تقني
معلومة مفيدة .. وقت قليل

Artificial Neural Networks (ANNs)

الشبكات العصبية الاصطناعية

Eng. Mustafa Othman
Data Scientist & Analyst

Today's Outline:

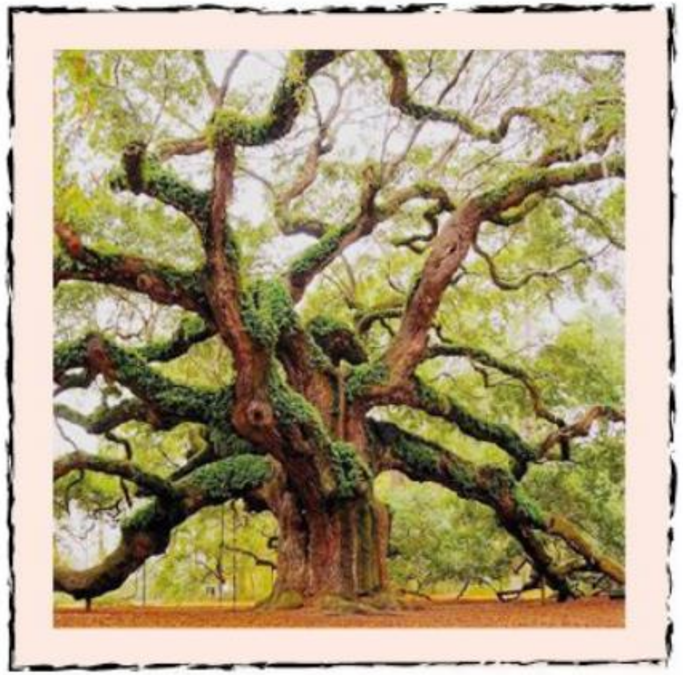
- **How do Computers "Learn"?**
  - Simple Predicting Machine
- **How do Humans "Learn"?**
  - ANNs Basics – Illustrated Example
- **Demo: Artificial Neural Network (ANN) in a Nutshell**
- **ANNs Basics**
  - ANNs Basics
  - Building a Neural Network
    - Make it Deep
  - Training Deep Networks
  - Improving Deep Networks

How do Computers "Learn"?

"Computers are able to see, hear and learn. Welcome to the future."
~ Dave Waters

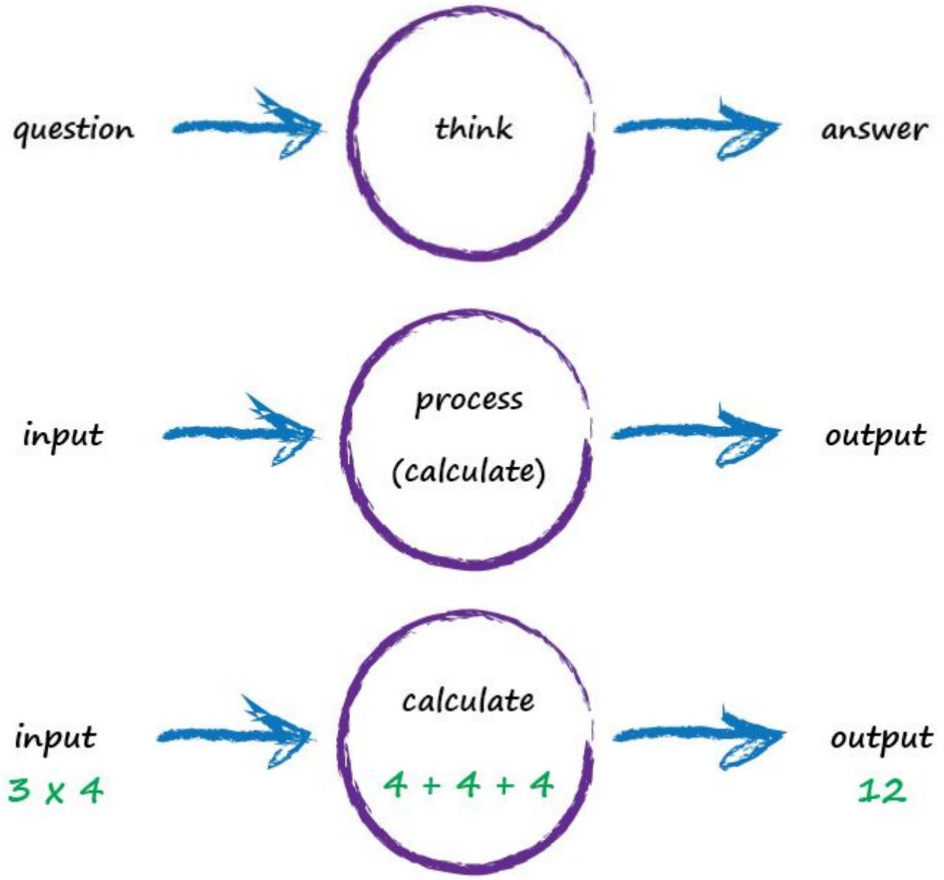
How do Computers “Learn”? (0)
(Humans vs. Computers)

- Easy for Me, Hard for You!



