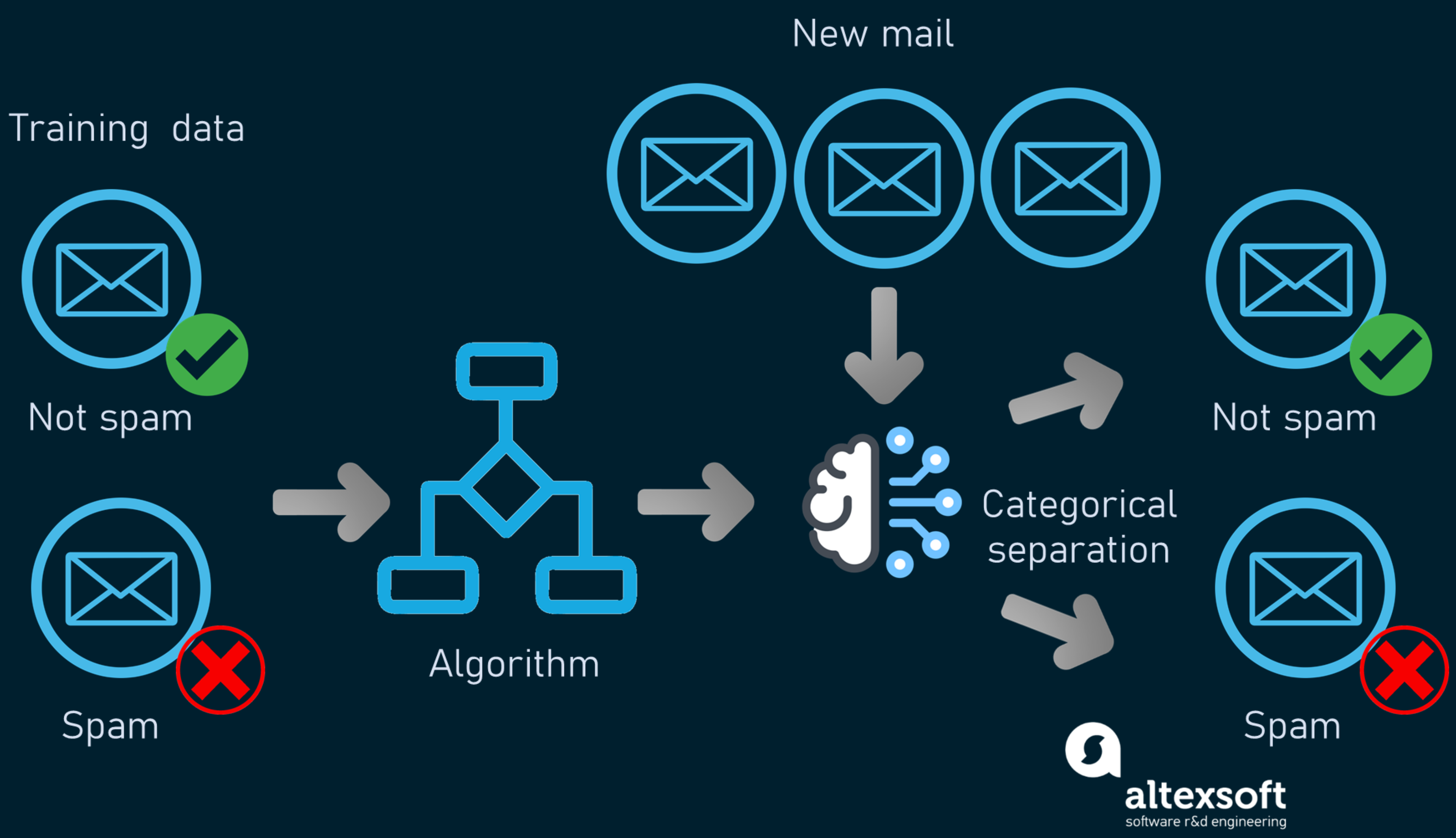
NEURAL NETWORKS

The fundamentals

Machine learning in spam filtering



SHOWCASE OF DATA SCIENCE DISCIPLINES & PROBLEMS WITH MRI IMAGE RECOGNITION

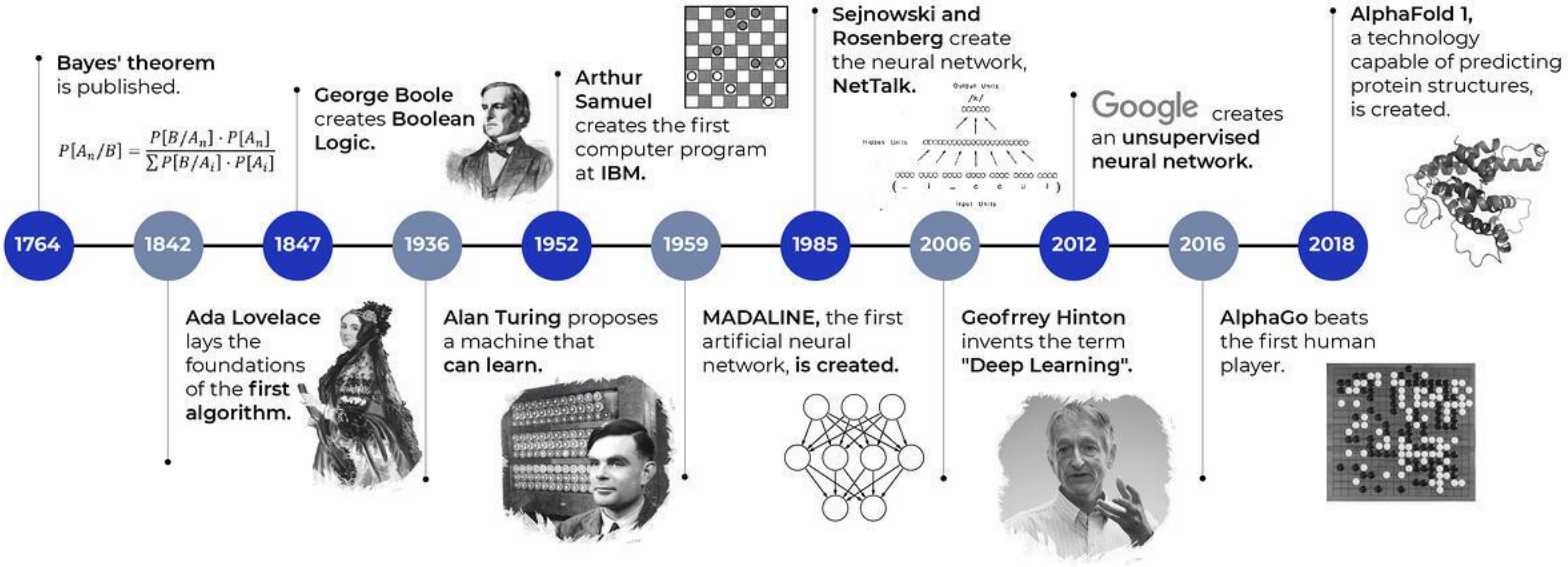
Data science	Help physicians with recognizing medical conditions on MRI scans
Data mining	Collect, prepare, label, and segment MRI scans data by types of medical conditions
ML/Deep learning	Recognize and classify diseases on MRI scans
AI	Build decision-support system for MRI image diagnostics

https://www.altexsoft.com/blog/data-science-artificial-intelligence-machine-learning-deep-learning-data-mining/

MACHINE LEARNING APPLICATION AREAS:

- **Recommendation algorithms:** FB feed, product recommendations
- **Image analysis/object detection:** cars in parking lots, face recognition
- **Fraud detection:** fraudulent credit card transactions
- **Chat bots:** Interactive machine responses that imitate human response
- **Manufacturing:** Predictive maintenance and condition monitoring
- **Retail:** Upselling and cross-channel marketing
- **Healthcare and life sciences:** Disease identification and risk satisfaction
- **Travel and hospitality:** Dynamic pricing
- **Financial services:** Risk analytics and regulation
- **Energy:** Energy demand and supply optimization

