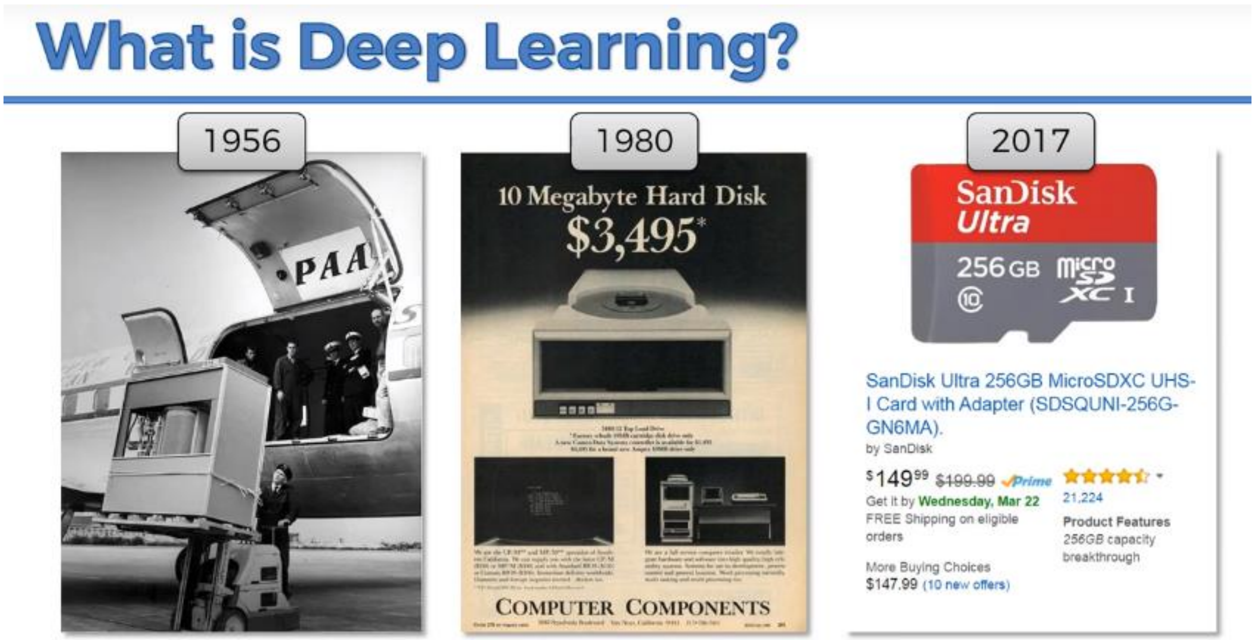
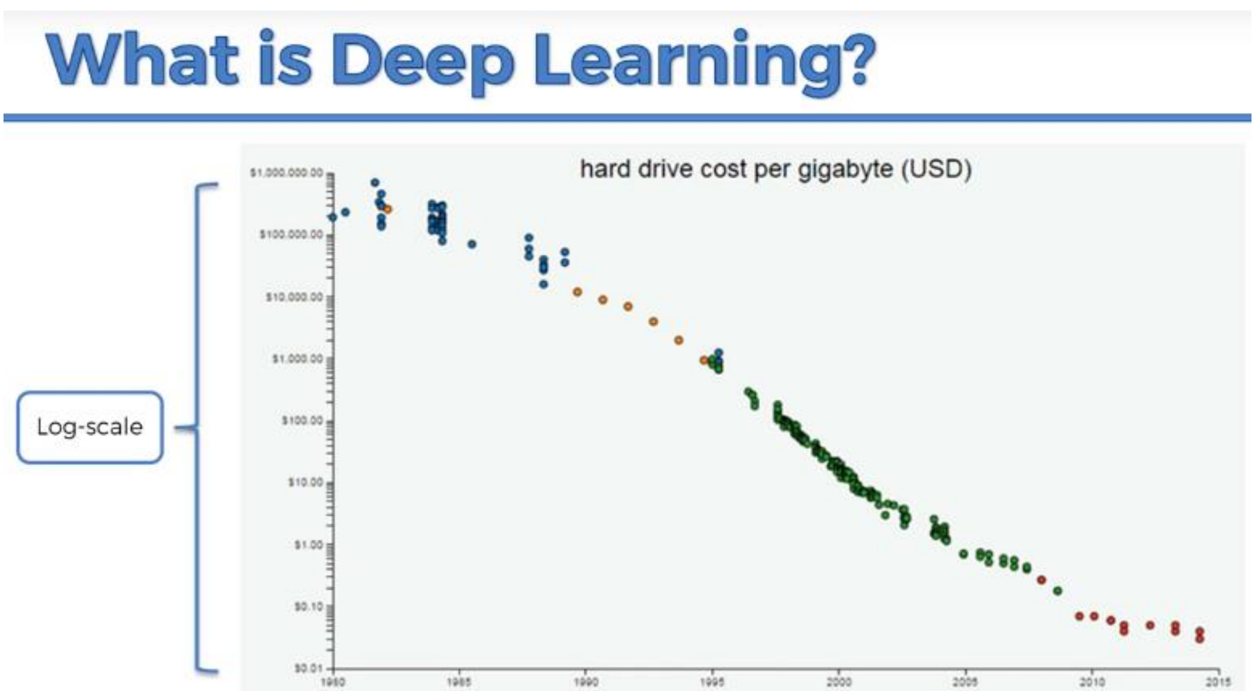
Part 1: Artificial Neural Networks



From 5MB, to 10 MB to 256 GB



What is Deep Learning?