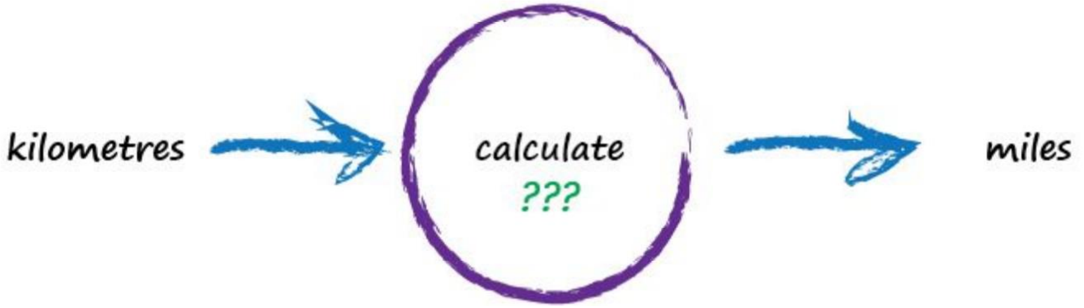
How do Computers "Learn"? (1)
(Super Calculator)

- Imagine a basic machine that takes a **question**, does some "**thinking**" and pushes out an **answer**.
- Computers **don't really think**, they're just **glorified calculators**.
- All useful computer systems have an **input**, and an **output**, with **calculation** in between.
- Neural networks are no different.

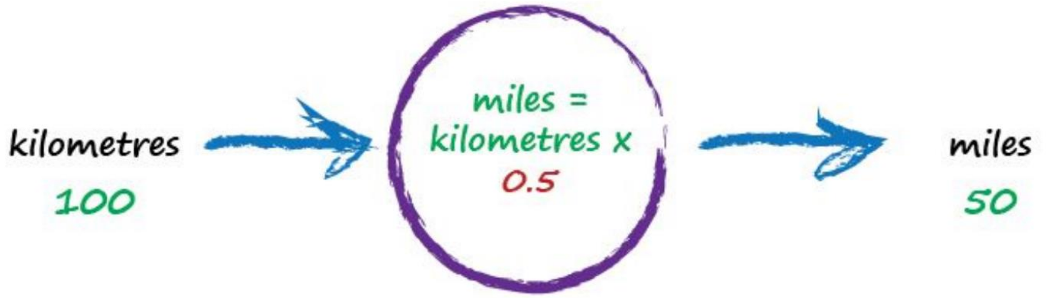


How do Computers "Learn"? (2)
(Simple Predicting Machine)

- Let's ramp up the **complexity** just a tiny notch.
- Imagine a **machine** that converts kilometers to miles

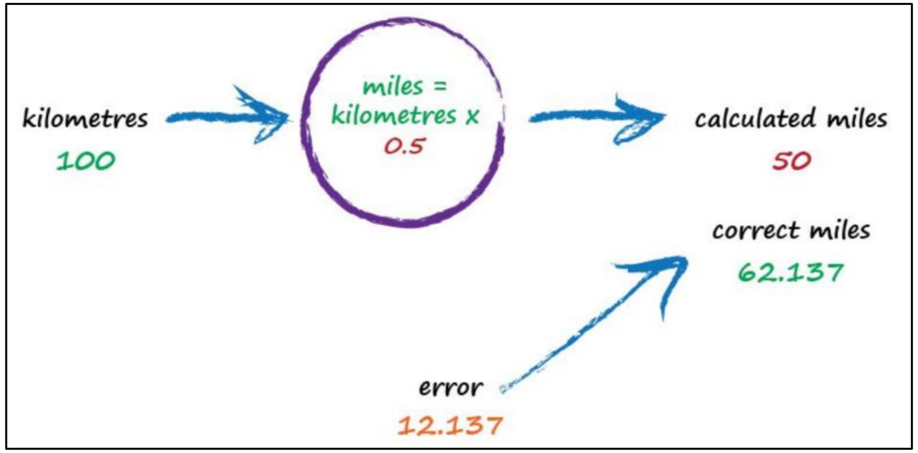


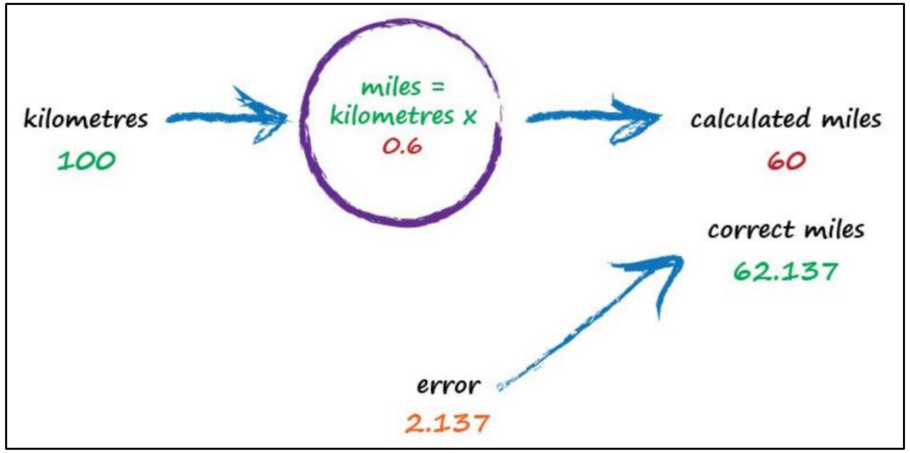
Truth Example	Kilometres	Miles
1	0	0
2	100	62.137