MACHINE LEARNING TIMELINE

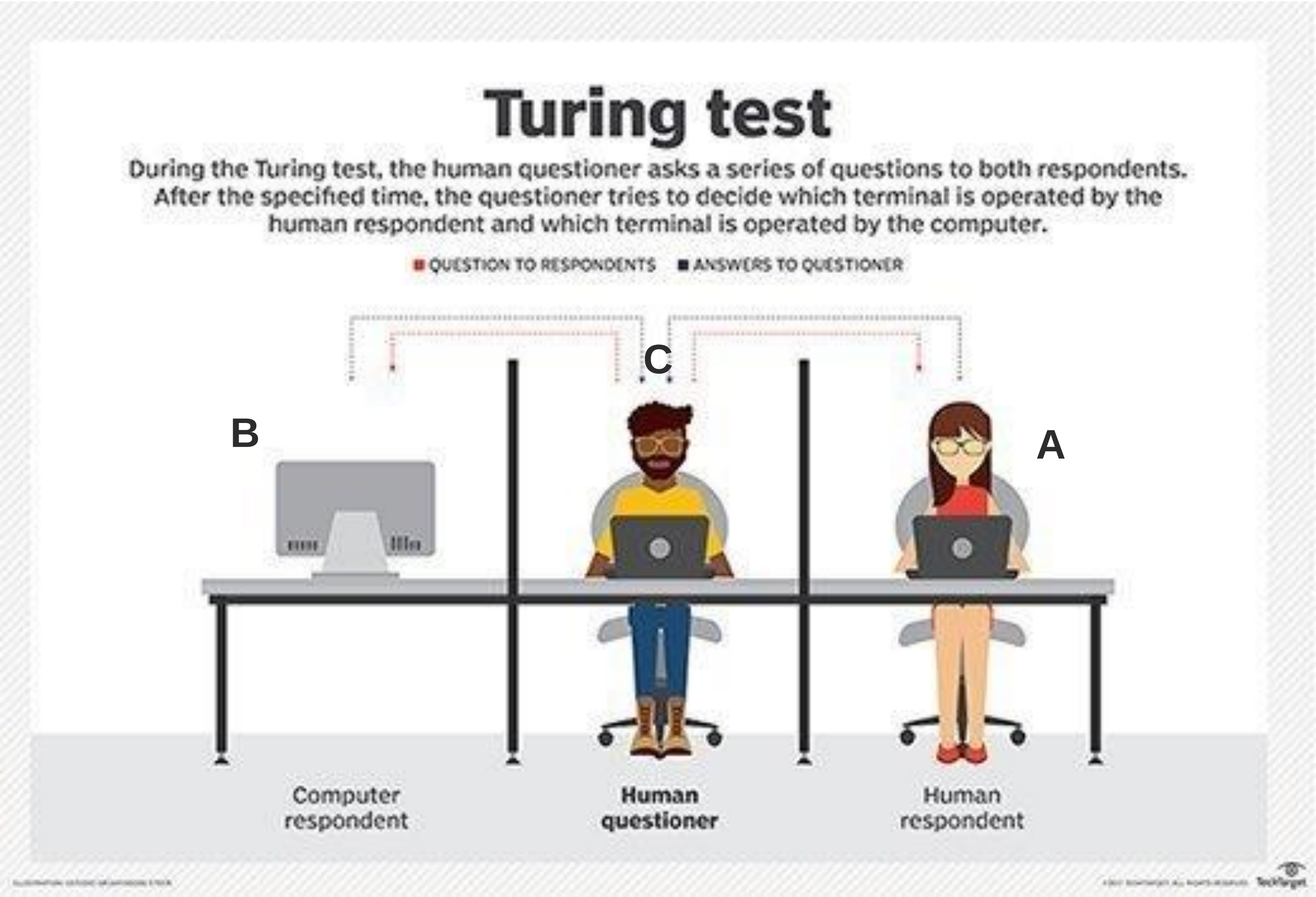


The Imitation Game

ALAN TURING

- British logician and computer pioneer **Alan Mathison Turing**.
- **London, 1947:** *“What we want is a machine that can learn from experience,” and that the “possibility of letting the machine alter its own instructions provides the mechanism for this.”*
- **Turing test:** Determine whether a computer or a machine can think intelligently like humans”?

ALAN TURING



C(Interrogator): Are you a computer?
A(Computer): No

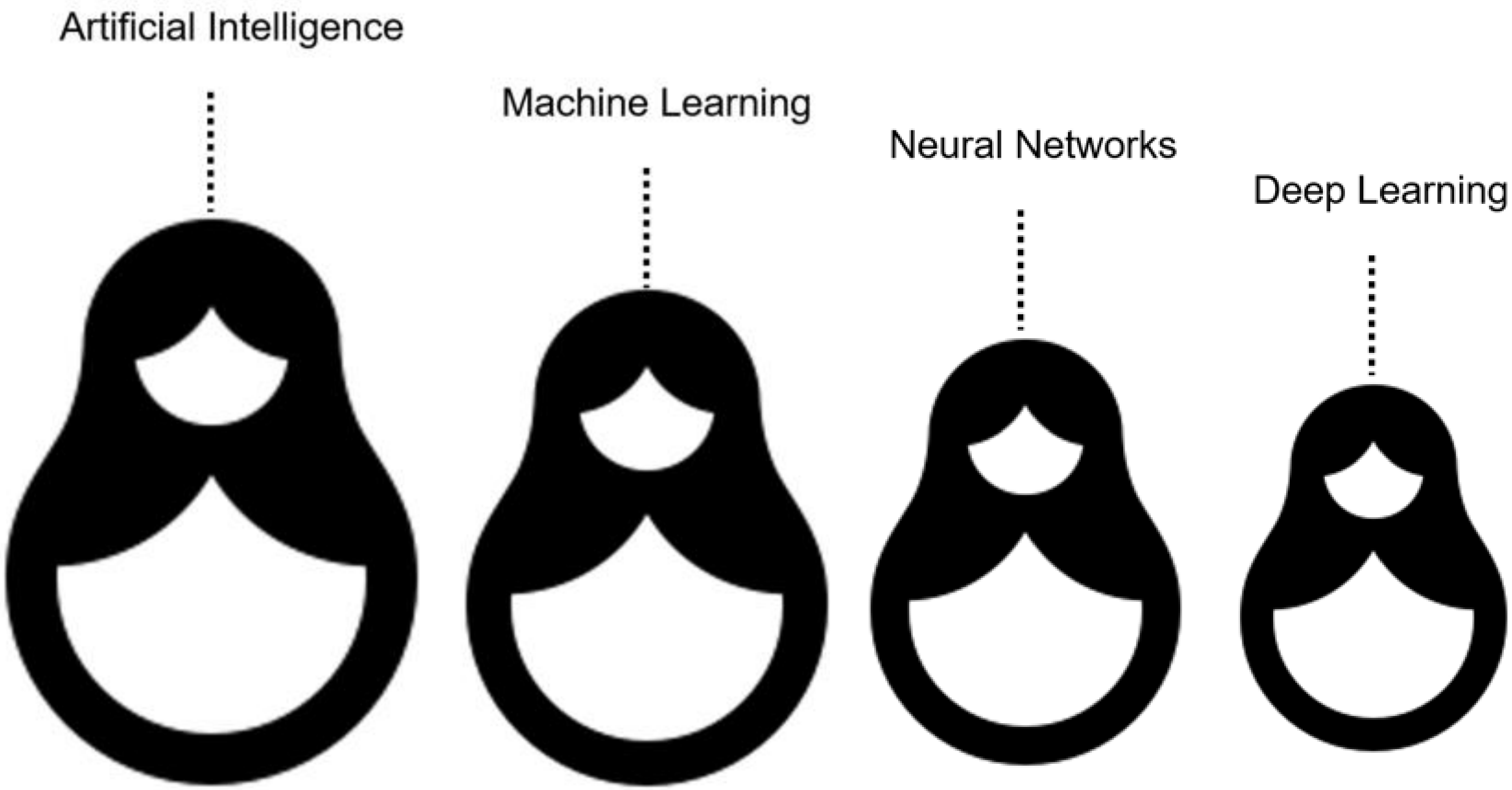
C: Multiply one large number to another, 158745887 * 56755647
A: After a long pause, an incorrect answer!

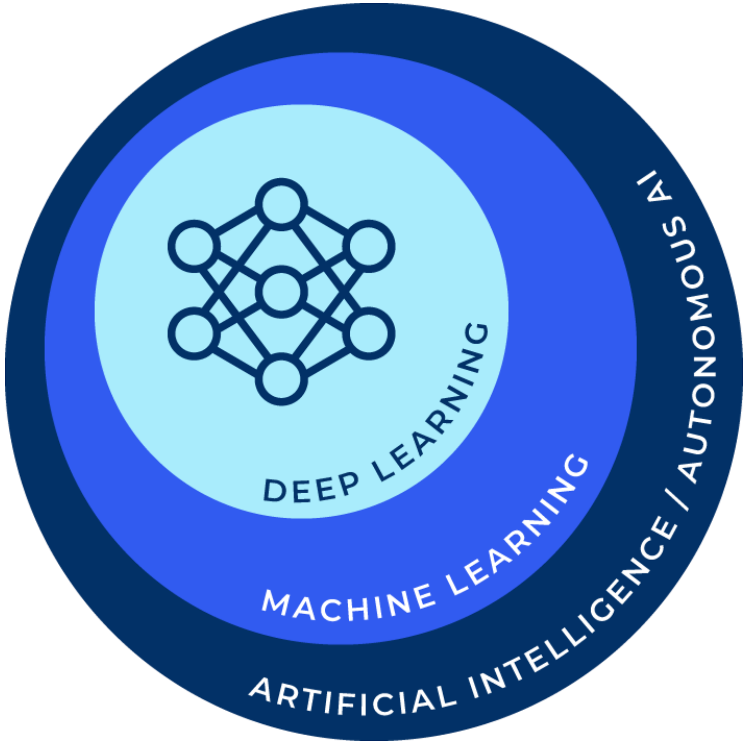
C: Add 5478012, 4563145
A: (Pause about 20 seconds and then give as answer)10041157

ARTHUR SAMUEL

- American pioneer in the field of computer gaming and artificial intelligence.
- **1959:** Popularized the term "**machine learning**“
- Defined it as the “*field of study that gives computers the ability to learn without being explicitly programmed*”.

