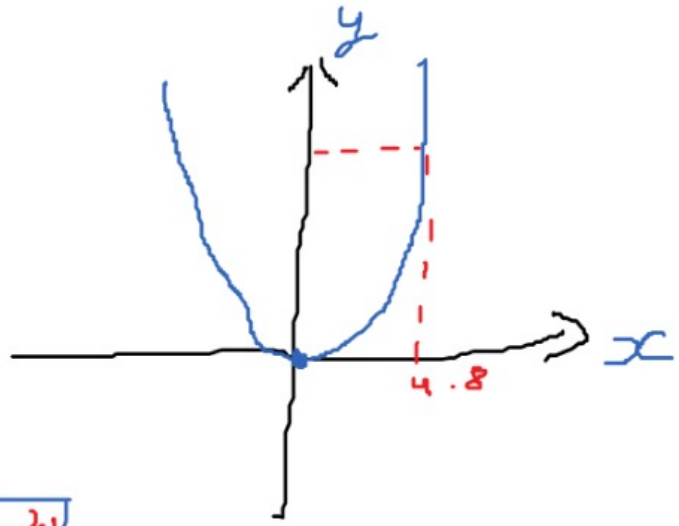
- Gradient Decent



Calculus

- Gradient Decent

$y = x^2$



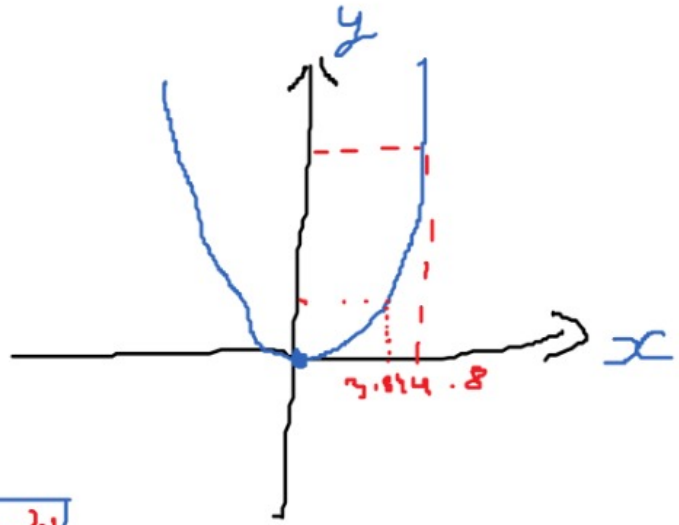
$$x_n = 4.8 - 0.1 \times 2 \times 4.8$$
$$= 3.84$$
$$y_n = (3.84)^2 = 14.74$$