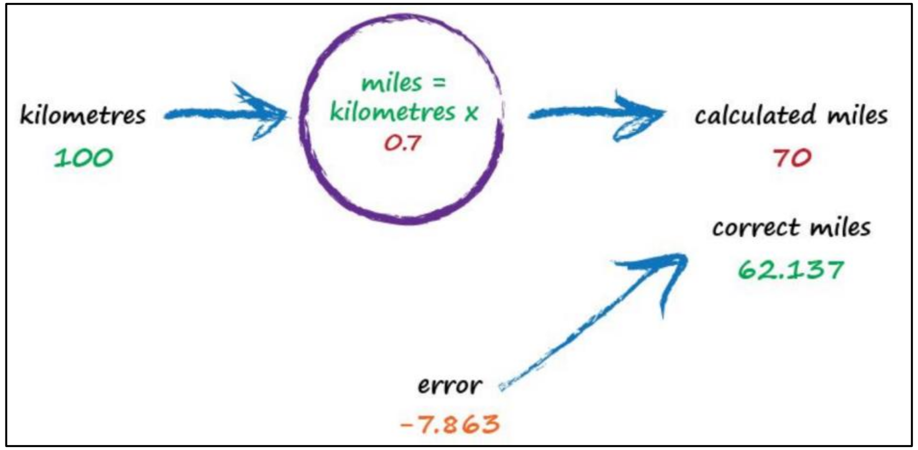


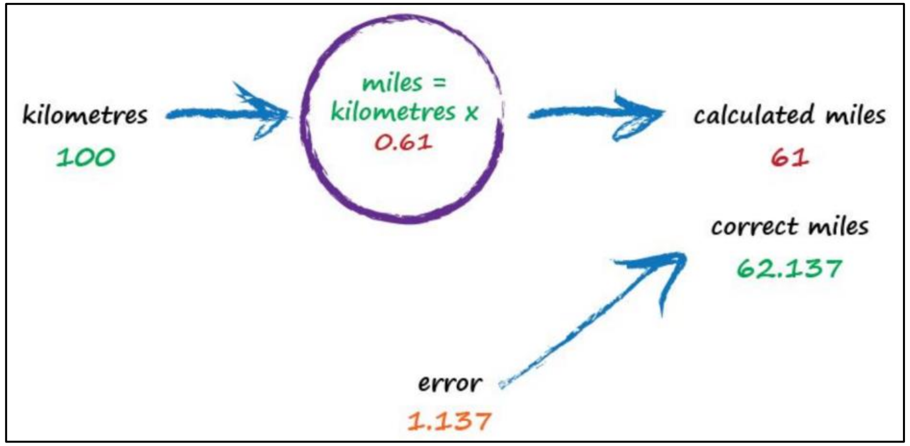

```
error = truth - calculated
      = 62.137 - 50
      = 12.137
```

How do Computers "Learn"? (3)
(Computer / Machine Learning Process)







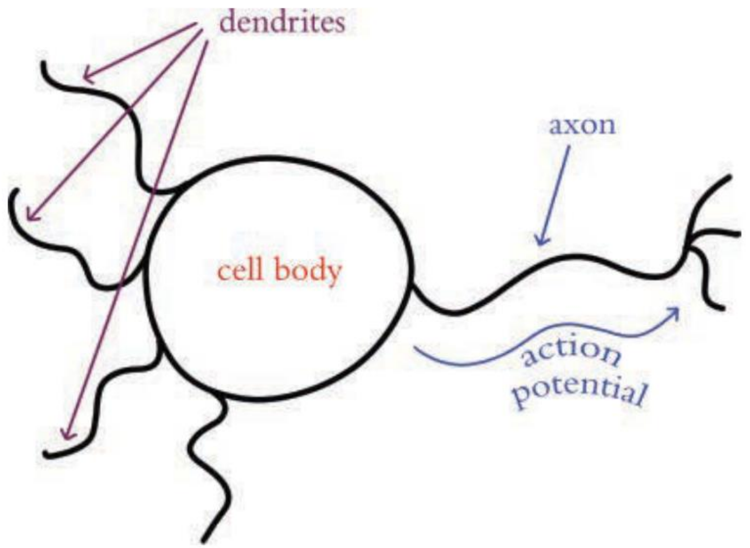


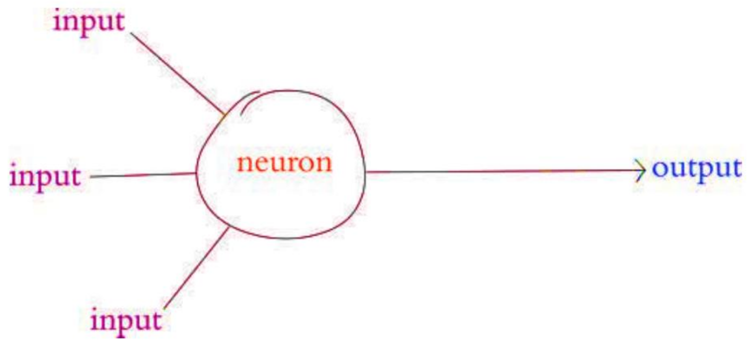
How do Humans “Learn”?

“Learn as if you were not reaching your goal and as though you were scared of missing it”. ~ Confucius

How do Humans “Learn”? (0)
(Biological Neuron)

