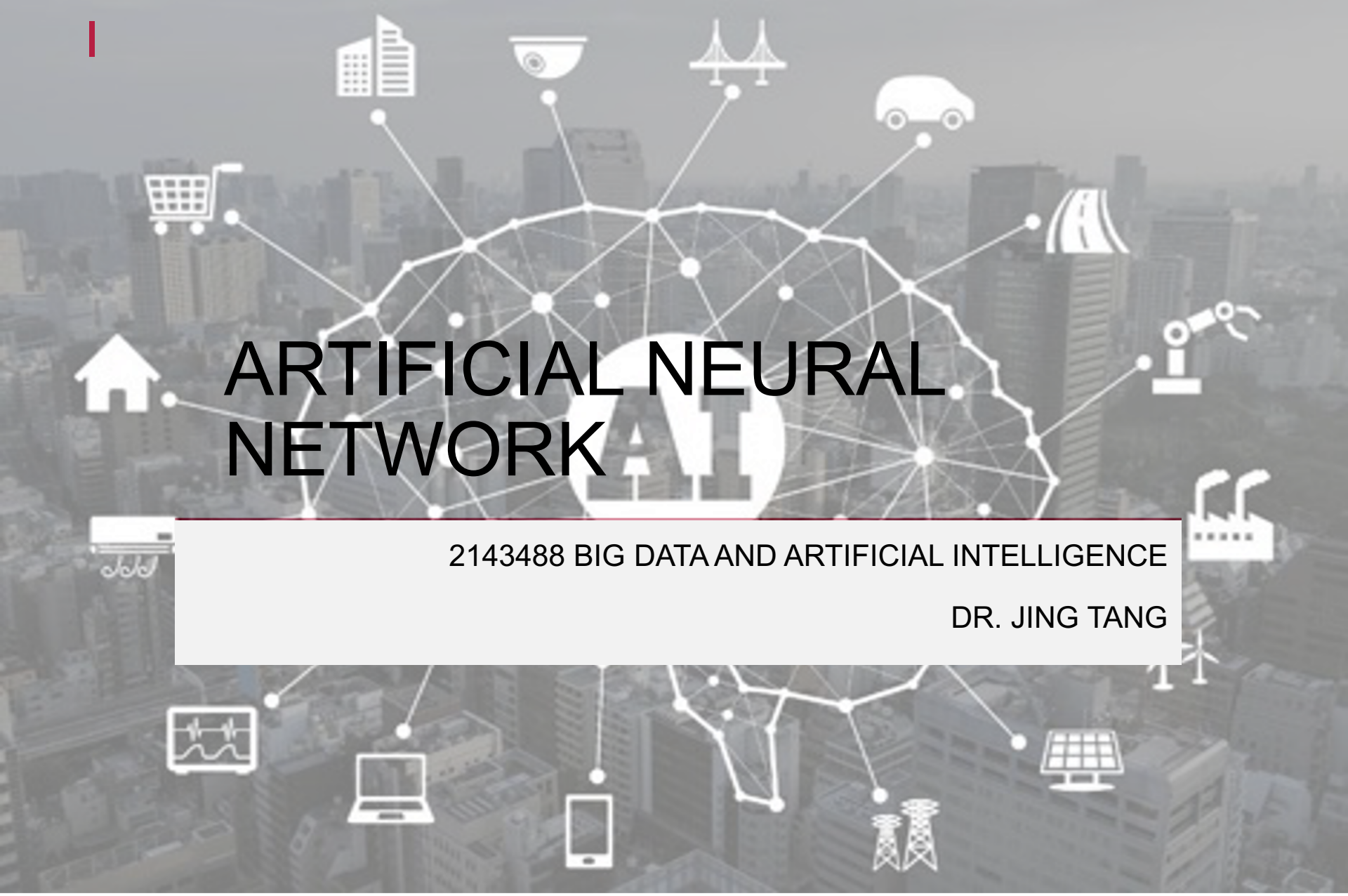


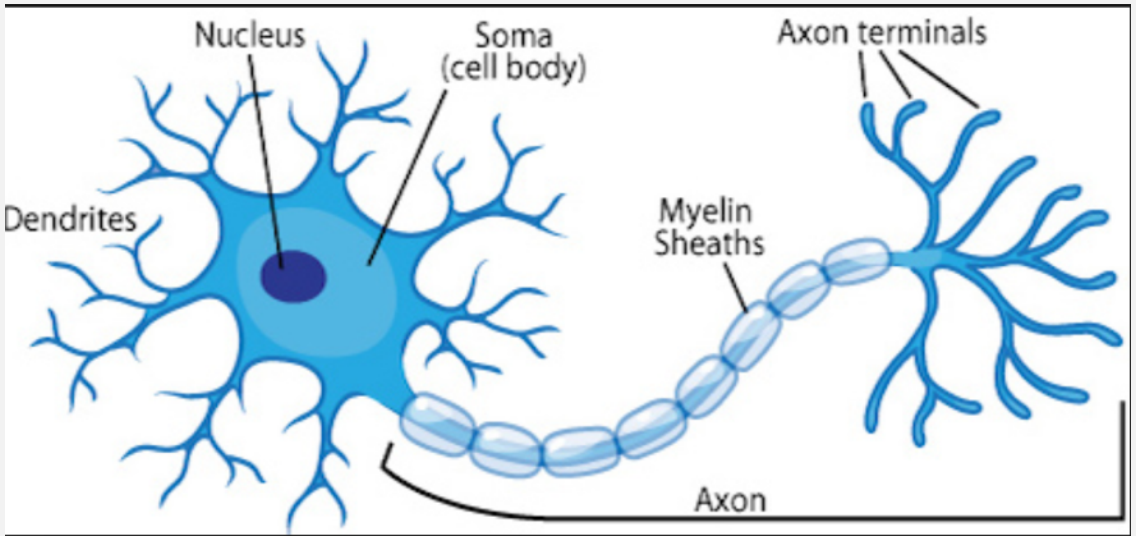
ARTIFICIAL NEURAL NETWORK

2143488 BIG DATA AND ARTIFICIAL INTELLIGENCE

DR. JING TANG

BIOLOGICAL NEURON NETWORK

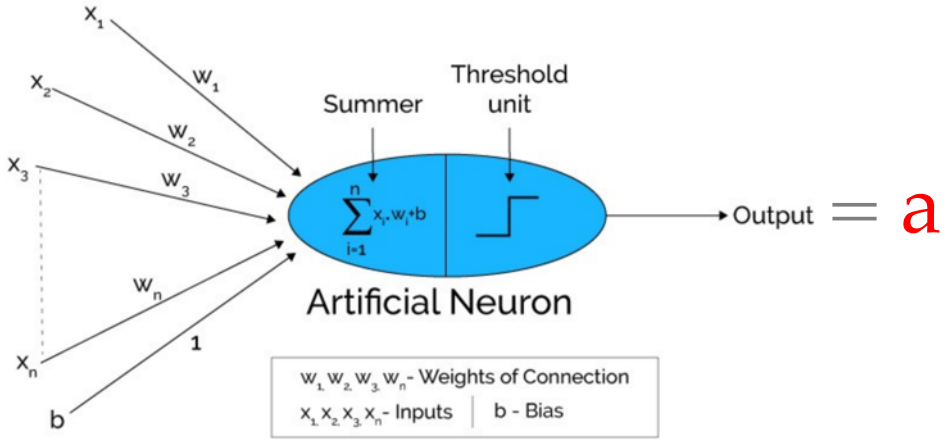
- **Dendrites**: accepts the inputs
- **Soma**: process the inputs
- **Axon**: turn the processed inputs into output
- **Synapses**: the electrochemical contact between the neurons.



HOW IS BRAIN DIFFERENT FROM COMPUTERS?

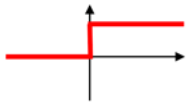
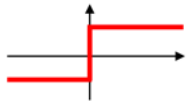
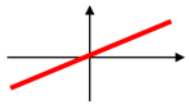
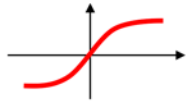
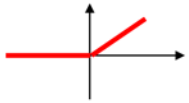
	