STORAGE LIMITS

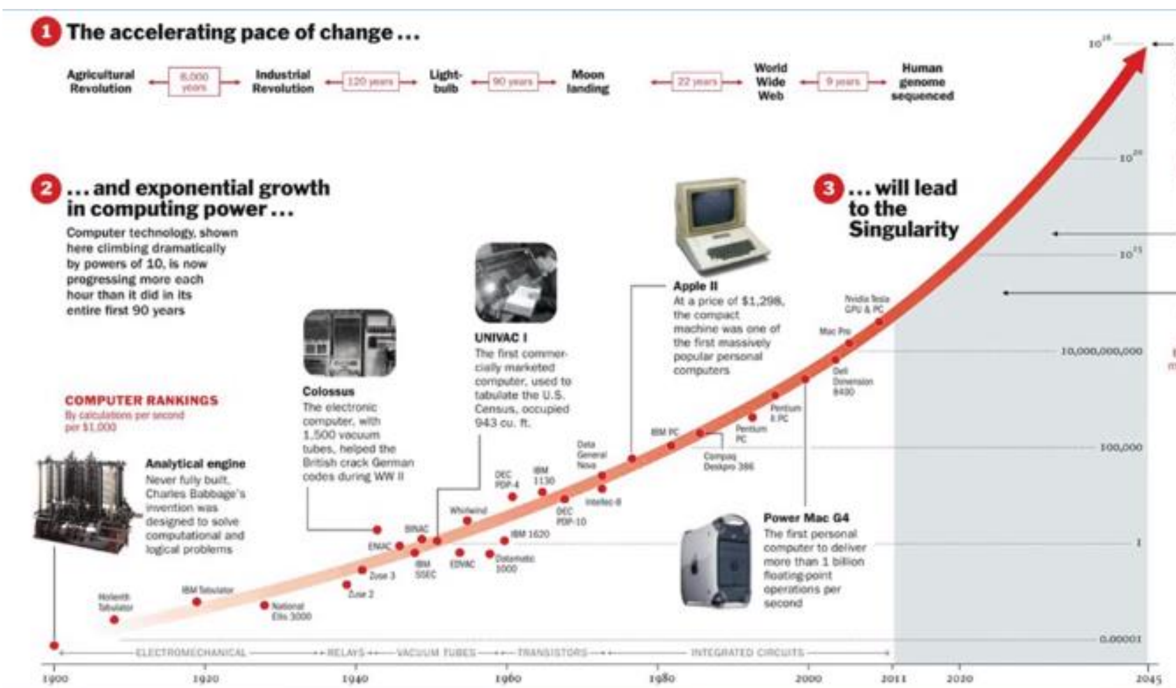
Estimates based on bacterial genetics suggest that digital DNA could one day rival or exceed today's storage technology.

	Hard disk	Flash memory	Bacterial DNA
Read–write speed (µs per bit)	~3,000–5,000	~100	<100
Data retention (years)	>10	>10	>100
Power usage (watts per gigabyte)	~0.04	~0.01–0.04	$<10^{-10}$
Data density (bits per cm³)	$\sim10^{13}$	$\sim10^{16}$	$\sim10^{19}$

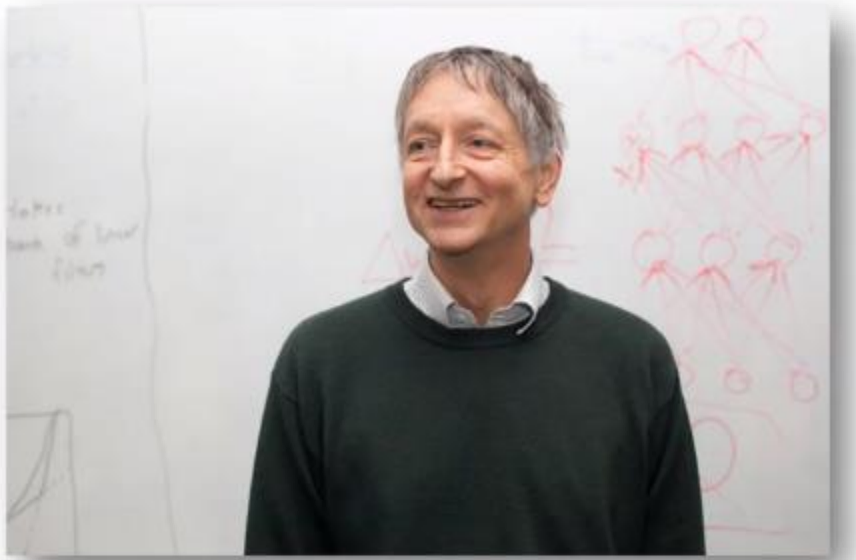
WEIGHT OF DNA NEEDED TO STORE WORLD'S DATA

