Fundamentals of Artificial Neural Networks (II)

AI: Deep Learning and Neural Networks

Thuan L Nguyen, PhD

Slide 2: Fundamentals of Artificial Neural Networks

1. Biological Neurons and Neural Networks
2. Artificial Neurons and Perceptron
3. Perceptron: A Simple Neural Network
4. Artificial Neural Networks: An Introduction
5. Artificial Neural Networks: Computation Power
6. Artificial Neural Networks: Architectures
7. Artificial Neural Networks: Applications

Slide 3: Fundamentals of Artificial Neural Networks

Perceptron: The Simplest Form of a Neural Network (Review)

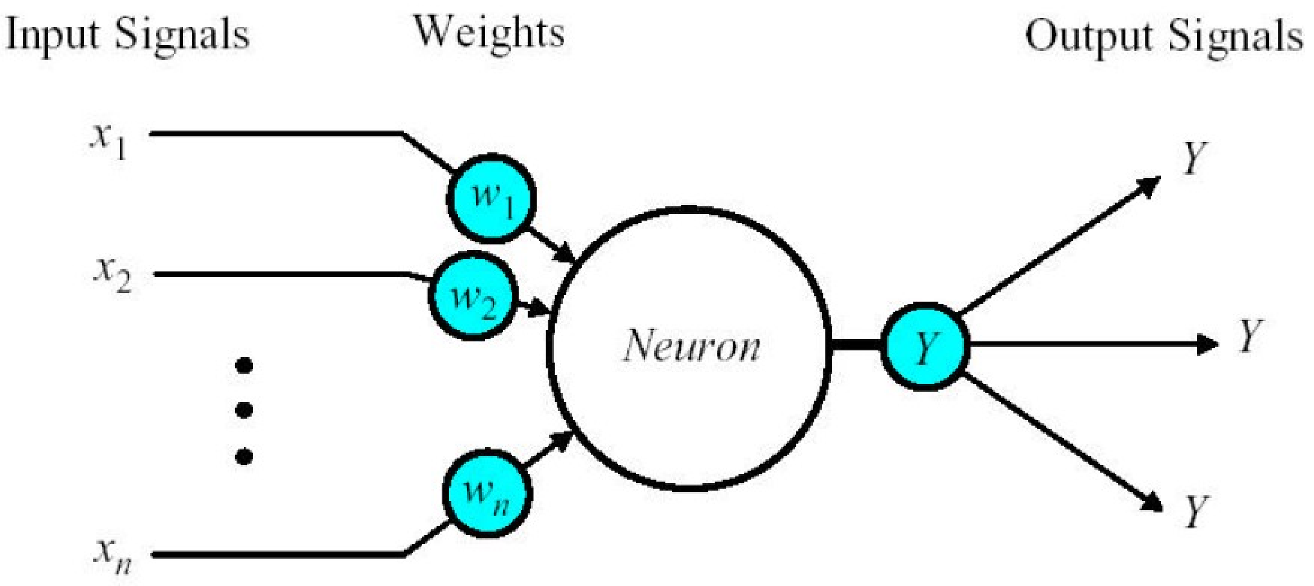
- A perceptron:
  - Frank Rosenblatt introduced the concept of a perceptron (1958):
    - He proposed a training algorithm that provided the first procedure for training a simple artificial neural network called perceptron.

- Perceptron: The simplest form of a neural network.
  - It consists of a single neuron with adjustable synaptic weights and a hard limiter.