AI VS ML VS ANN VS DL





https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks
https://www.algotive.ai/blog/machine-learning-what-is-ml-and-how-does-it-work

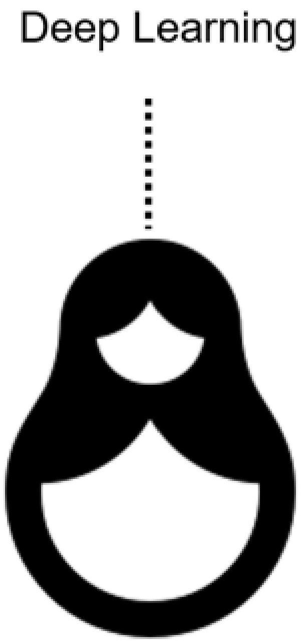
ARTIFICIAL INTELLIGENCE.

- **Highest Level**
  - Leveraging **computers or machines** to ***mimic the problem solving*** and decision-making capabilities of the **human mind**

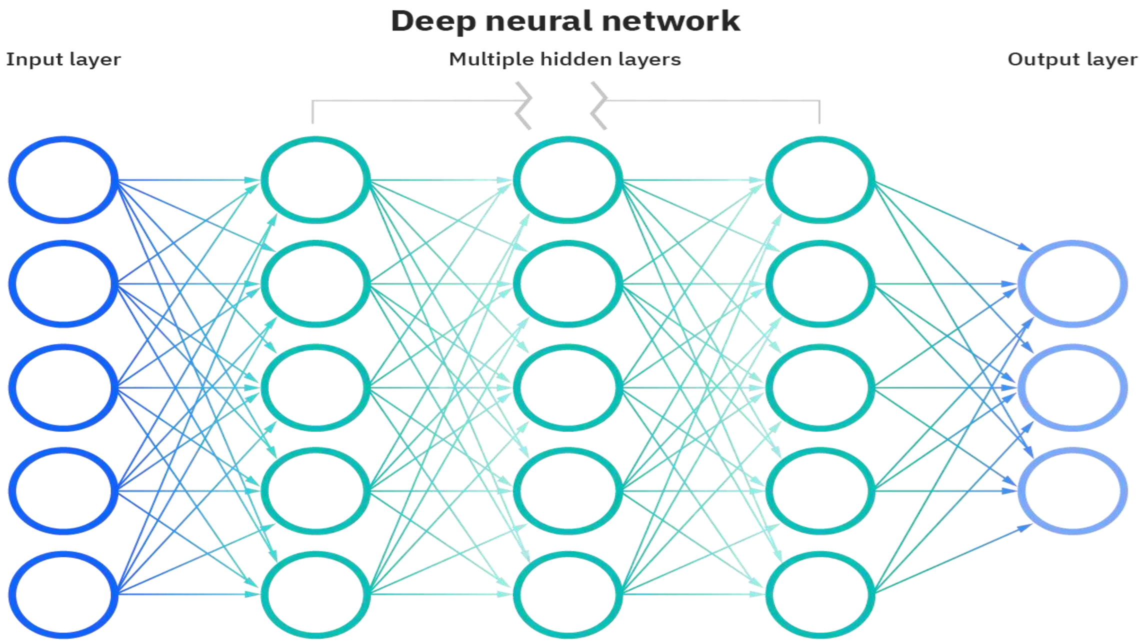
MACHINE LEARNING.

- **Subset of AI**
- Includes **self-learning algorithms** ***that derive knowledge*** from data in order **predict outcomes**

DEEP LEARNING.



- Another subset of ML within AI
- Scalable ML
- Automates process and eliminates human intervention and uses large datasets



https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

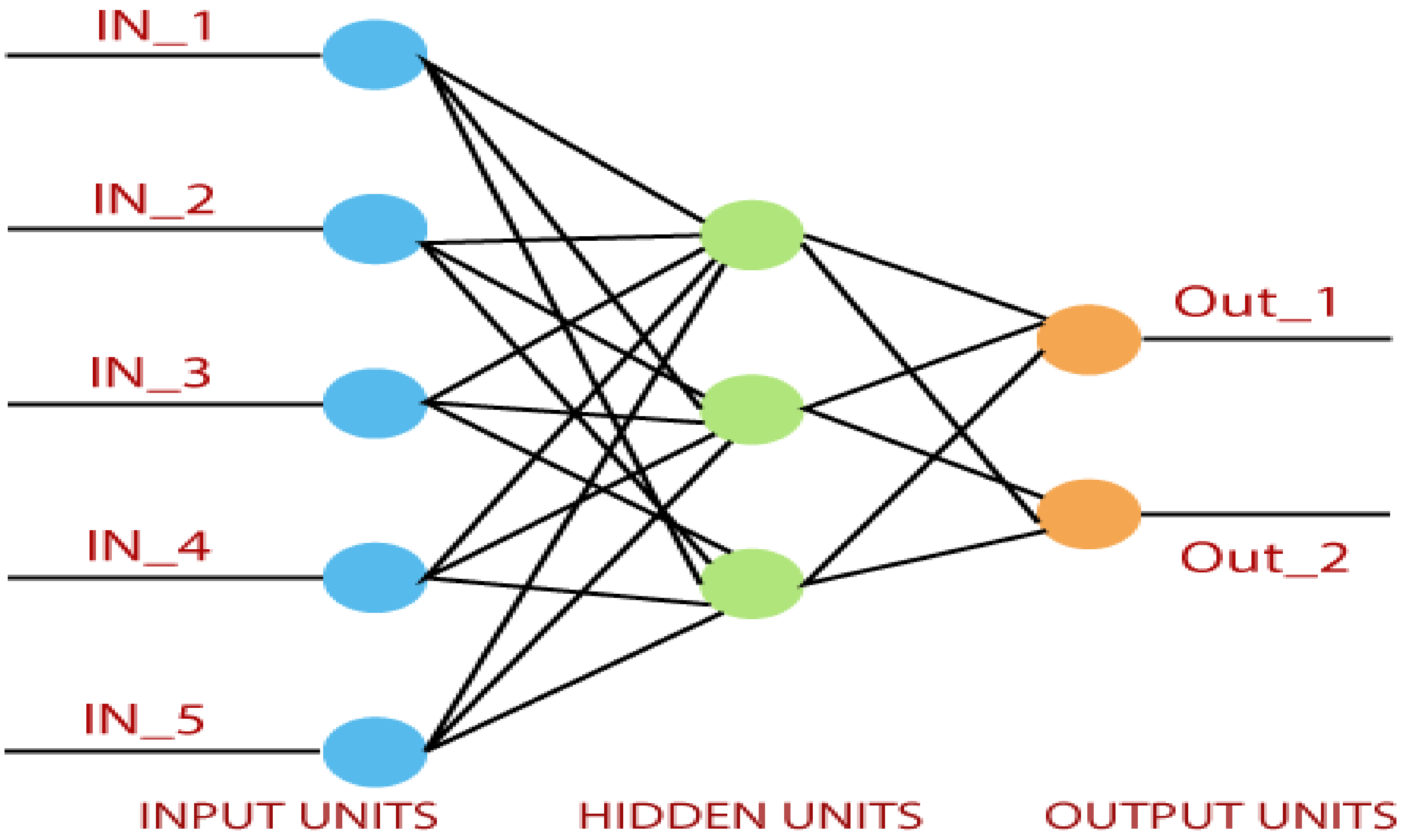
TYPES OF MACHINE LEARNING.