~1 kg

©nature



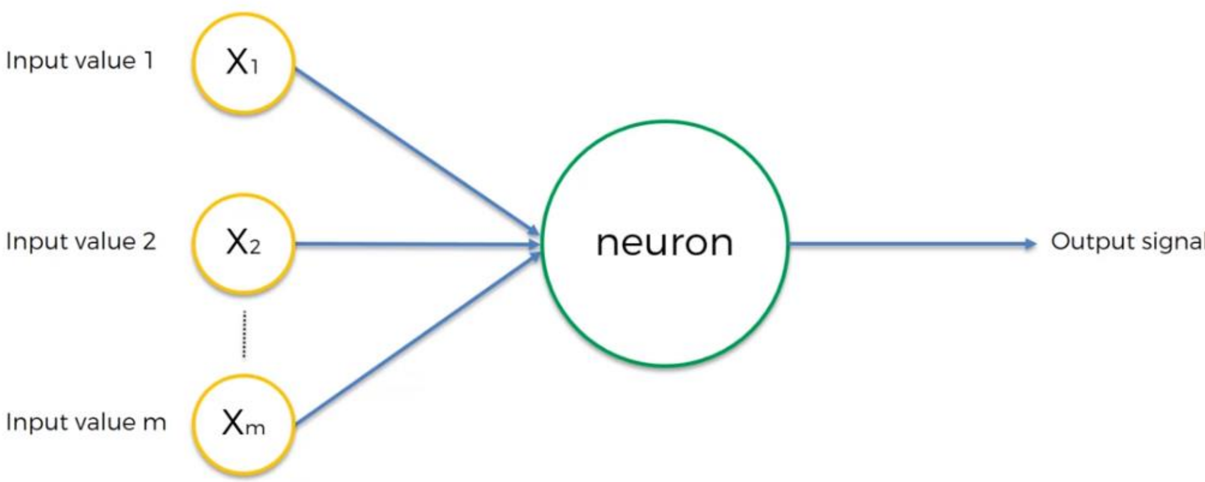
Source.

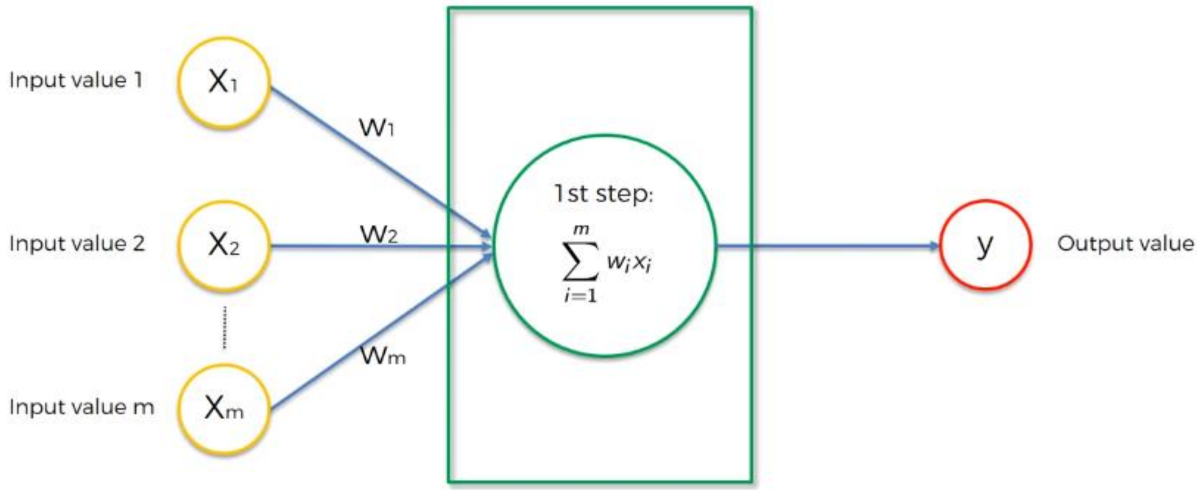


Geoffrey Hinton

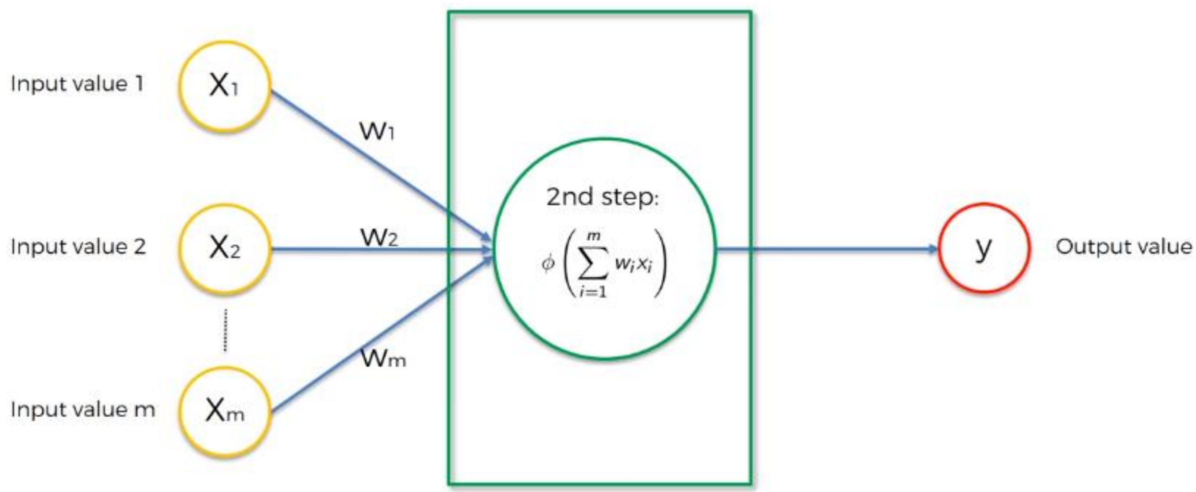
Neurons:

Approximately 100 billion neurons in human brain





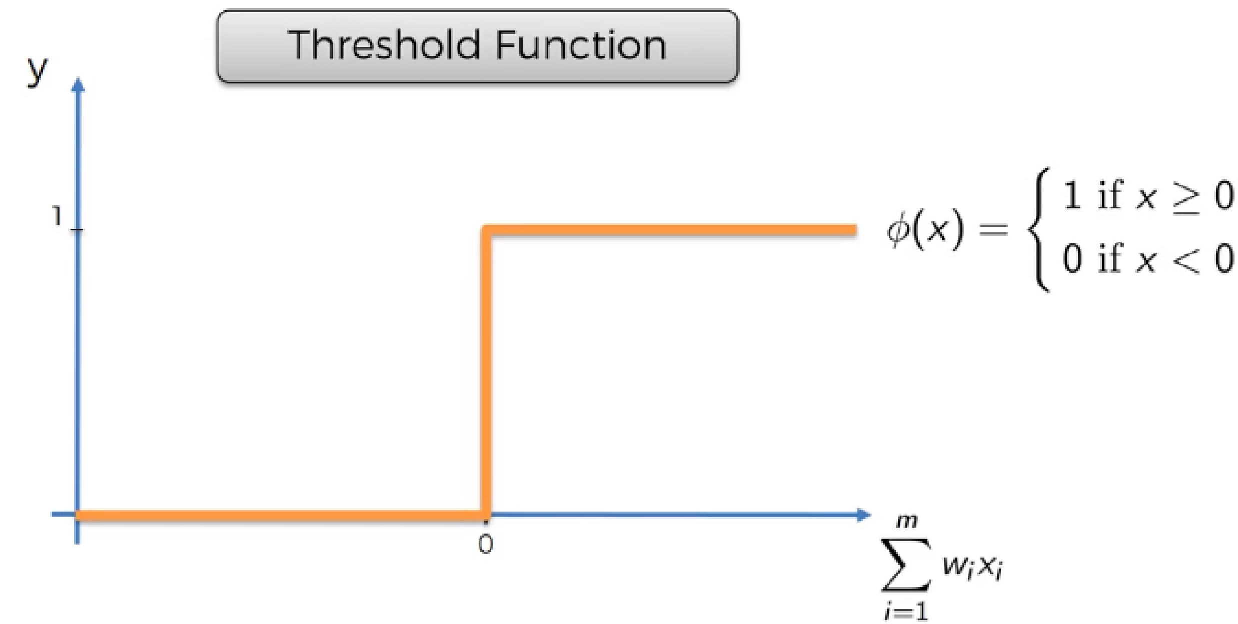
Input value 1
Input value 2
Input value m
X1
X2
Xm
W1
W2
Wm
1st step:
$\sum_{i=1}^{m} w_i x_i$
y
Output value



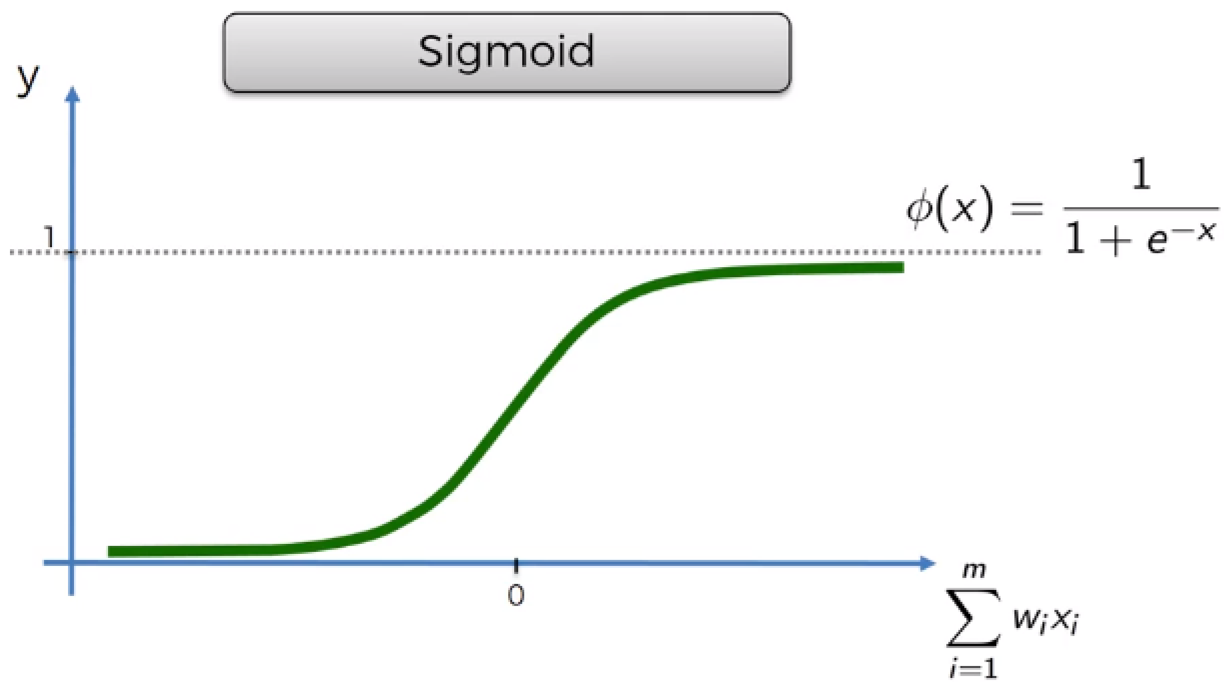
Input value 1
Input value 2
Input value m
X1
X2
Xm
W1
W2
Wm
2nd step:
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
y
Output value