Slide 4: Fundamentals of Artificial Neural Networks

Perceptron: A Network of The McCulloch-Pitts Neurons

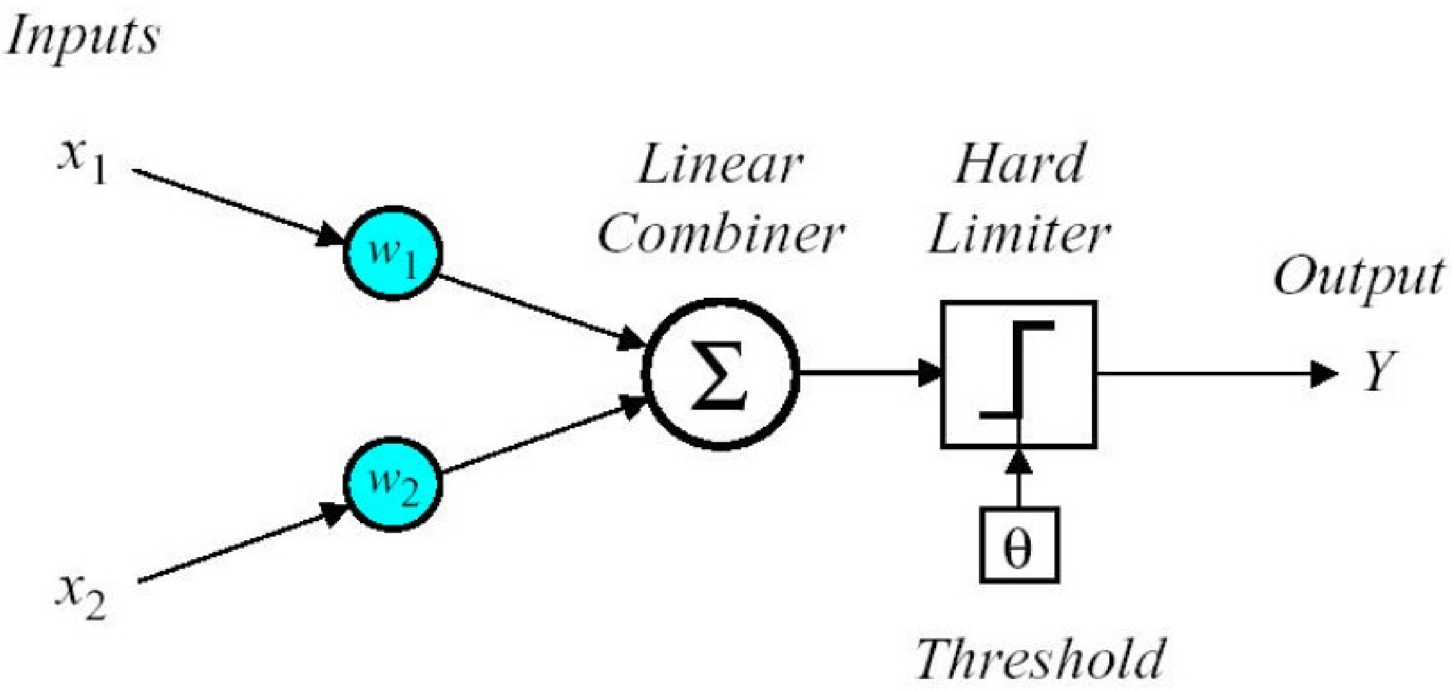
A **simple rate coding model** of real neurons is also known as a **Threshold Logic Unit** :



McCulloch-Pitts Model (Source: Wikipedia)

Slide 5: Fundamentals of Artificial Neural Networks

Perceptron: A Network of The McCulloch-Pitts Neurons



McCulloch-Pitts Model (Source: Wikipedia)

Slide 6: Fundamentals of Artificial Neural Networks

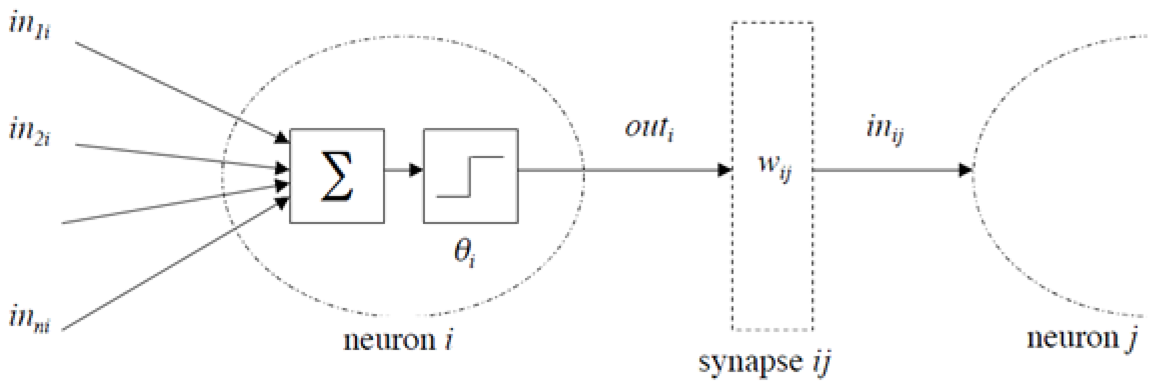
Perceptron: A Network of The McCulloch-Pitts Neurons