- **SUPERVISED LEARNING**
  - **Labelled datasets** to train algorithms to classify data or predict outcomes Example: Yes/No or categorical
  - **Classification models**
    - Customer retention: Minimizing customer churn (telecom example)
    - Historical data use **Classification models** to build a supervised learning model
  - **Regression models**
    - Build equation to generate an estimate for an output value
    - Airlines charges for flights (predict accurate currency) to maximise their profits
- **UNSUPERVISED LEARNING**
  - Analyze and cluster unlabelled datasets
  - Helps find patterns and groups without the need for human intervention
  - CLUSTERING
    - Customer segmentation: Targetted/Effective marketing (customer similarity or not)
  - DIMENSIONALITY REDUCTION

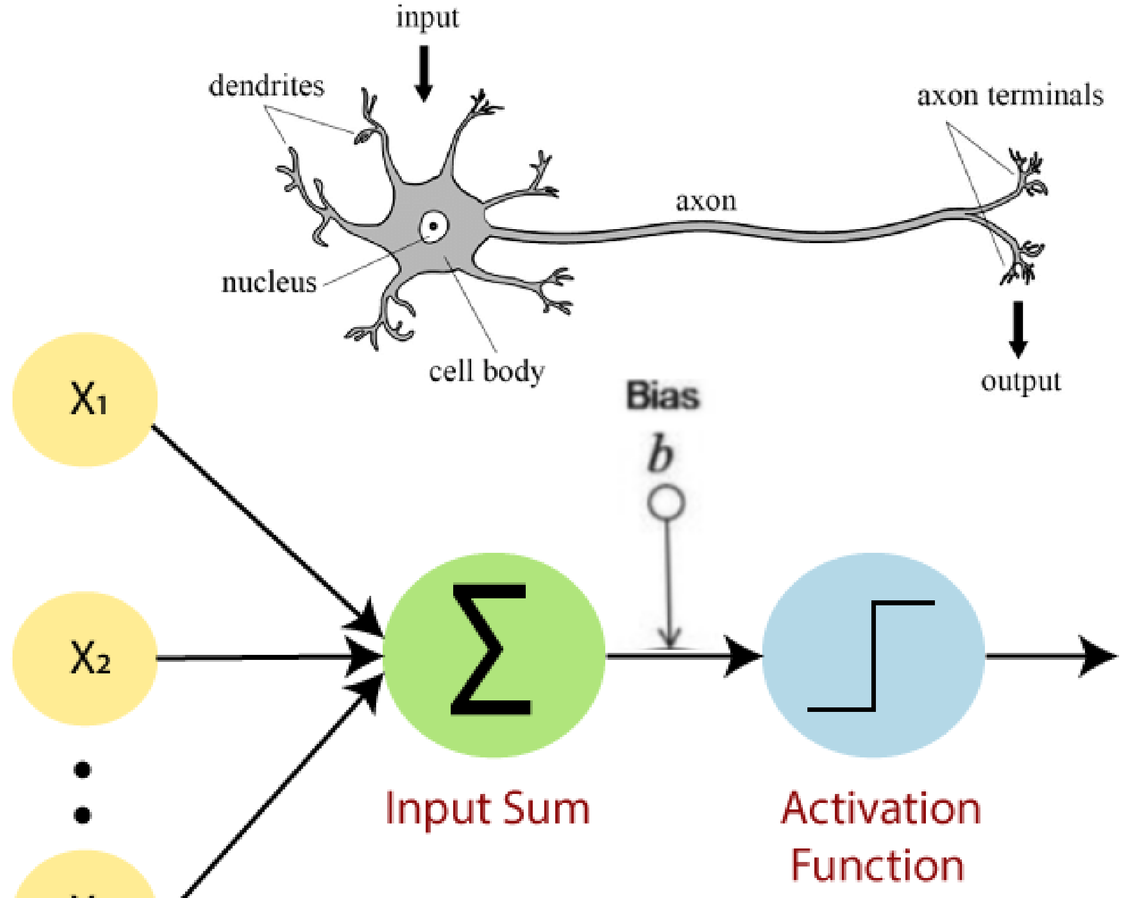
TYPES OF MACHINE LEARNING.

- **REINFORCEMENT LEARNING**
  - Semi-supervised learning
  - System exists and it takes actions in an environment
  - Environment rewards system if correct else punished
  - Example: Self-driving cars (speed limit, collisions, road zones)

NEURAL NETWORKS.

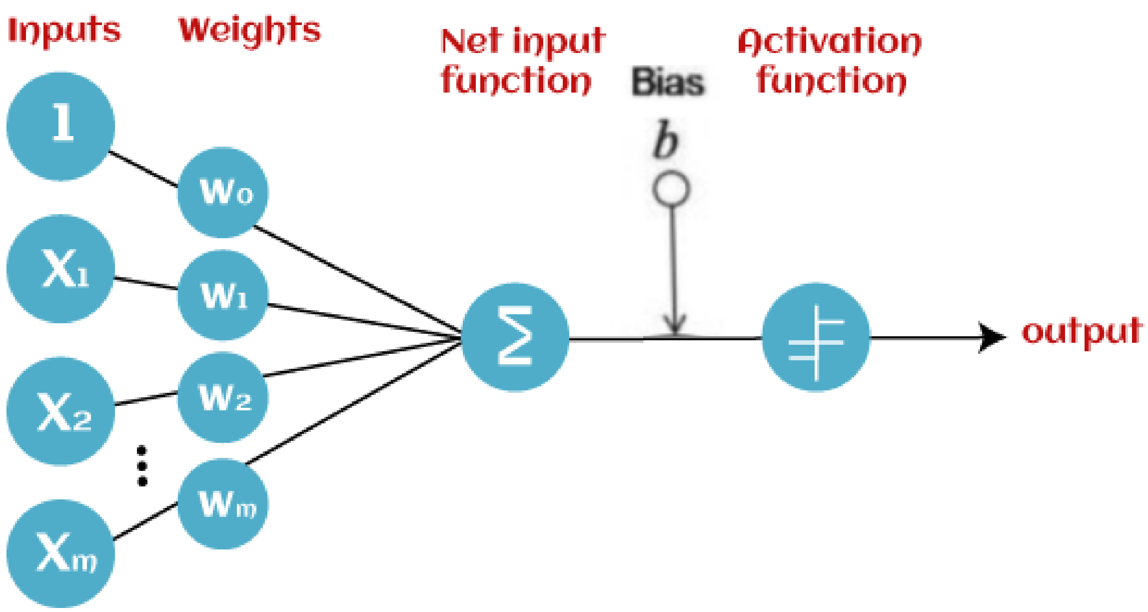


NEURAL NETWORKS.



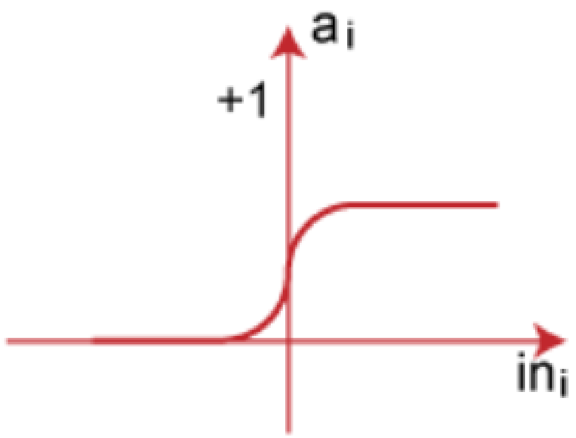
$$Y = \sum (weight * input) + bias$$

Perceptron or Artificial Neuron

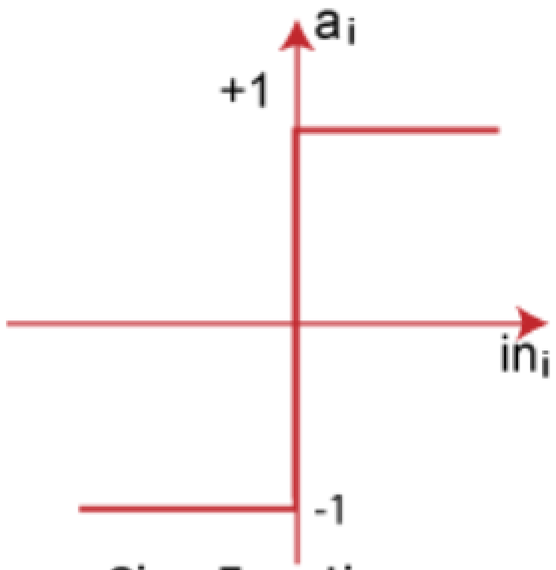


- Building block of an Artificial Neural Network.
- **Mr. Frank Rosenblatt:** Perceptron's can learn anything that it can be programmed to do.