Four main types of activation functions

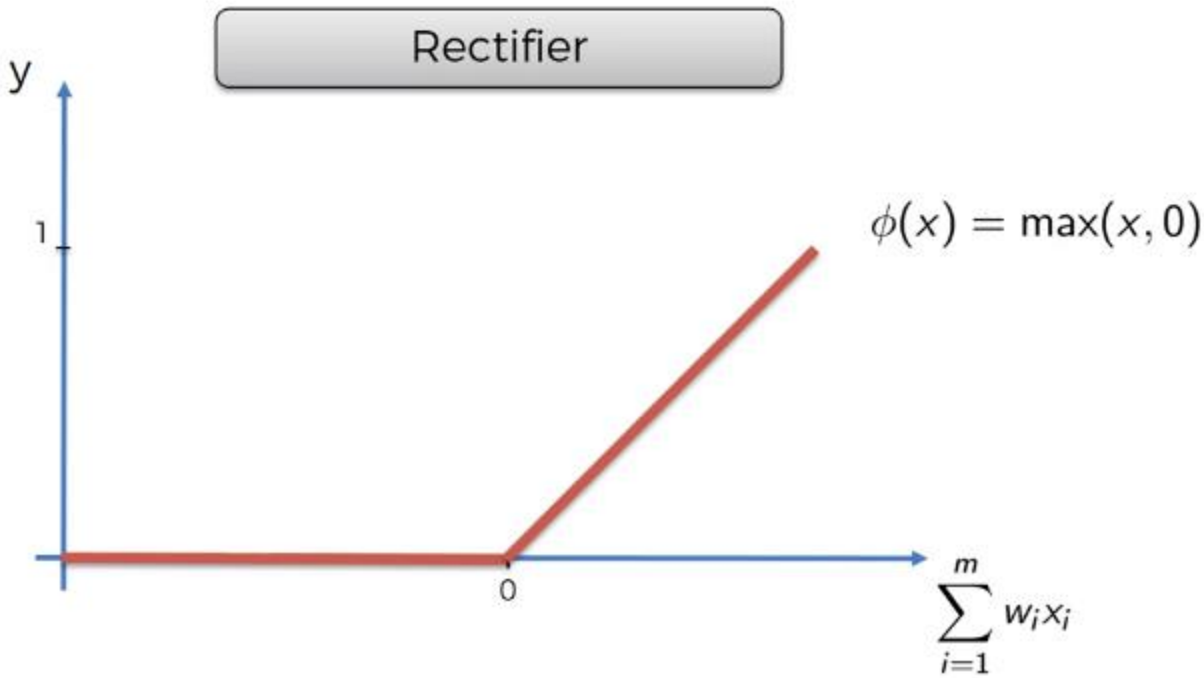
1.



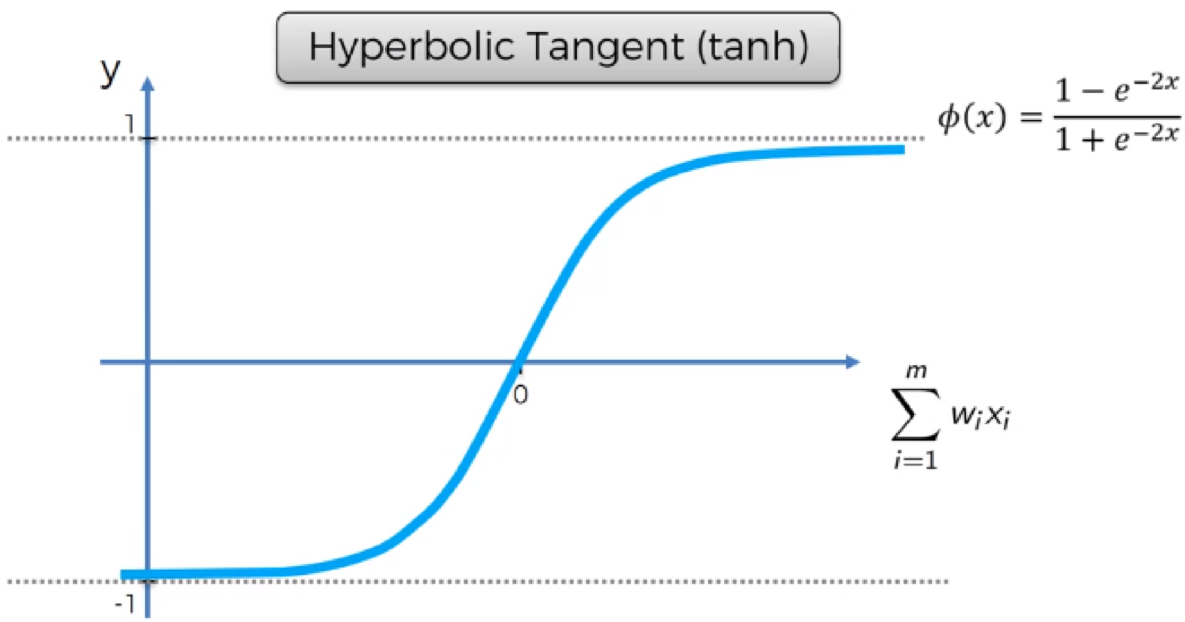
2.