- Frank Rosenblatt introduced the concept of a perceptron (1958)
  - Each input I_i is multiplied by a weight w_{ji} (synaptic strength)
  - These weighted inputs are summed to give the activation level, A_j
  - The activation level is then transformed by an activation function to produce the neuron's output, Y_i
  - W_{ji} is known as the weight from unit i to unit j
    - $W_{ji} > 0$, synapse is excitatory
    - $W_{ji} < 0$, synapse is inhibitory
  - Note that I_i may be
    - External input
    - The output of some other neuron

Slide 7: Fundamentals of Artificial Neural Networks

Perceptron: Networks of McCulloh-Pitts Neurons

- To finish a meaningful computation task, it is necessary to have a network of multiple neurons:



$$out_k w_{ki} = in_{ki} \qquad out_i = step(\sum_{k=1}^{n} in_{ki} - \theta_i) \qquad out_i w_{ij} = in_{ij}$$

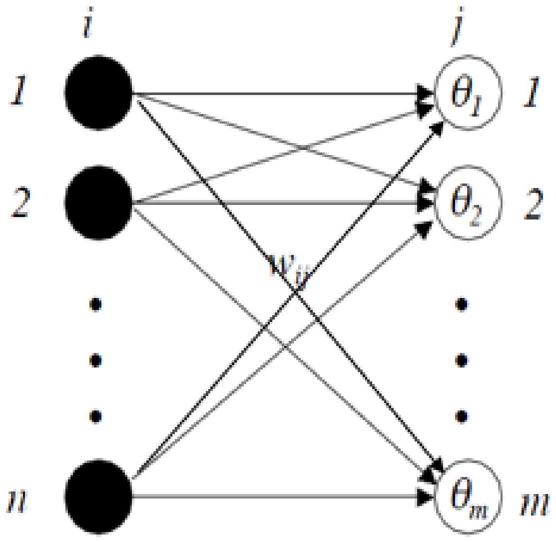
McCulloch-Pitts Model (Source: Wikipedia)

Slide 8: Fundamentals of Artificial Neural Networks

Deep Learning: Simple Single-Layer Neural Networks

Perceptron:

- The fundamental unit of an artificial neural network
- A simple – single-layer – artificial neural network:
    - A simple neural network that has one layer of input neurons feeding forward to one output layer of McCulloch-Pitts neurons, with full connectivity.



$$out_j = step\left(\sum_{i=1}^{n} in_i w_{ij} - \theta_j\right)$$

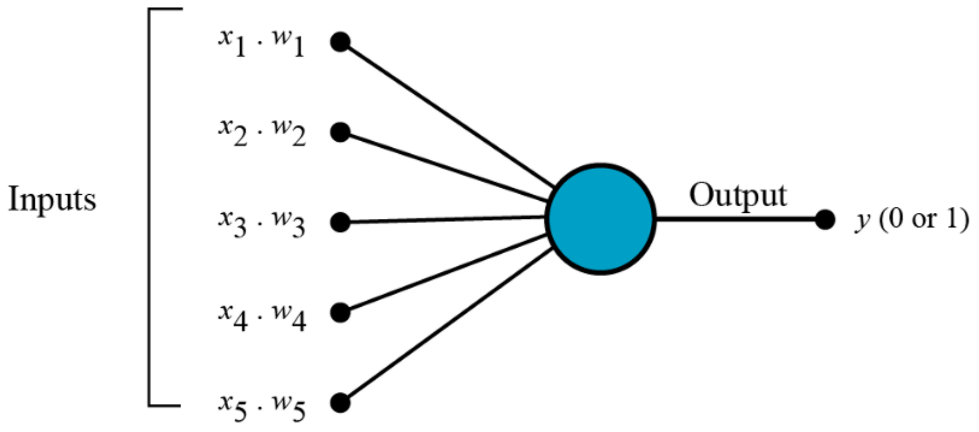
AI Deep Learning: Perceptron (Source: Wikipedia)