NEURAL NETWORKS: ACTIVATION FUNCTIONS

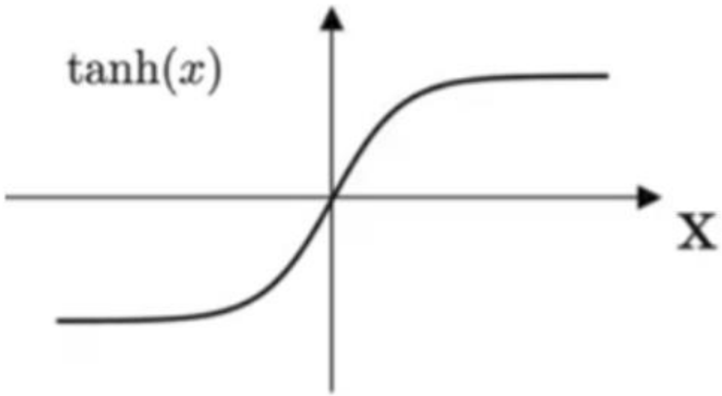


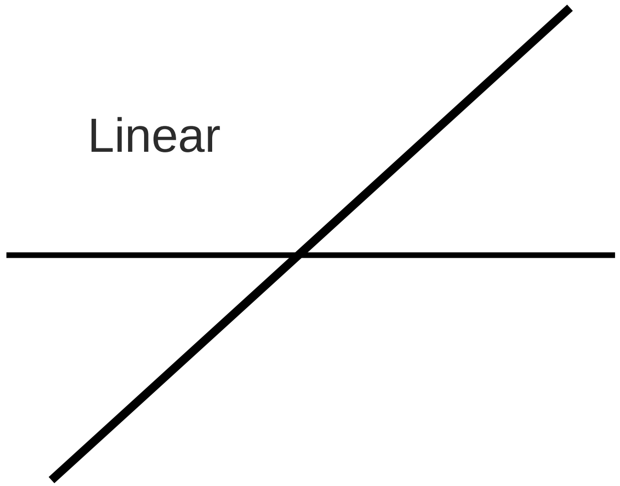
Sigmoid Function



Sign Function

Hyper Tangent Function





https://www.javatpoint.com/perceptron-in-machine-learning

NEURAL NETWORKS/ ARTIFICIAL NEURAL NETWORKS.

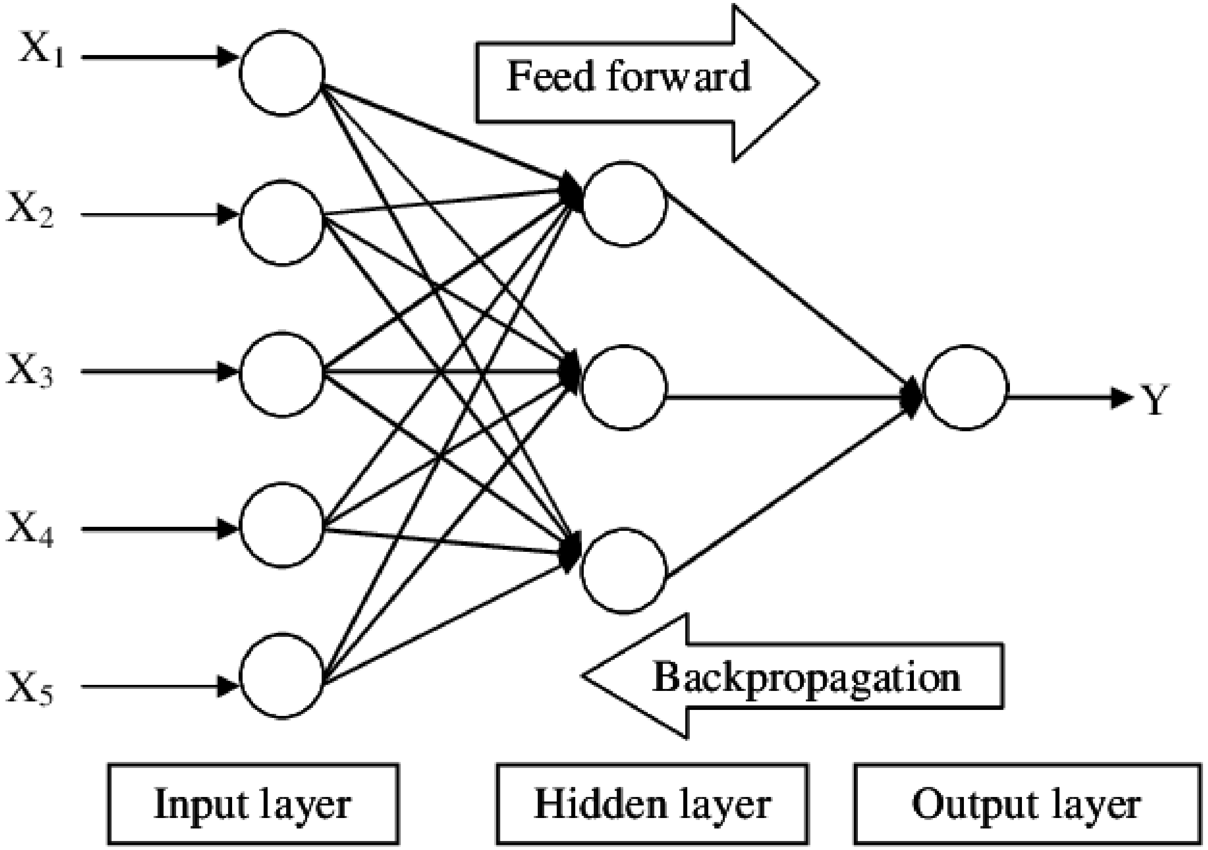
- Dr. **Robert Hecht-Nielsen**: Inventor of one of the first neurocomputers,.
- Defines a neural network as: "...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.“
- **Neural network is a connection of inputs and output units or nodes and each connection has a weight attached to it.**
- Basic computational unit of the brain is a **neuron.**
- Basic unit of computation in a **neural network** is the **neuron**, often called a **node or unit**
  - It receives **input from some other nodes**, or from an **external source** and **computes an output**.
  - **Each input** has an associated **weight (w)**, which is assigned on the basis of its **relative importance** to other inputs.
  - The node applies a function to the weighted sum of its inputs**.**
- **Activation function: Defines the output** of that node **given an input or set of inputs**.

NEURAL NETWORKS.

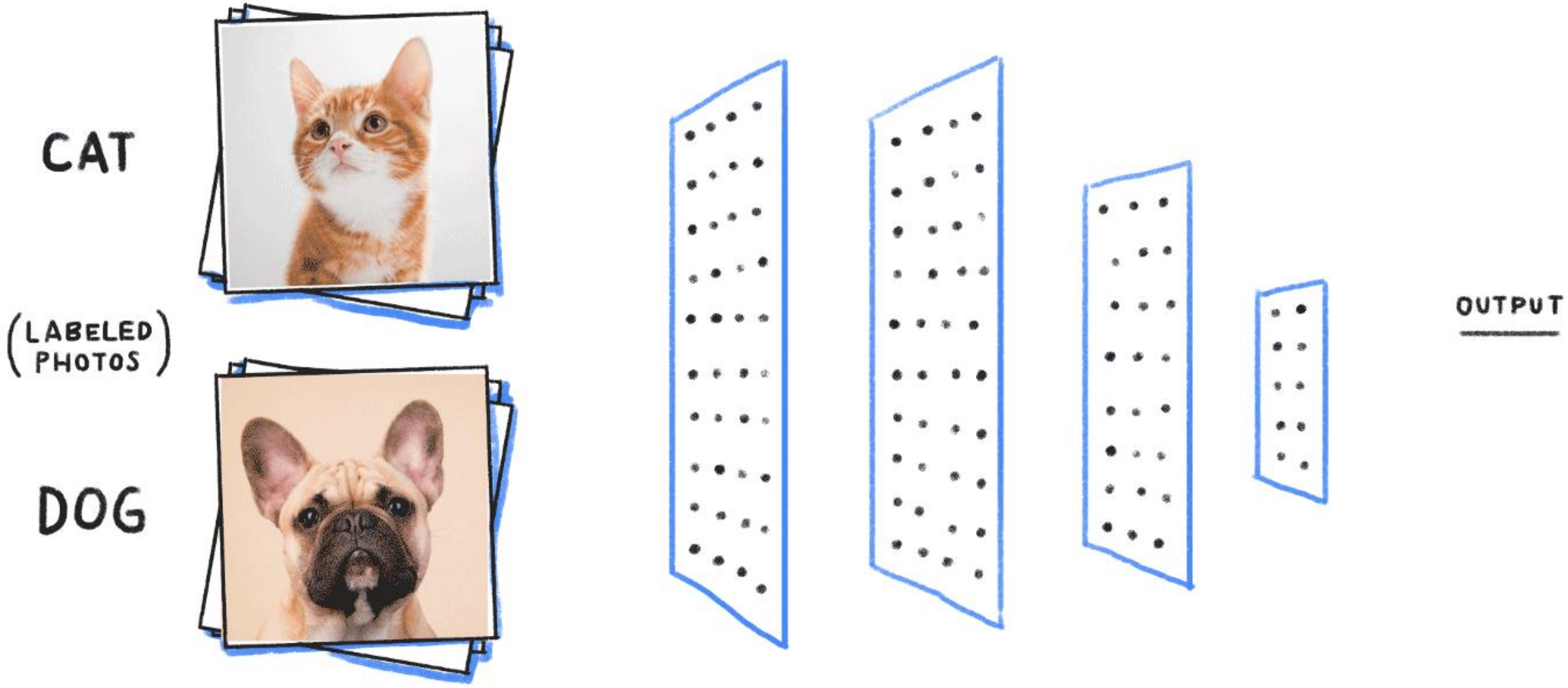
- **Learning rule:** The learning rule is a **rule or an algorithm** which ***modifies the parameters*** of the neural network, for a given input to the network to **produce a favoured output**. This learning process typically amounts to modifying the weights and thresholds. *LEARNING PHASE.*
- **Have two operations**
  1. **FEED FORWARD** network: The network is **feed-forward i**n that none of the weights cycles back to an input unit or to an output unit of a previous layer.
  2. **Back Propagation: Network learns** by iteratively processing a set of training data (samples).