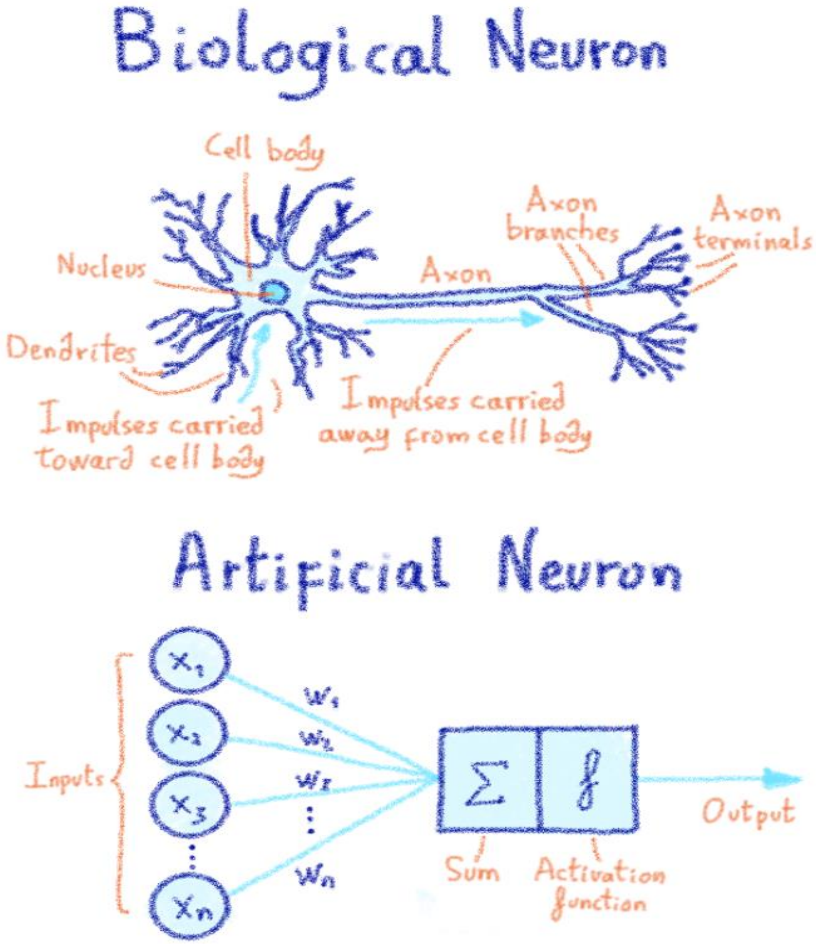
- A given **biological neuron** receives **input** into its **cell body** from many (generally thousands) of **dendrites**, with each dendrite receiving signals of information from another neuron in the nervous system.
- When the **signal** conveyed along a dendrite reaches the cell body, it **causes a small change** in the **voltage** of the cell body (positive or negative).
- The neuron will fire something called an **action potential** away from its cell body, down its **axon**, thereby transmitting a **signal** to other neurons in the network.



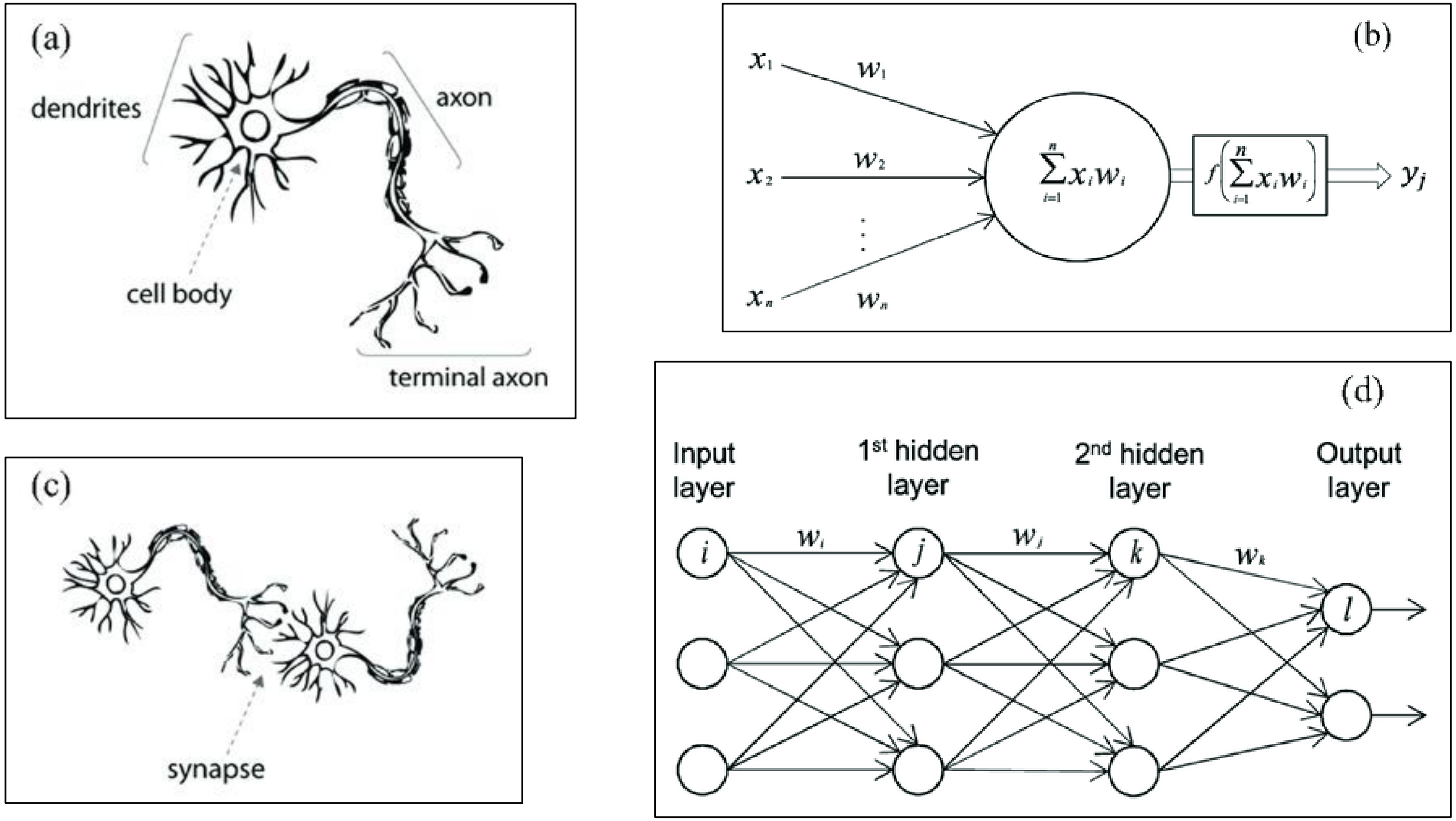


How do Humans "Learn"? (1)
(The Perceptron "The Artificial Neuron")