Slide 9: Fundamentals of Artificial Neural Networks

Deep Learning: Simple Single-Layer Neural Networks

Perceptron:

- The McCulloch-Pitts neuron model is actually the **simplest** single-layer neural network.
  - One or more inputs → One output
- Therefore, the McCulloch-Pitts neuron model represents a **perceptron,** the simplest neural network.



McCulloch-Pitts Model (Source: towardsdatascience.com)

Slide 10: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: Proof of Computation Power

It is possible to implement any form of logic operators – based on one or more of the basic logic gates: NOT, AND, and OR – using McCulloch-Pitts neurons:

- All to be done is to find the appropriate connection weights and neuron thresholds to produce the right outputs for each set of inputs.
- Then to show explicitly that it is possible to construct simple networks that perform the logic operator – NOT, AND, and OR.
- It is well known that many other logical functions can be constructed from these three basic operations.
    - Therefore, a network of McCulloch-Pitts neurons, i.e., artificial neural networks can compute almost all logical functions.
- In conclusion, the McCulloch-Pitts neural model can be represented and used for computation.

Slide 11: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: AND Networks

Given a simple neural network:

- Two input weights – w_1 and w_2
- A threshold θ
- For each training pattern, it is necessary to achieve: out = sgn $((w_1 i_1 + w_2 i_2) - \theta)$

The training data can result in four inequalities:

in_1	in_2	out
0	0	0
0	1	0
1	0	0
1	1	1

$$\Rightarrow \quad \begin{array}{l} w_1 0 + w_2 0 - \theta < 0 \\ w_1 0 + w_2 1 - \theta < 0 \\ w_1 1 + w_2 0 - \theta < 0 \\ w_1 1 + w_2 1 - \theta \geq 0 \end{array} \quad \Rightarrow \quad \begin{array}{c} \theta > 0 \\ w_2 < \theta \\ w_1 < \theta \\ w_1 + w_2 \geq \theta \end{array}$$

- So, there are an infinite number of solutions to this system of inequalities of the AND network.
- It is the same for the neural networks of the other two basic logic operations: NOT and OR.

Slide 12: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: Proof of Computation Power

NOT

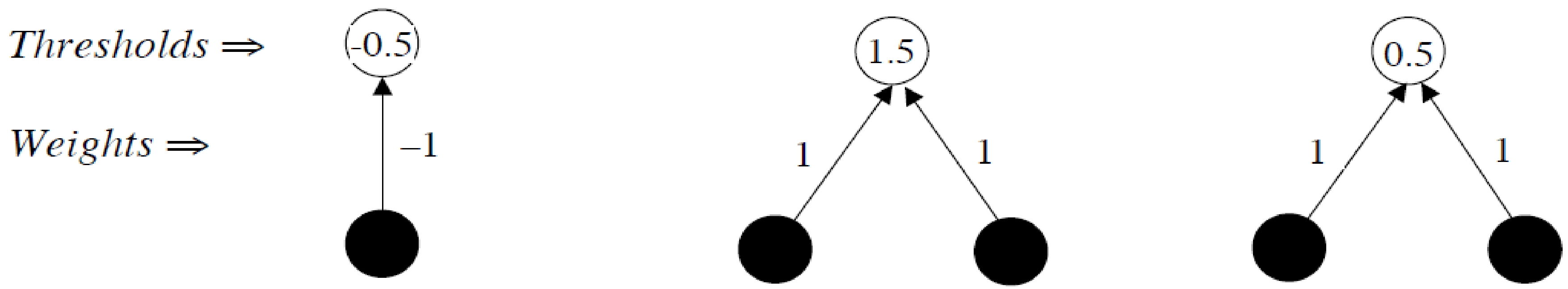
in	out
0	1
1	0

AND

in_1	in_2	out
0	0	0
0	1	0
1	0	0
1	1	1

OR

in_1	in_2	out
0	0	0
0	1	1
1	0	1
1	1	1



Slide 13: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: Limitations of Simple Neural Networks