Biological neurons or nerve cells	Silicon transistors
200 billion neurons, 32 trillion interconnections.	1 billion bytes RAM, trillion of bytes on disk.
Neuron size: 10-6 m.	Single transistor size: 10-9m.
Energy consumption: 6-10 joules per operation per sec.	Energy consumption: 10-16 joules per operation per second.
Learning capability	Programming capability

ARTIFICIAL NEURON

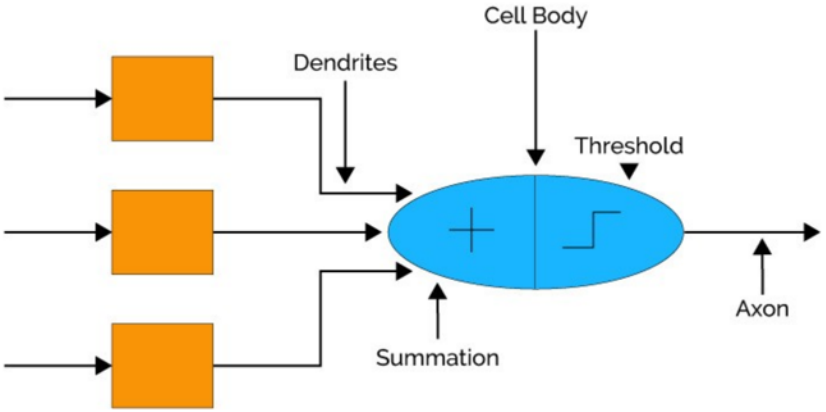


- **Input to the network** are represented by the mathematical symbol **(x_i)**
- Each of these inputs are multiplied by a **connection weight (w_i)**
- These products are simply summed together with **bias (b)**, and fed through the **transfer function (f)** to generate the result.