$$\frac{dy}{dx} = \frac{d}{dx}(x^2)$$
$$= 2x$$

Calculus

- Gradient Decent

$y = x^2$



$x_n = 4.8 - 0.1 \times 2 \times 4.8$

$= 3.84$

$y_n = (3.84)^2 = 14.74$

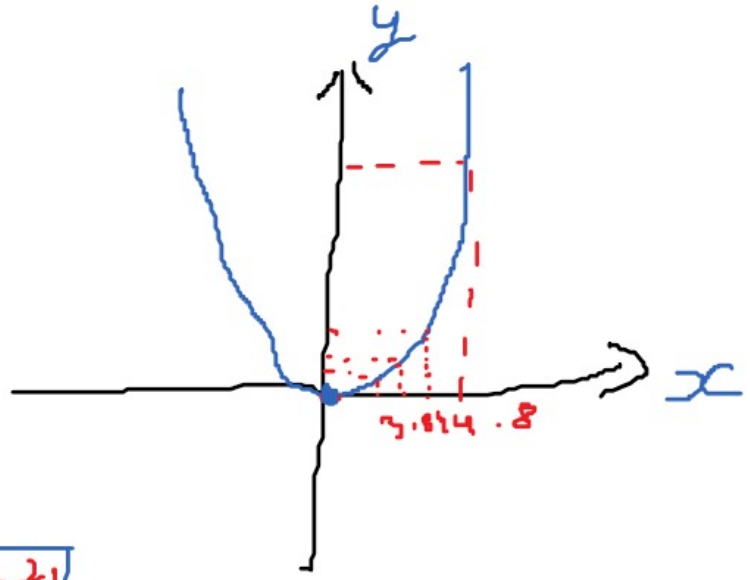
$\frac{dy}{dx} = \frac{d}{dx}(x^2)$

$= 2x$

Calculus

- Gradient Decent

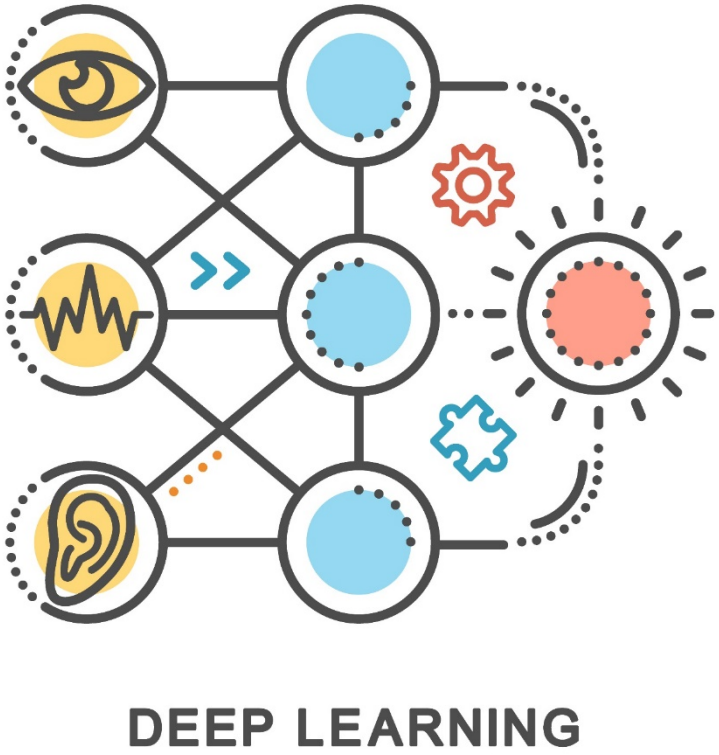
$y = x^2$



$$\frac{dy}{dx} = \frac{d}{dx}(x^2)$$

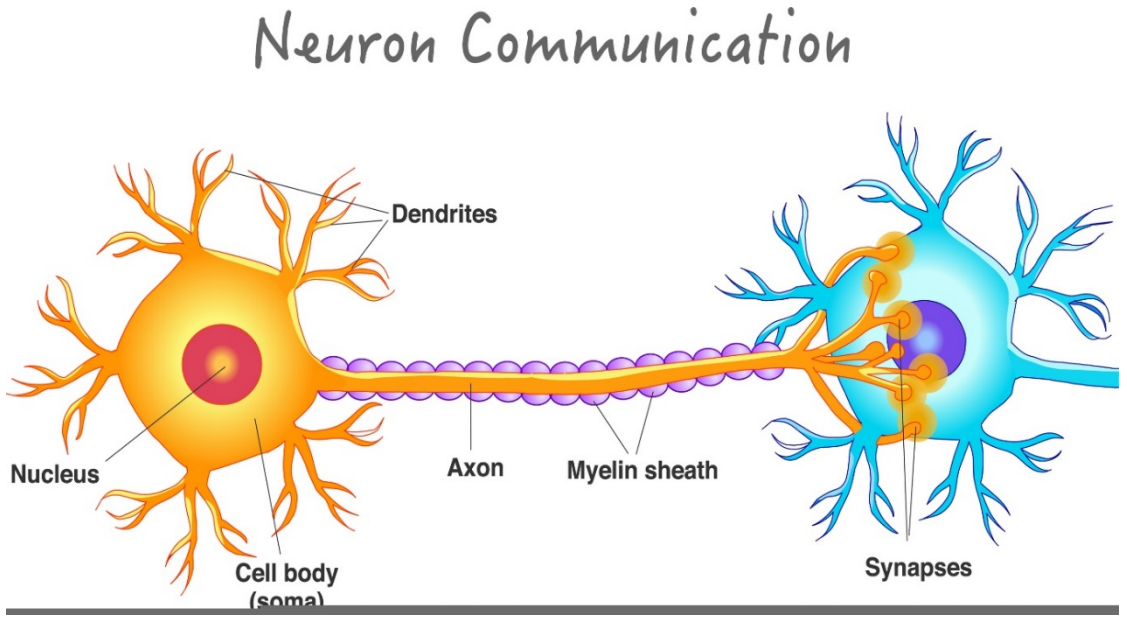