Let's do the same for the XOR logic operation. The training data can lead to four inequalities:

in_1	in_2	out
0	0	0
0	1	1
1	0	1
1	1	0

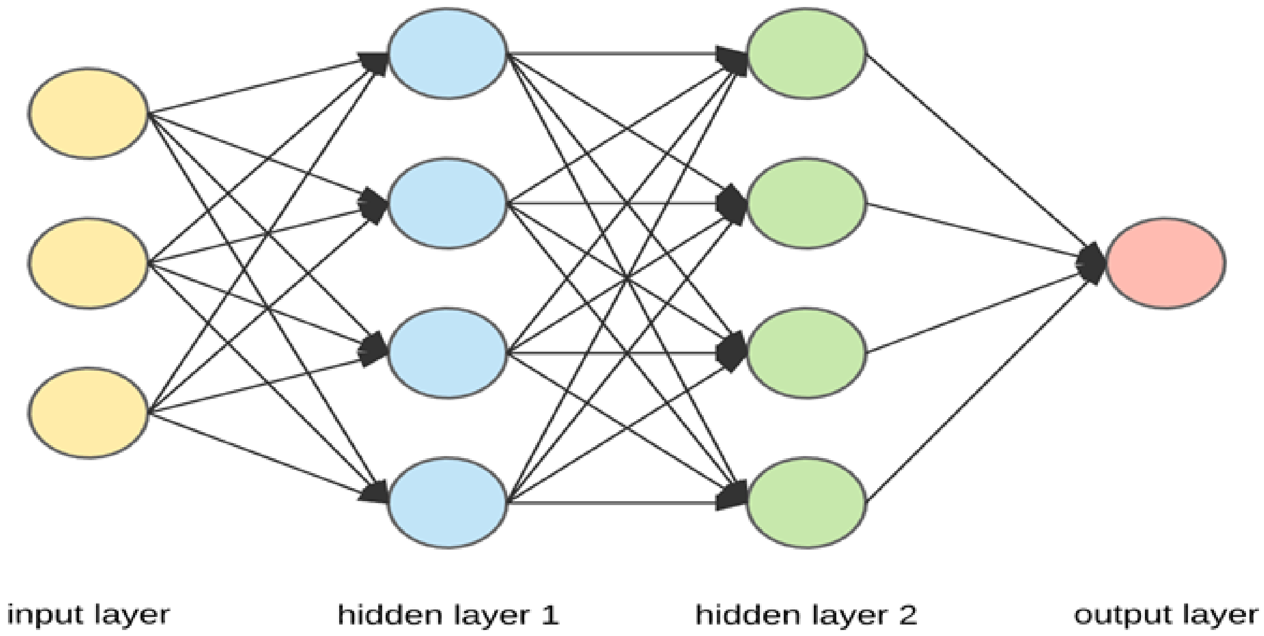
$$\Rightarrow \quad \begin{array}{l} w_1 0 + w_2 0 - \theta < 0 \\ w_1 0 + w_2 1 - \theta \geq 0 \\ w_1 1 + w_2 0 - \theta \geq 0 \\ w_1 1 + w_2 1 - \theta < 0 \end{array} \quad \Rightarrow \quad \begin{array}{c} \theta > 0 \\ w_2 \geq \theta \\ w_1 \geq \theta \\ w_1 + w_2 < \theta \end{array}$$

- Clearly the first, second and third inequality are incompatible with the fourth.
  - So, there is no solution.
- It's necessary to use more complex networks – Multi-Layered Neural Networks (MLP).
  - The networks that combine together many simple networks.
- With more complex neural networks:
  - It is much more difficult to determine all the weights and thresholds by hand.
  - It is necessary to use AI software frameworks such as TensorFlow, PyTorch, etc.