FEED FORWARD NETWORK.

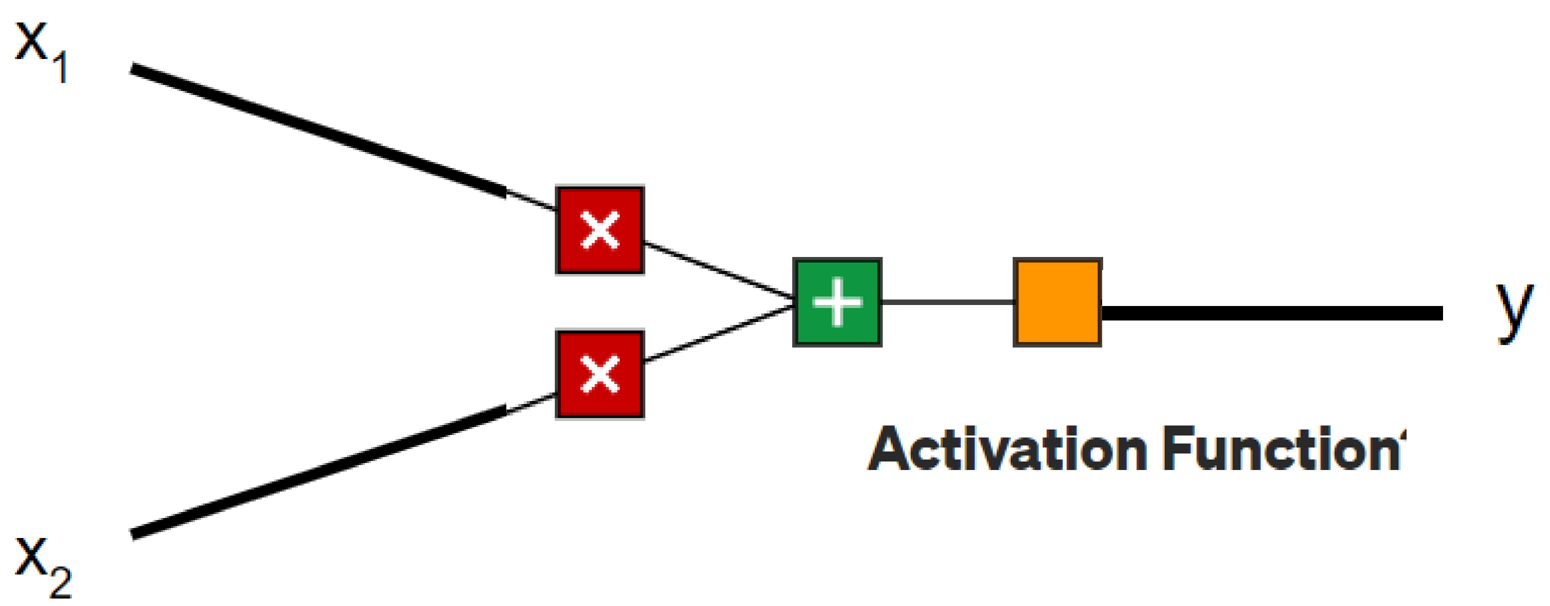
- **Inputs** are **fed simultaneously** into the input layer.
- **Weighted outputs** of input units are fed into **hidden layer.**
- Weighted outputs of the last hidden layer are inputs to units making up the output layer.
- **INPUT:** records without class attribute with normalized attributes values.
- **INPUT VECTOR**: $X = \{ x_1, x_2, \ldots x_n\}$ where **n** is the number of (non class) attributes.
  - **INPUT LAYER:** As many nodes as non-class attributes i.e. as the length of the input vector.
- **HIDDEN LAYER**: Number of nodes in the hidden layer and the number of hidden layers depends on implementation.
- **OUTPUT LAYER:** Corresponds to the class attribute.



NEURAL NETWORKS VISUALISED.



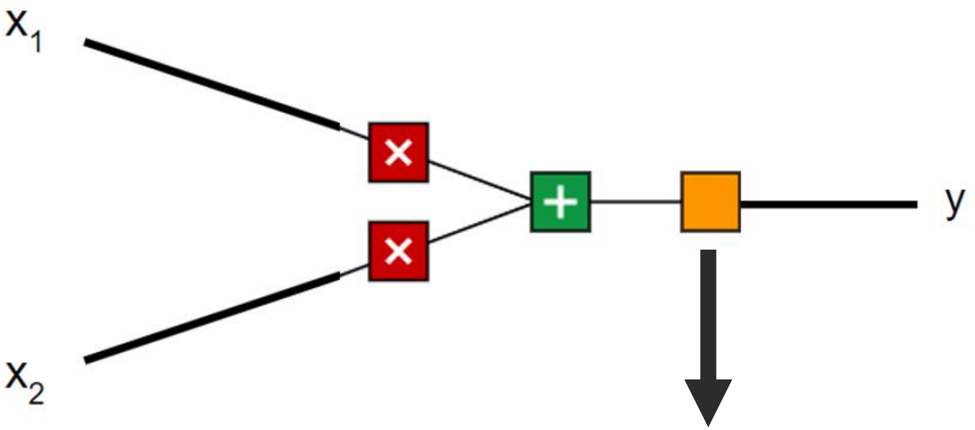
SIMPLE EXAMPLE 1: FEED FORWARD



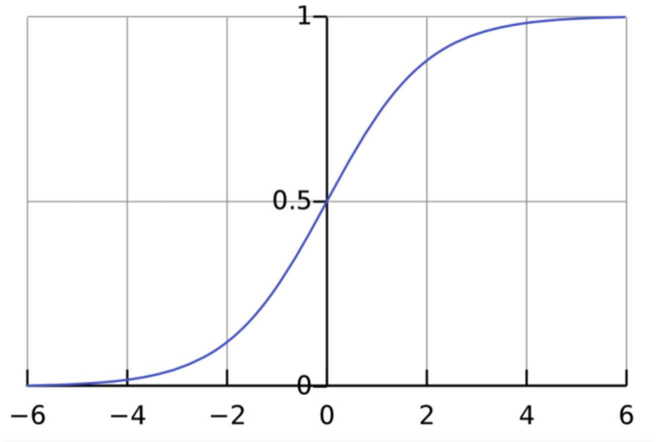
Steps:

1. Inputs are **multiplied** by weights
2. All the weighted inputs are added together with a **bias**
3. Sum is passed through an **activation function**

SIMPLE EXAMPLE 1: FEED FORWARD



The sigmoid function only outputs numbers in the range (0, 1)(0,1).