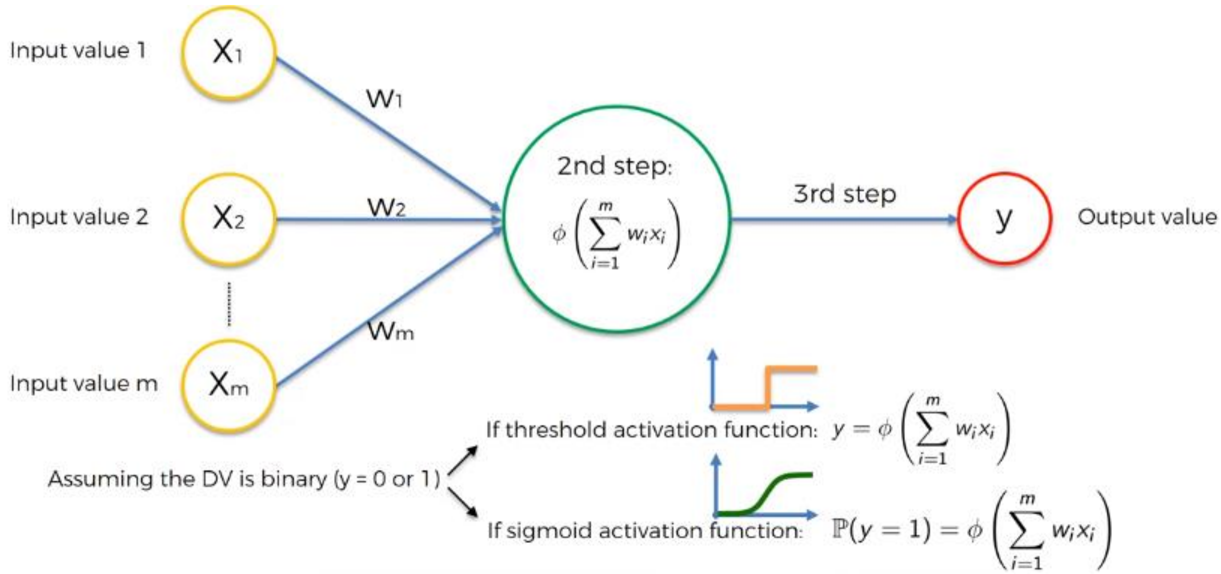
3.

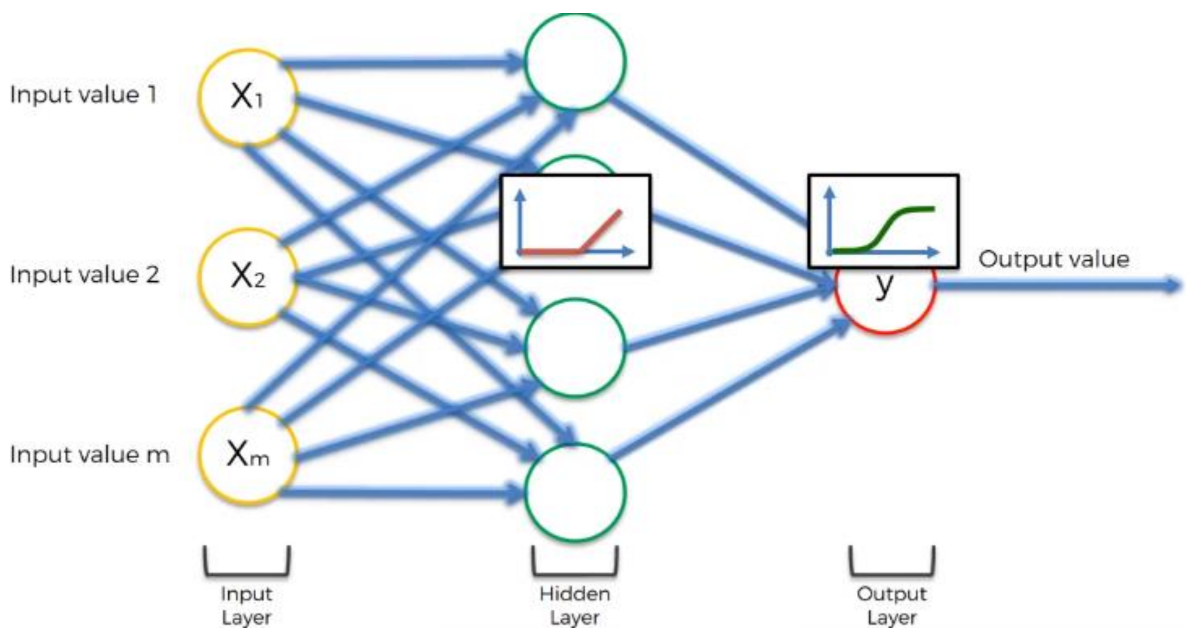


4.