Slide 14: Fundamentals of Artificial Neural Networks

Deep Learning: Multi-Layer Neural Networks

- Single-layer perceptrons: very limited regarding the computation power
- Multi-layer perceptrons, i.e., multi-layer neural networks, were constructed.



AI Deep Learning: Multi-layer Neural network (Source: medium.com)

Slide 15: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: Architectures

- Mathematically, artificial neural networks can be represented as weighted directed graphs.
- It is possible to simply think of artificial neural networks as the flows of activation transferring between neurons, i.e., processing units, via one-way connections.
- The three most common ANN architectures are:
  - Single-Layer Feed-forward Neural Networks
  - Multi-Layer Feed-forward Neural Networks
  - Recurrent Neural Networks

Slide 16: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: Architectures

- Single-Layer Feed-Forward Neural Networks:
  - One input layer and one output layer of processing units.
  - No feed-back connections.
  - For example, a simple Perceptron

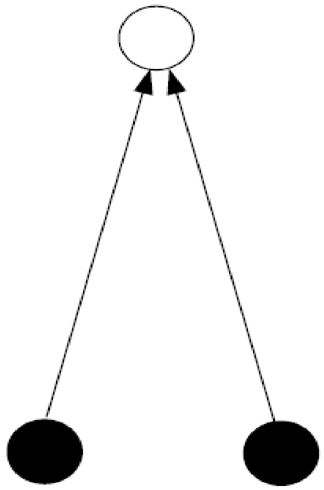
- Multi-Layer Feed-Forward Neural Networks:
  - One input layer, one output layer, and one or more hidden layers of processing units.
  - No feed-back connections.
  - Hidden layers sit in between the input and output layers, i.e., hidden from the outside world.
  - For example, a Multi-Layer Perceptron (MLP)

- Recurrent Neural Networks:
  - Any network with at least one feed-back connection.
  - It may, or may not, have hidden units.
  - For example, a Simple Recurrent Network

Slide 17: Fundamentals of Artificial Neural Networks

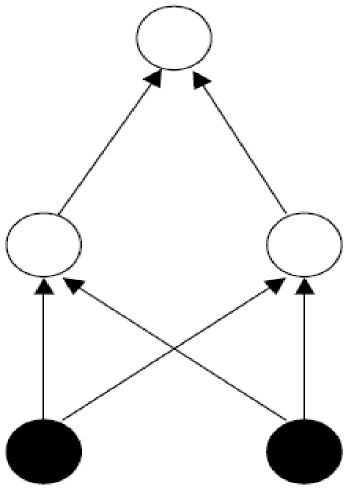
Artificial Neural Networks: Architectures