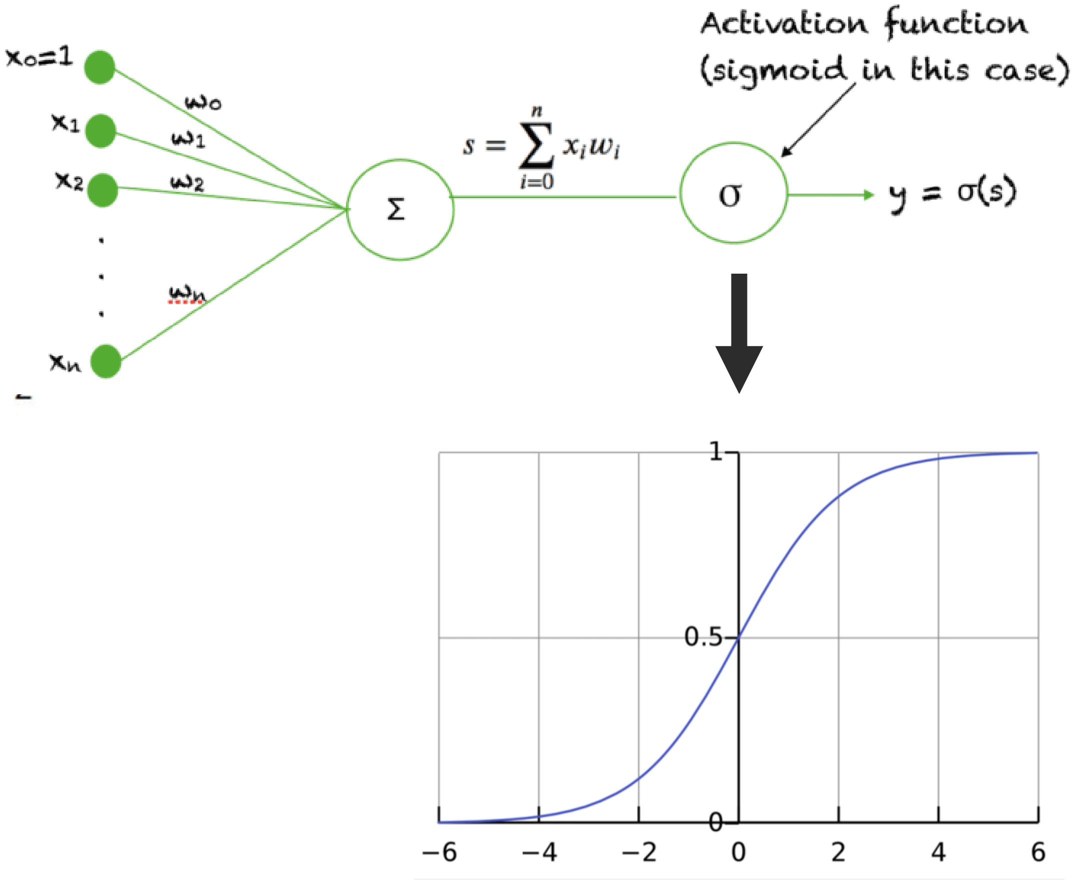
$$x_1 \rightarrow x_1 * w_1$$

$$x_2 \rightarrow x_2 * w_2$$

$$(x_1 * w_1) + (x_2 * w_2) + b$$

$$y = f(x_1 * w_1 + x_2 * w_2 + b)$$

SIMPLE EXAMPLE 1: FEED FORWARD



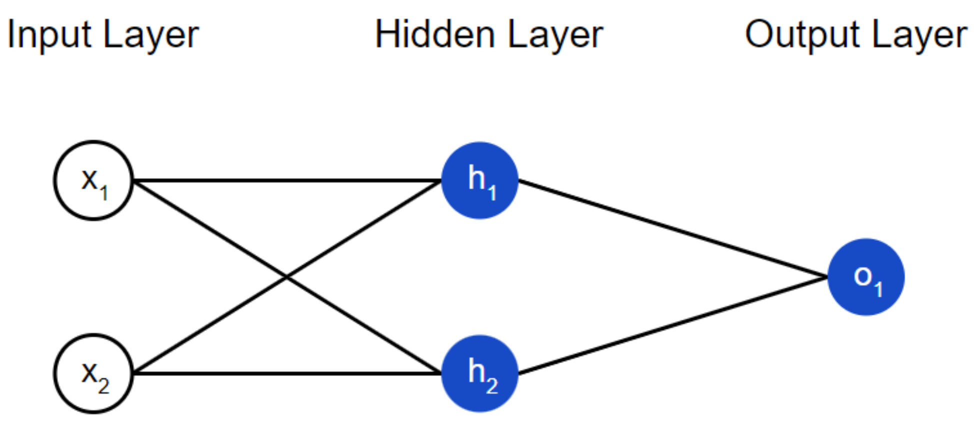
- We have a 2-input neuron that uses a sigmoid activation function with following parameters:

  w = [0,1] and b = 4
- w = [0,1] equivalent to writing w_1=0 and w_2=1 in vector form
- Give neuron input of x = [2,3]
- (w . x) + b = $((w_1 * x_1 + w_2 * x_2) + b)$

  = (0∗2+1∗3+4)

  = 7
- Now, y = f (w . x + b) = f (7) = **0.999**
- The neuron outputs 0.9990 given the inputs x = [2, 3] and w=[0,1]
- This process of passing inputs forward to get an output is known as **feedforward**.

SIMPLE EXAMPLE 2: FEED FORWARD



- Network has 2 inputs , one hidden layer with 2 neurons (h_1 and h_2) and one output layer with one neuron (o_1)
- Inputs for o_1 are from h_1 and h_2 and that's what makes this a network.

- Here we have, w = [0,1] and b = 0 and sigmoid activation function.
- Give input x = [2,3]
- h1 = h2 = f(w.x + b)

  $= f((w_1 * x_1 + w_2 * x_2) + b)$

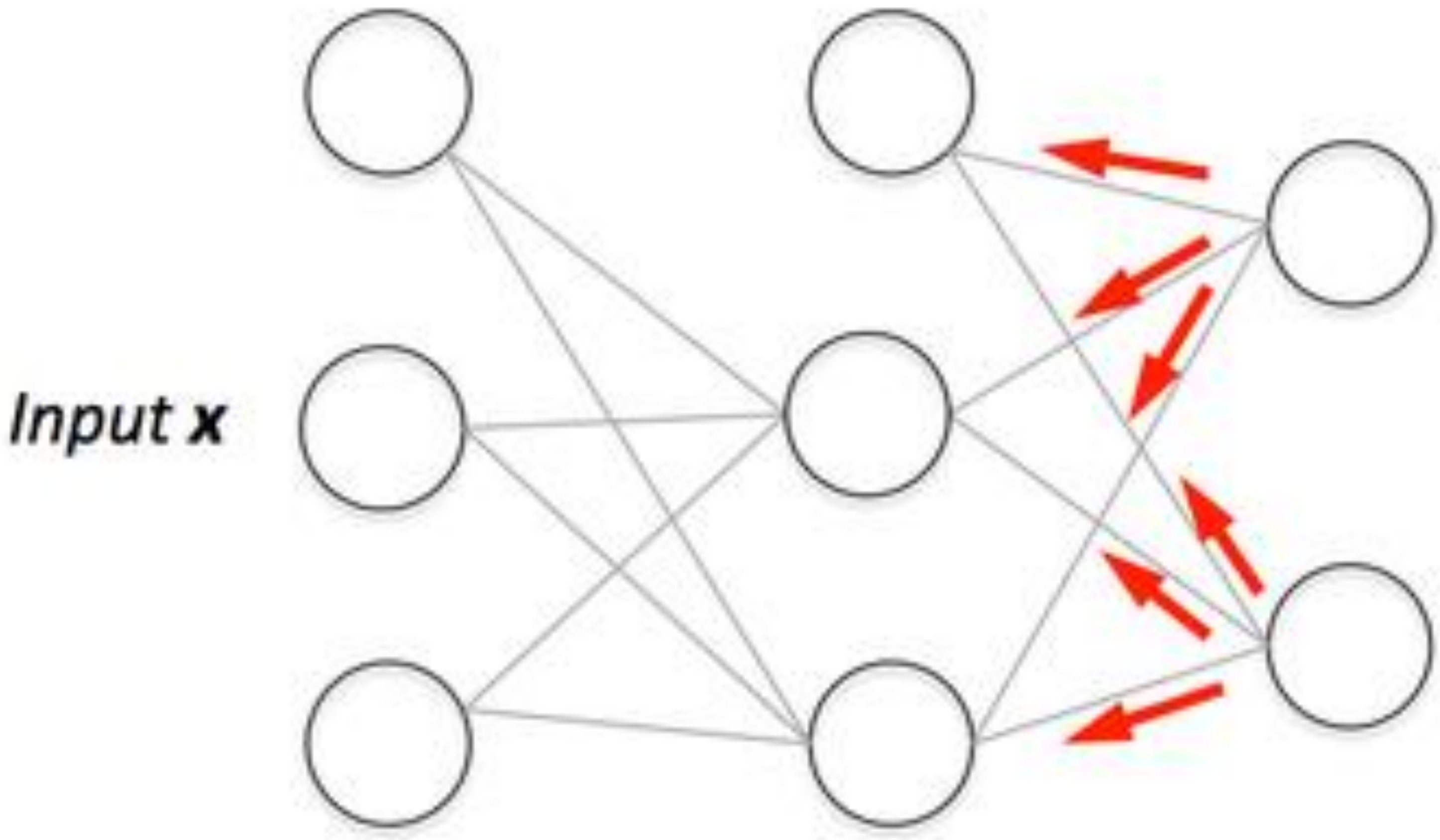
  $= f(0*2+1*3+0)$

  = f(3) = **0.9526**
- Now, o_1 = f (w.[h1, h2] + b)