$$= 2x$$

Terminologies Related To Deep Learning

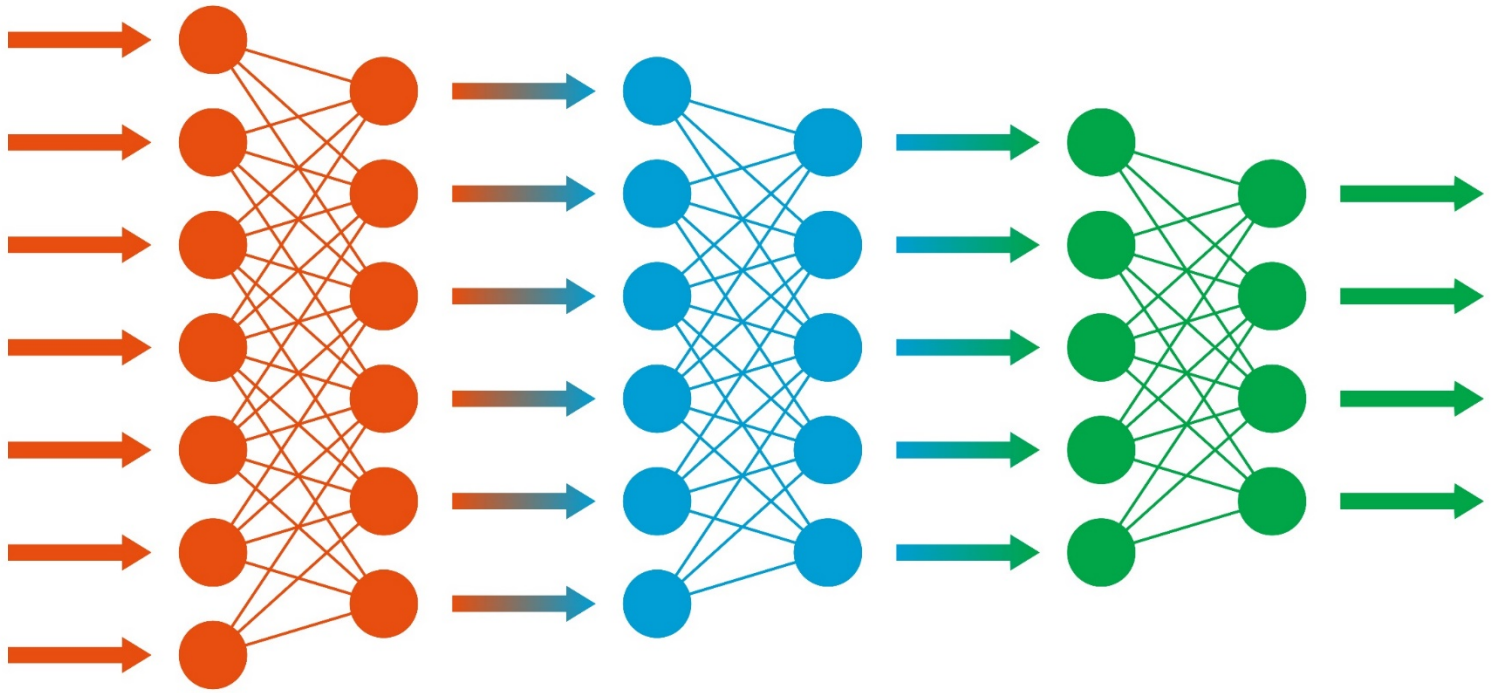


Neural Networks

- Neural Networks mimic the working of human brain.
- Information flows from one neuron to another via axon of first to dendrites of the other.
- In Neural Networks, neurons(nodes) receives input signal(input values) which passes through neuron and delivers output signal.