- **The Artificial Neuron** (Single-Layer Perceptron)
  - https://youtu.be/cNxadbrN_aI

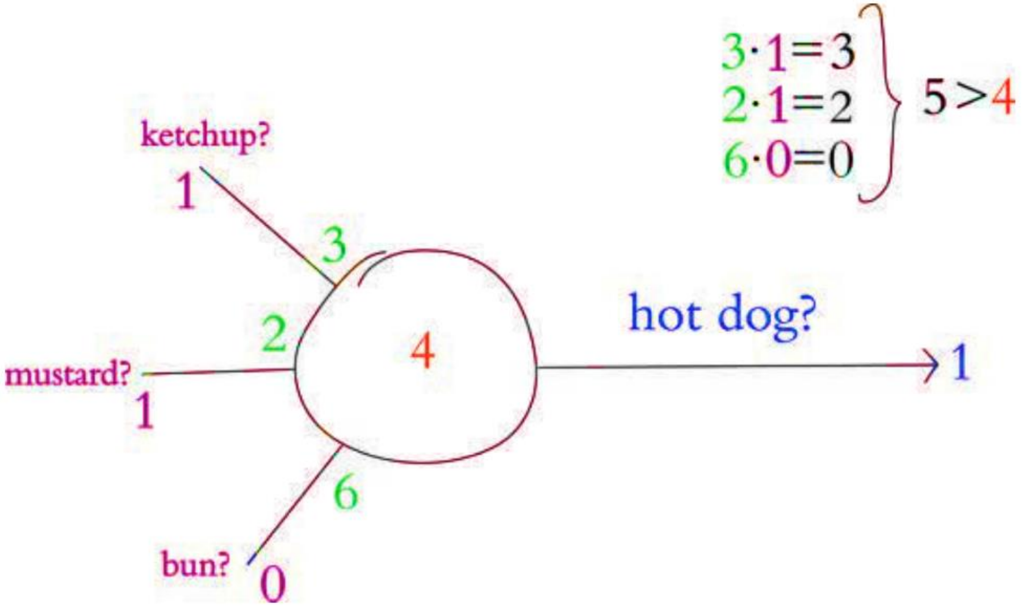


How do Humans "Learn"? (2)
(BNN vs. ANN)



How do People "Learn"? (4)
(An Illustrated Example of the Perceptron)

- We're going to look at a **perceptron** that is specialized in distinguishing whether a given object is a **hot dog** or, well . . . **not a hot dog**.



$$\sum_{i=1}^{n} w_i x_i$$

$$\sum_{i=1}^{n} w_i x_i \begin{cases} > \textit{threshold}, \text{ output } 1 \\ \leqslant \textit{threshold}, \text{ output } 0 \end{cases}$$

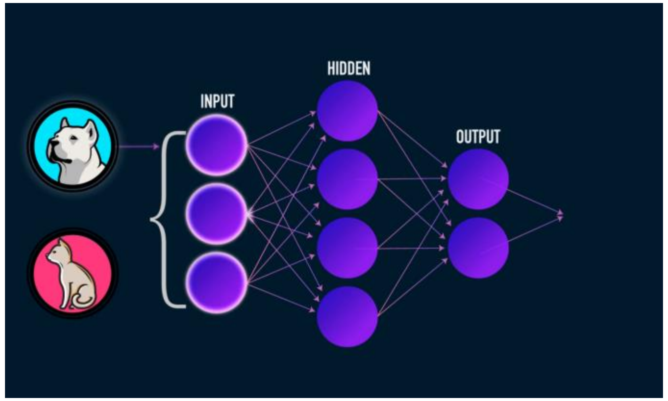
$$\text{output} \begin{cases} 1 \text{ if } w \cdot x + b > 0 \\ 0 \text{ otherwise} \end{cases}$$

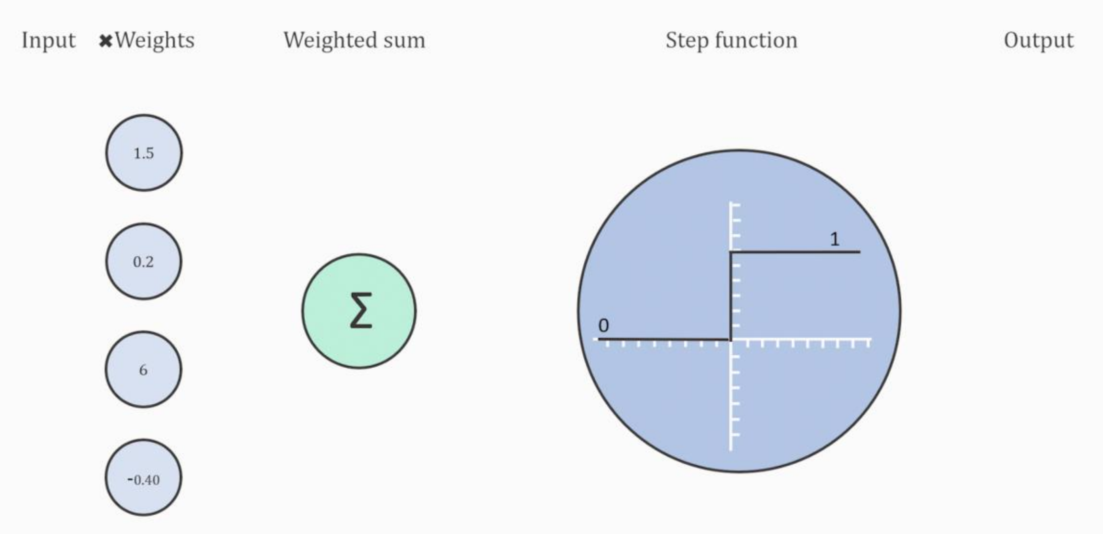
Artificial Neural Networks (ANNs) Basics