  = f ((0* h1) + (1*h2) + 0)

  = f(0.9526) = **0.7216**
- The outputs 0.7216 given the inputs x = [2, 3]

BACK PROPOGATION.



$$w_{ij} = w_{ij} + l \times Error_j \times Output_i$$

$$Error_i = Output_i(1 - Output_i)\sum_{j=1}^{n} Error_j w_{ij}$$

$$Error_i = Output_i(1 - Output_i)(Actual_i - Output_i)$$

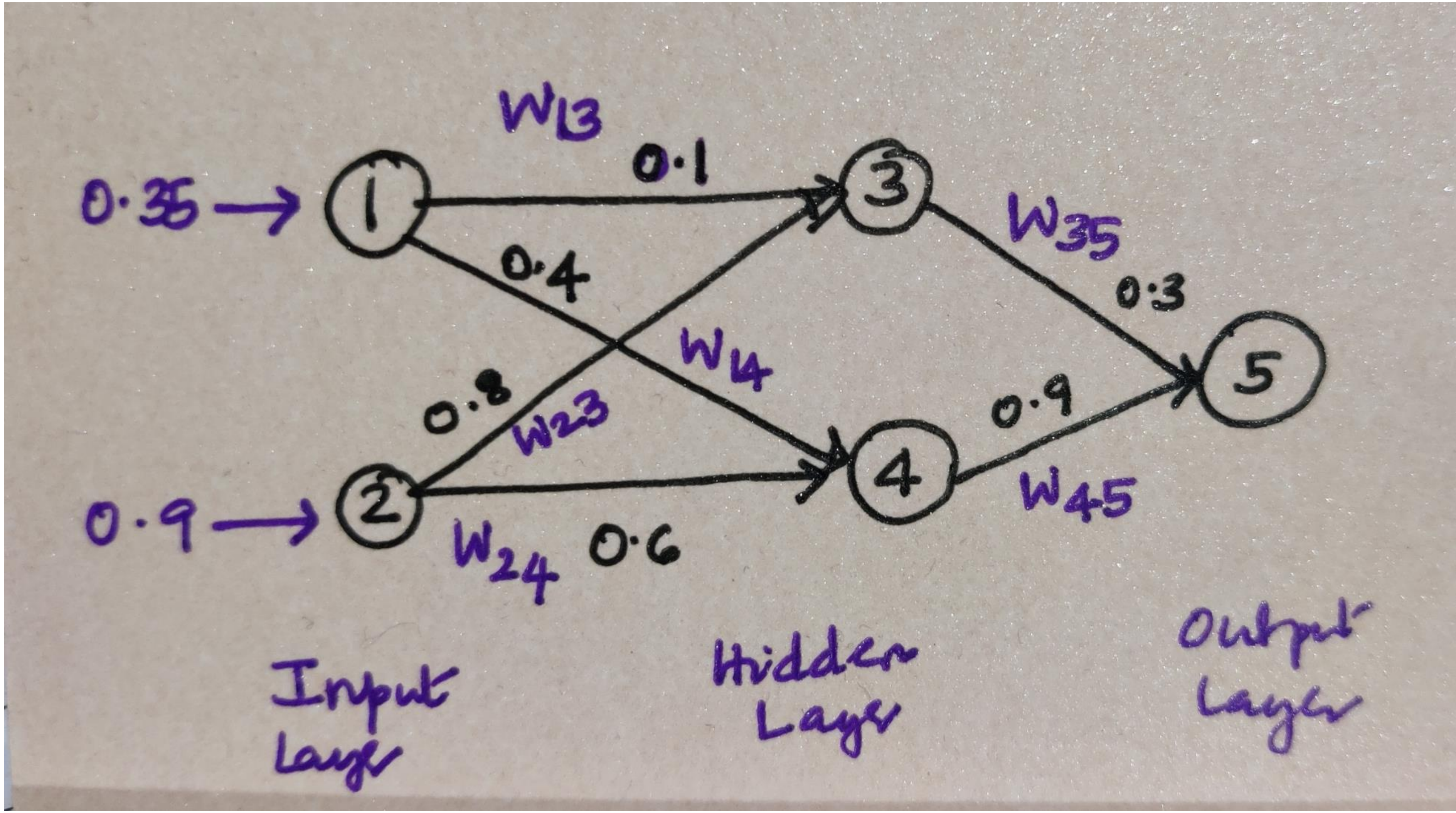
Sigmoid : $$Output = \frac{1}{1 + e^{-Input}}$$

Tanh : $$Output = \frac{e^{Input} - e^{-Input}}{e^{Input} + e^{-Input}}$$

$$Input = \sum_{j=1}^{n} I_j w_j$$

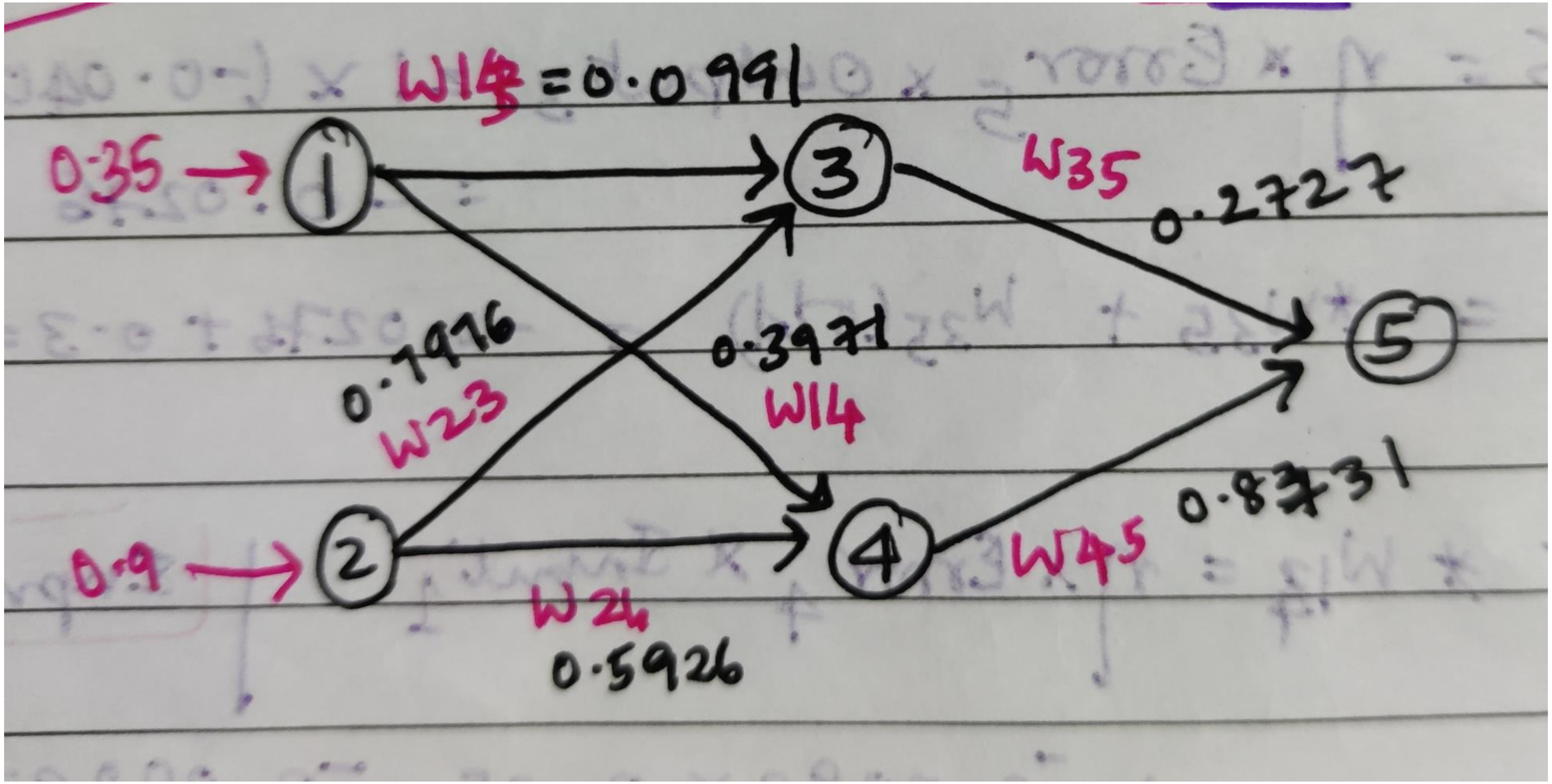
BACK PROPOGATION.

- LOOK AT the BOARD for solving example.
- Learning rate:1
- Actual output=0.5



BACK PROPOGATION.

- Updated weights for next feedforward.



TAKE A BREAK.