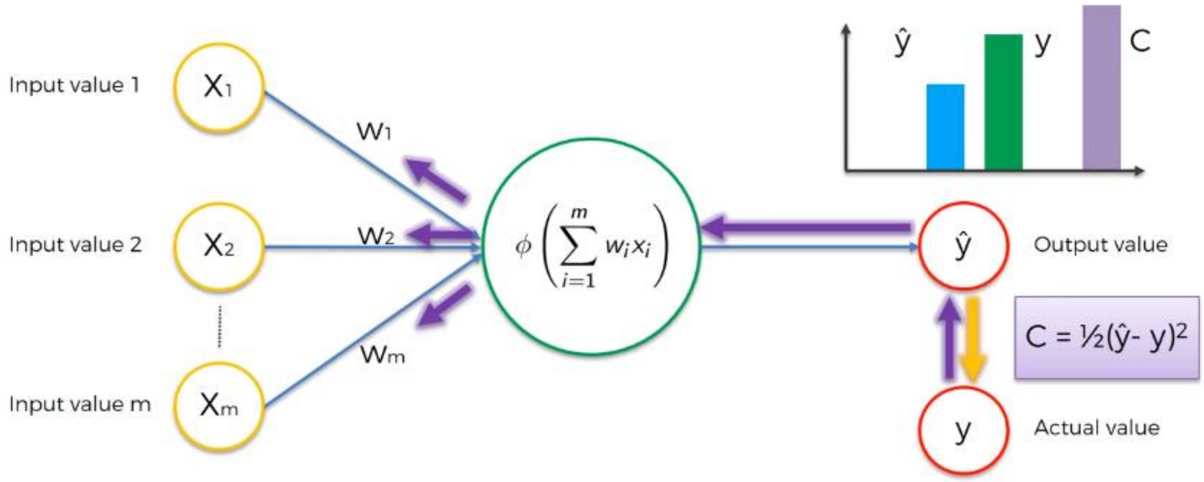


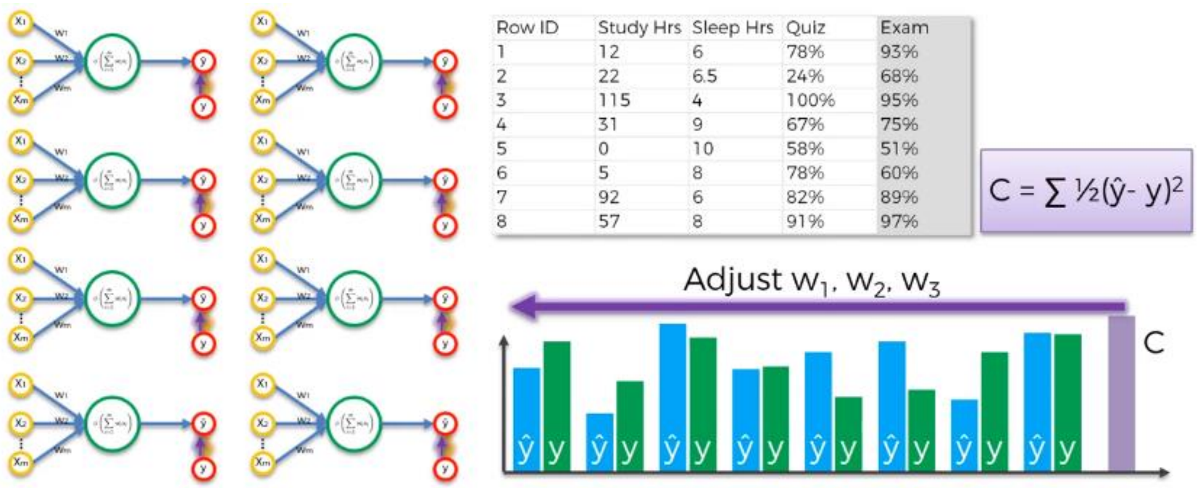
When do we use these?


Input value 1
X_1
w_1
2nd step:
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
3rd step
y
Output value
Input value 2
X_2
w_2
w_m
Input value m
X_m
If threshold activation function: $y = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$
Assuming the DV is binary (y = 0 or 1)
If sigmoid activation function: $\mathbb{P}(y = 1) = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$


Input value 1
X_1
Input value 2
X_2
Input value m
X_m
y
Output value
Input Layer
Hidden Layer
Output Layer

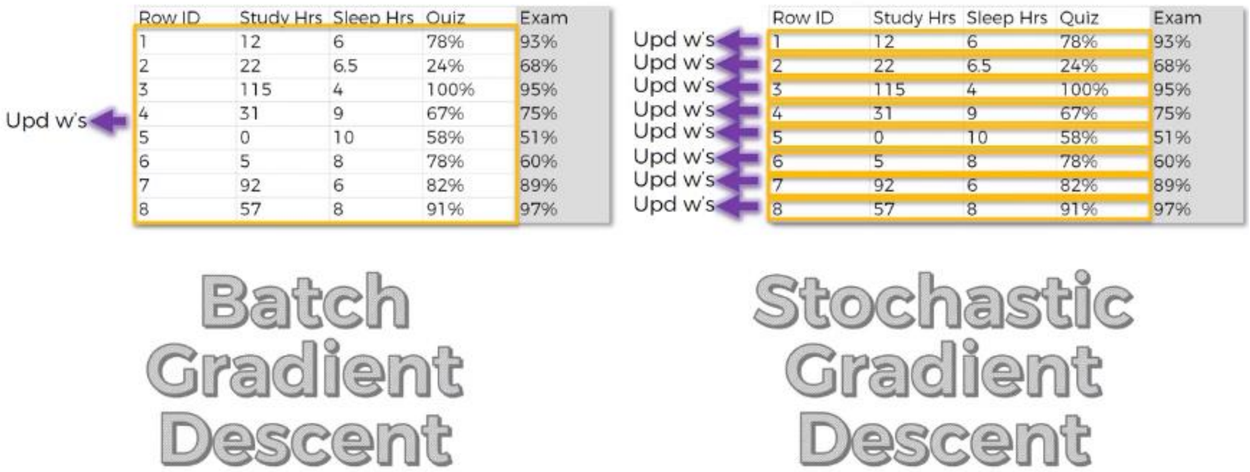
How do neural networks learn





Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

Stochastic gradient descent



Mini batch gradient descent combines these two approaches.