"We are all now connected by the Internet, like neurons in a giant brain." ~ Stephen Hawking

ANNs Basics
(Multi-Layer Perceptron)

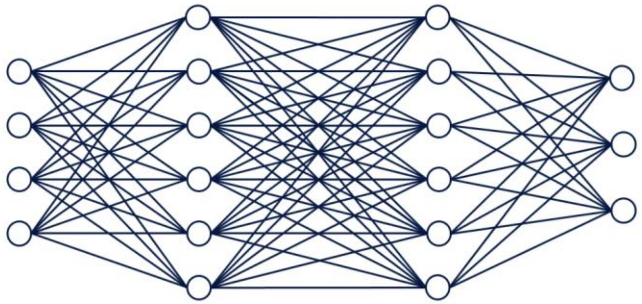
- **Artificial neural networks (ANNs)**, usually simply called **neural networks** are computing systems inspired by the **biological neural networks** that constitute animal brains.
- Artificial neural networks (ANNs) are comprised of a node layers, containing an **input** layer, one or more **hidden** layers, and an **output** layer.
- Each node, or artificial neuron, connects to another and has an associated **weight** and **threshold**.
- If the output of any individual node is **above** the specified **threshold** value, that node is **activated**, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

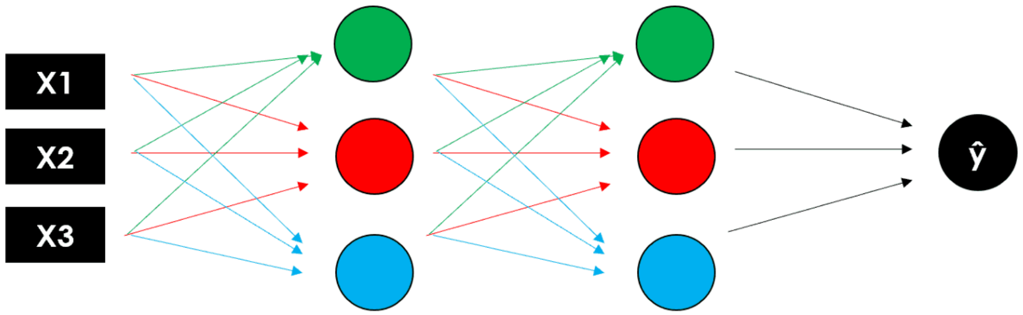




Building a Neural Network (0)
(Input / Hidden / Output Layers)

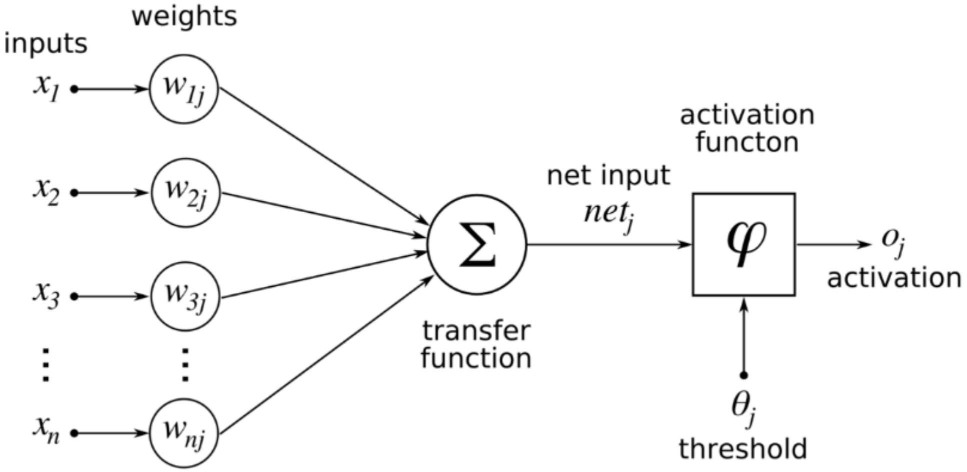
- **Input Layer(s):**
  - Neurons in the **input layer** don't perform any calculations. This is essential because the use of ANNs involves performing computations on matrices that have **predefined dimensions**.
- **Hidden Layer(s):**
  - There are many kinds of **hidden layers**, but the most general type is the **dense layer**, which can also be called a **fully connected layer** and can be found in many deep learning architectures.
- **Output Layer(s)**

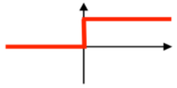
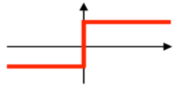
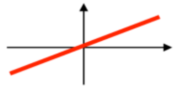
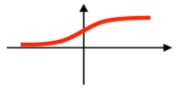
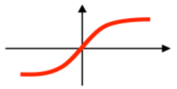
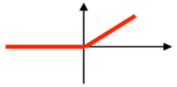
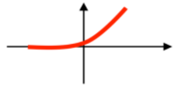