Deep Neural Network



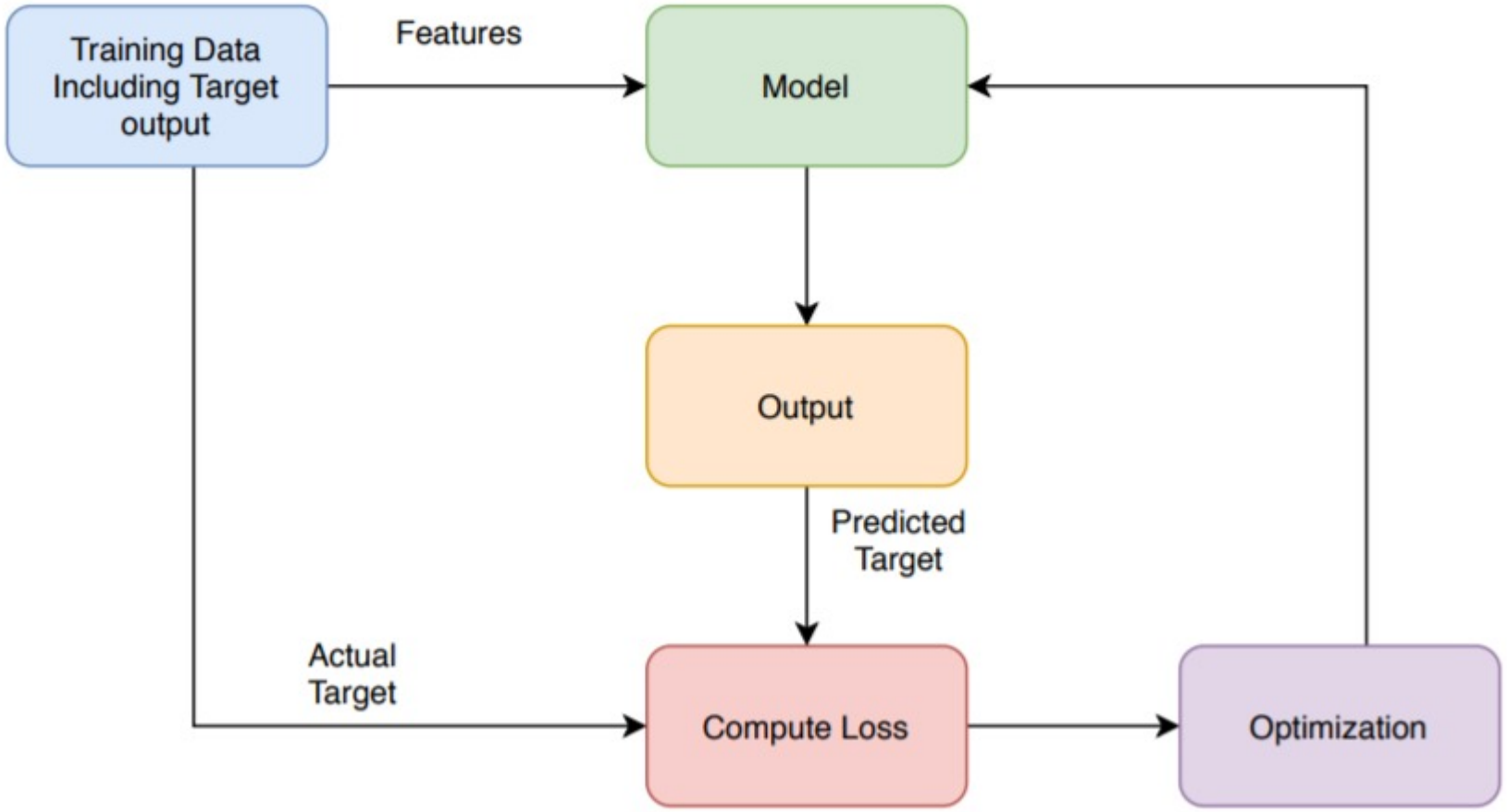
Deep Neural Networks

Component	Functionality
Input Layer	Takes the input data such as text, number, image, audio, video.
Hidden Layer	Performs calculations on the basis of weighted sum.
Activation Function	Maps the output values ranging from 0 to 1 or from -1 to 1.
Output Layer	Provides the desired output.

Deep Neural Networks

Activation Function	Output Range
Relu	Converts negative values to positive
Tanh	-1 to 1
Sigmoid	0 to 1
Softmax	Value ranges based on probability distribution.

Deep Neural Network



Deep Neural Network

Frameworks for Deep Learning

- **TensorFlow**
- **Keras**
- PyTorch
- DL4J
- Caffe

Demo

Applications of Deep Learning

Applications of Deep Learning
Self Driving Cars
News Aggregation and Fraud News Detection
Virtual Assistants
Healthcare: Detecting Developmental Delay in Children
Automatic Game Playing
Adding sounds to silent movies
Automatic Machine Translation
Deep Dreaming
Demographic and Election Predictions

greatlearning
Power Ahead