Single Layer Feed-forward



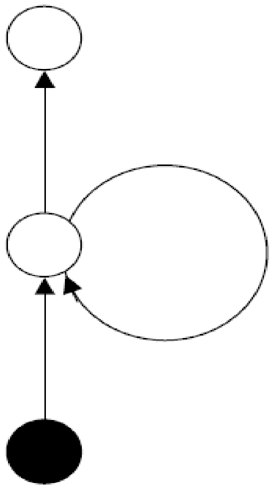
Single-Layer Perceptron

Multi-Layer Feed-forward



Multi-Layer Perceptron

Recurrent Network



Simple Recurrent Network

Slide 18: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: Applications

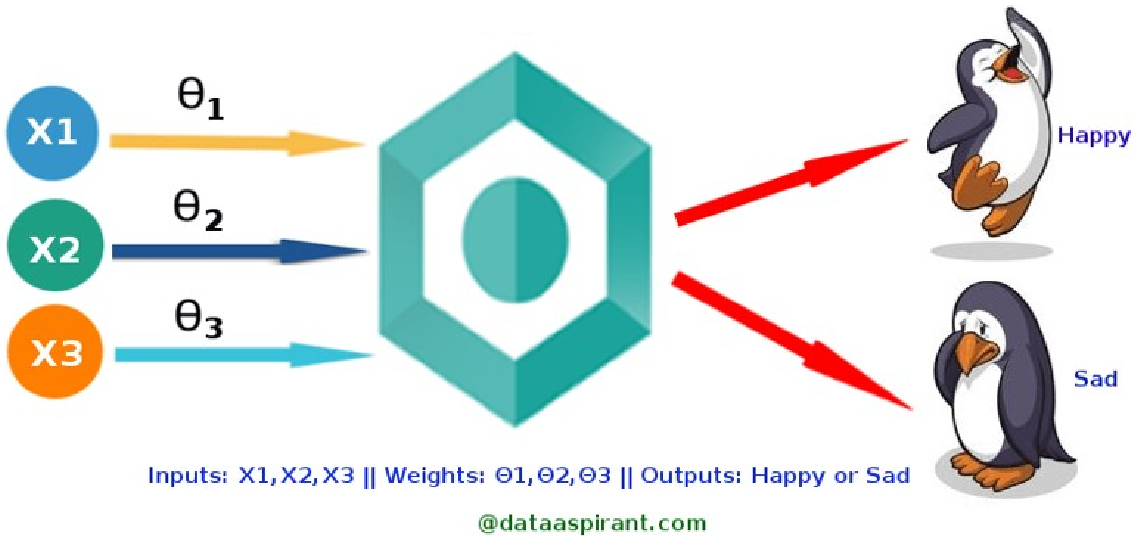
- Neural networks perform input-to-output mappings.
  - As shown, neural networks are capable of performing complex logic computations

- The network inputs and outputs can be simple numbers such as integers or complex structures of values such as n-dimension arrays

- The applications fall into two broad categories – **classification** and **regression**.

Slide 19: Fundamentals of Artificial Neural Networks

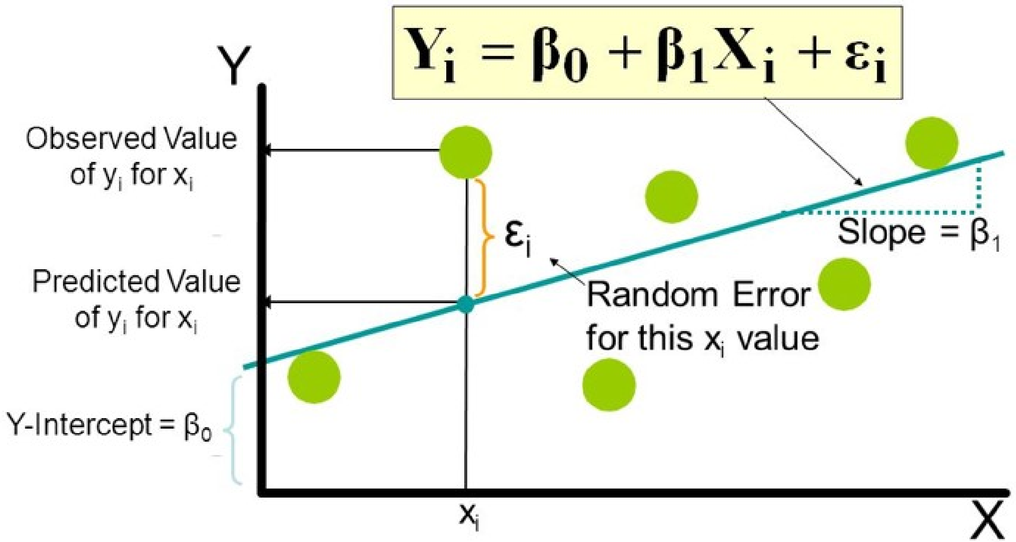
Artificial Neural Networks: Applications: Classification



- Classification
  - This task has the neural network decide from a set of inputs which class a given object falls in.
  - The inputs will be suitable “measurements” of the object.
  - The target outputs represent the associated class, usually as binary vectors.
  - Ideally, the network will output the probability of the object being in each possible class.

Slide 20: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: Applications: Regression



- Regression problems are essentially a form of function approximation.
  - The network's linear outputs are real valued numbers corresponding to some underlying function of the inputs that needs to be determined from noisy training data.
  - A special case: Time series prediction
    - Inputs are measurements at successive points in time
    - Output is the prediction of that measurement at later points in time.

Slide 21: Fundamentals of Artificial Neural Networks

Artificial Neural Networks: Activation Functions: Examples