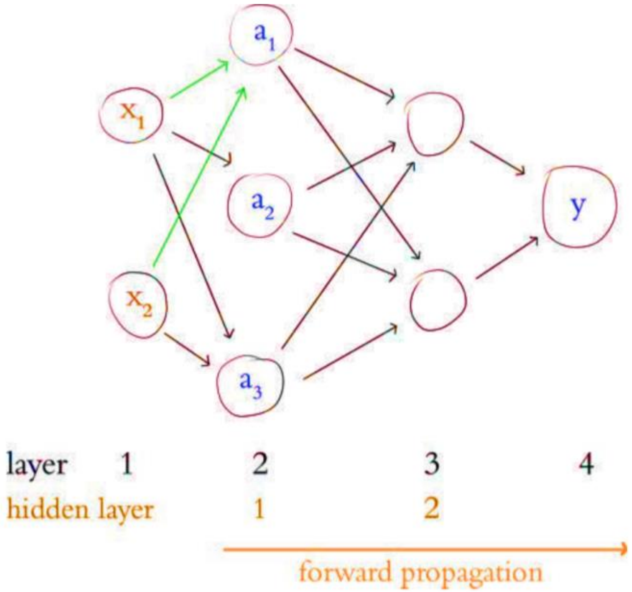
Building a Neural Network (1)
(The Activation Function)

- An **activation function** in a neural network defines how the weighted sum of the **input** is transformed into an **output** from a node or nodes in a layer of the network.



Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	

Building a Neural Network (2)
(Feedforward Propagation Example)



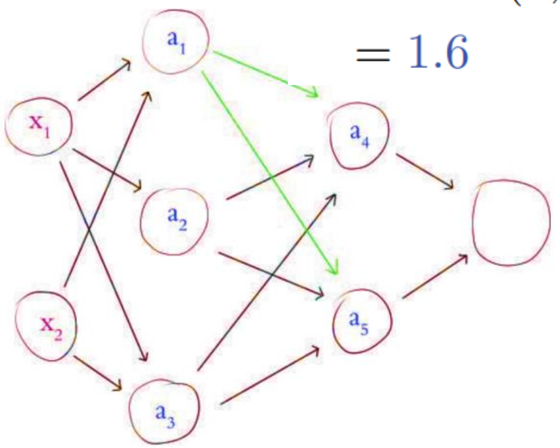
$$z = w \cdot x + b$$

$$z = (w_1 x_1 + w_2 x_2) + b$$

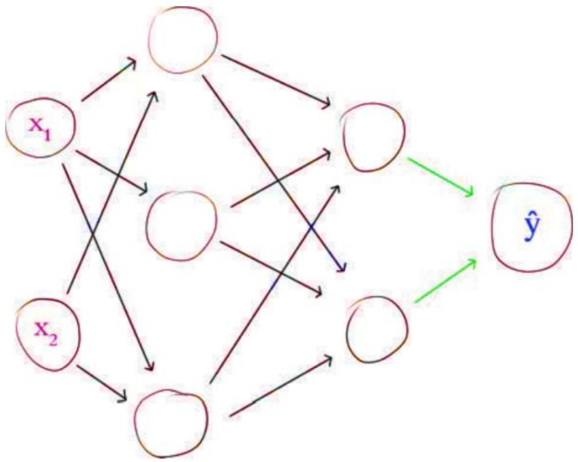
$$a = max(0, z)$$

$$\begin{aligned} z &= w \cdot x + b \\ &= w_1 x_1 + w_1 x_2 + b \\ &= -0.5 \times 4.0 + 1.5 \times 3.0 - 0.9 \\ &= -2 + 4.5 - 0.9 \\ &= 1.6 \end{aligned}$$

$$\begin{aligned} a &= max(0, z) \\ &= max(0, 1.6) \\ &= 1.6 \end{aligned}$$



$$\begin{aligned} z &= w \cdot x + b \\ &= w_1 x_1 + w_2 x_2 + b \\ &= 1.0 \times 2.5 + 0.5 \times 2.0 - 5.5 \\ &= 3.5 - 5.5 \\ &= -2.0 \end{aligned}$$



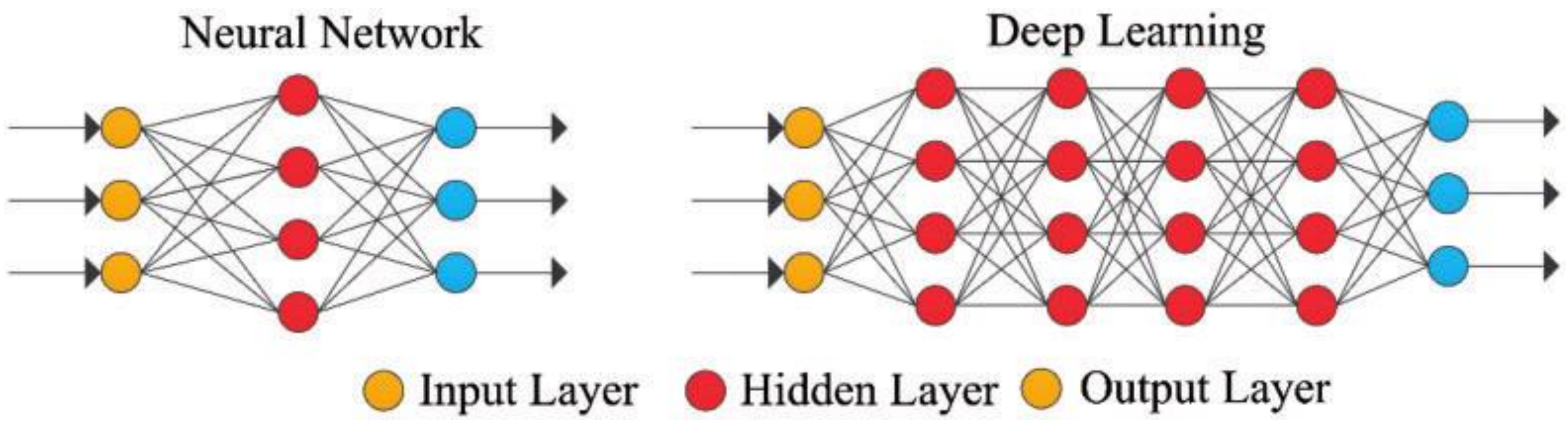
Building a Neural Network (3)
(Make it Deep "Deep Neural Network")

Neural Network

- A neural network is a model of neurons inspired by the human brain. It is made up of many neurons that at inter-connected with each other.
- It generally takes **less time** to train them. They have **lower** accuracy than deep learning systems.