Backpropagation

STEP 1: Randomly initialise the weights to small numbers close to 0 (but not 0).

STEP 2: Input the first observation of your dataset in the input layer, each feature in one input node.

STEP 3: Forward-Propagation: from left to right, the neurons are activated in a way that the impact of each neuron's activation is limited by the weights. Propagate the activations until getting the predicted result y.

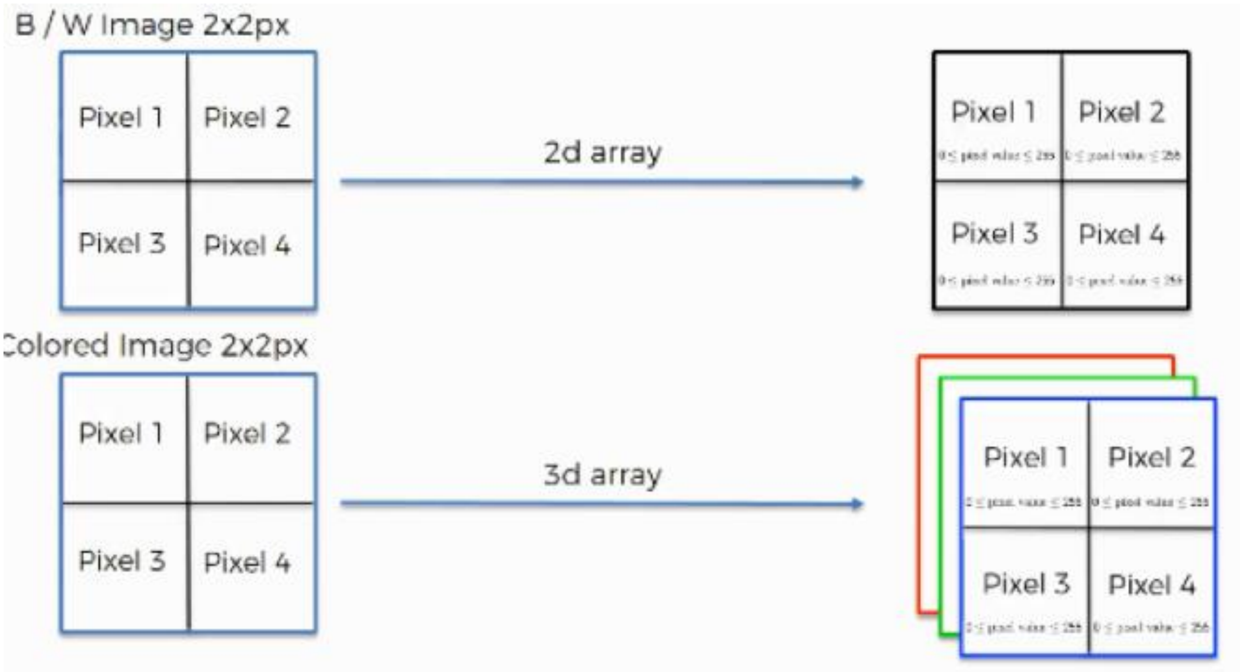
STEP 4: Compare the predicted result to the actual result. Measure the generated error.

STEP 5: Back-Propagation: from right to left, the error is back-propagated. Update the weights according to how much they are responsible for the error. The learning rate decides by how much we update the weights.

STEP 6: Repeat Steps 1 to 5 and update the weights after each observation (Reinforcement Learning). Or:
Repeat Steps 1 to 5 but update the weights only after a batch of observations (Batch Learning).

STEP 7: When the whole training set passed through the ANN, that makes an epoch. Redo more epochs.

Convolutional Neural Networks



The way black & white images are scanned differs in one major way from how colored images are. Both types of images are similar in the following respects:

- Each pixel contains 8 bits (1 byte) of information.
- Colors are represented on a scale from 0 to 255. The reason for this is that bits are binary units, and since we have 8 of these per byte, a byte can have any of 256 (2^8) possible values. Since we count 0 as the first possible value, we can go up to 255.
- In this model, 0 is pitch black and 255 is pure white, and in between are the various (definitely more than 50!) shades of gray.
- The network does not actually learn colors. Since computers understand nothing but 1's and 0's, the colors' numerical values are represented to the network in binary terms.

Differences:

- Black & white images are two-dimensional, whereas colored images are three-dimensional. The difference this makes is in the value assigned to each pixel when presented to the neural network. In the case of two-dimensional black & white images, each pixel is assigned one number between 0 and 255 to represent its shade.

- On the other hand, each pixel inside a colored image is represented on three levels. Since any color is a combination of red, green, and blue at different levels of concentration, a single pixel in a colored image is assigned a separate value for each of these layers.

- That means that the red layer is represented with a number between 0 and 255, and so are the blue and the green layers. They are then presented in an RGB format. For example, a “hot pink” pixel would be presented to the neural network as (255, 105, 180).

For notes, refer to self notes folder