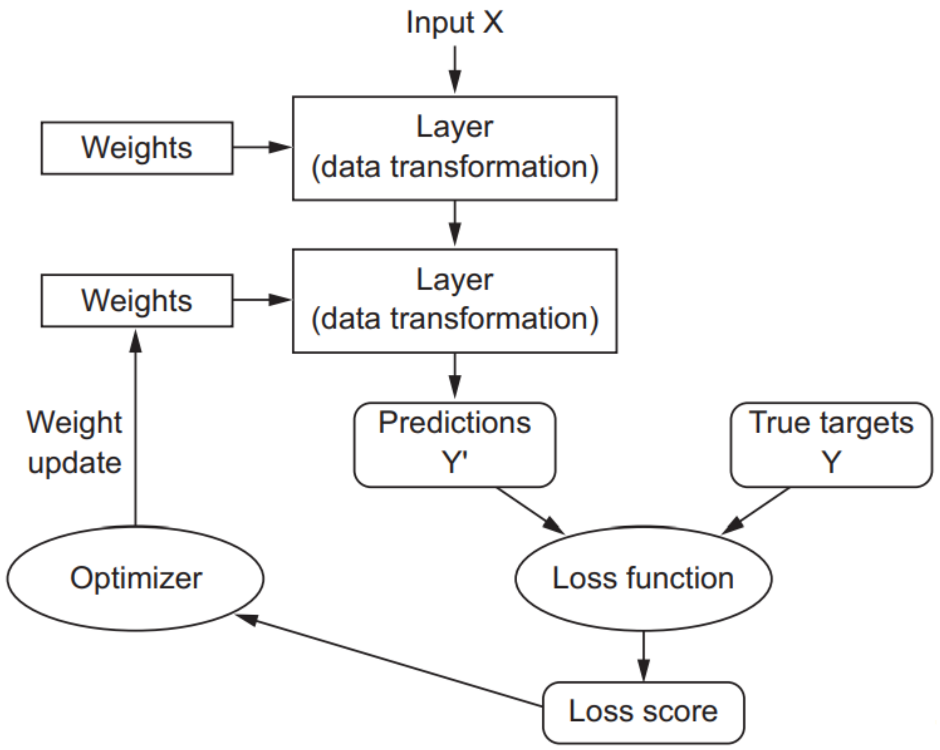
Deep Neural Network

- Deep learning neural networks are distinguished from neural networks based on their depth or number of hidden layers.
- It generally takes **more time** to train them. They have **higher** accuracy than neural networks.



Training a Neural Network (0)
(How does Deep Learning work?)

- **Loss Function** (Objective Function):
  - The **loss function** in a neural network quantifies the difference between the **expected outcome** and the **predicted outcome** produced by the machine learning model.
  - The quantity that will be **minimized** during training. It represents a measure of success for the task at hand.
- **Optimizer:**
  - **Optimizers** are **algorithms** or methods used to change the attributes of the neural network such as **weights, batch size** and **learning rate** **(optimization hyperparameters)** to reduce the losses.
  - Determines how the network **will be updated** based on the loss function.
  - This **weight updating** process is known as **backpropagation**.



Training a Neural Network (1)
(Loss Functions and Optimizers)

Loss Functions:

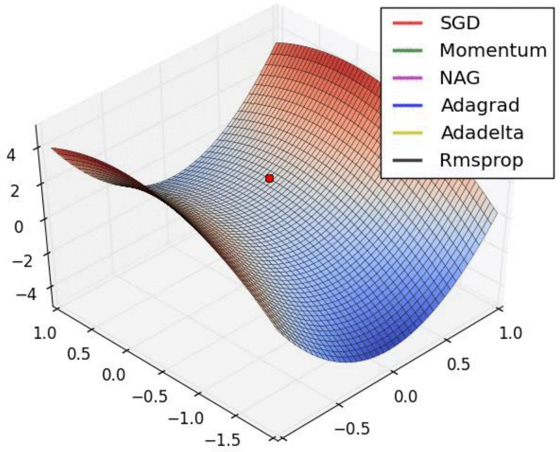
- **Regression Problems:**
  - Mean Squared Error (MSE)
- **Classification Problems:**
  - Cross-Entropy
    - Binary & Multi-classes

$$C = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$C = -\frac{1}{n}\sum_{i=1}^{n}[y_i \ln \hat{y}_i + (1 - y_i)\ln(1 - \hat{y}_i)]$$

Optimizers:

- Gradient Descent (GD)
- Stochastic Gradient Descent (SGD)
- Adaptive Gradient Descent (AdaGD)



Further Readings

- Make Your Own Neural Network, Tariq Rashed
  - Part-I: (pg. 12 – 18)
- Deep Learning Illustrated, Jon Krohn
  - Chapters 6, 7, 8, & 9
- Deep Learning with Python, François Chollet
  - Chapters 1, 2, 3, & 4

THANKS

Keep Moving Forward! ☺

Eng. Mustafa Othman
Data Scientist & Analyst