TRANSFER FUNCTIONS (THRESHOLD)

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	

TERMINOLOGY



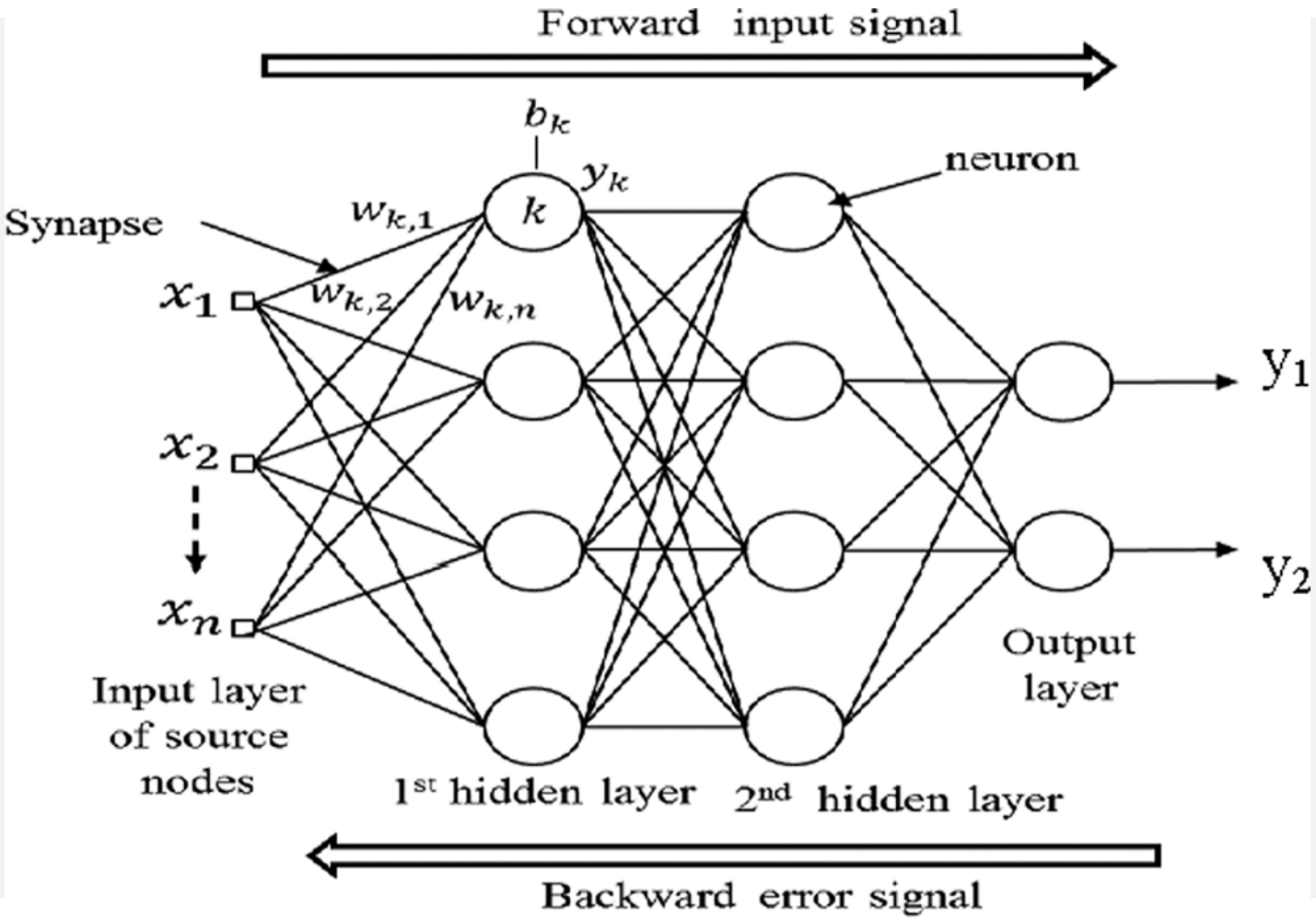
Biological neuron	Artificial neuron
Neuron	Node/unit/cell
Synapse	Connection/edge/link
Synapse efficiency	Connection strength/weight
Firing frequency	Node output

ARTIFICIAL NEURAL NETWORK (ANN)

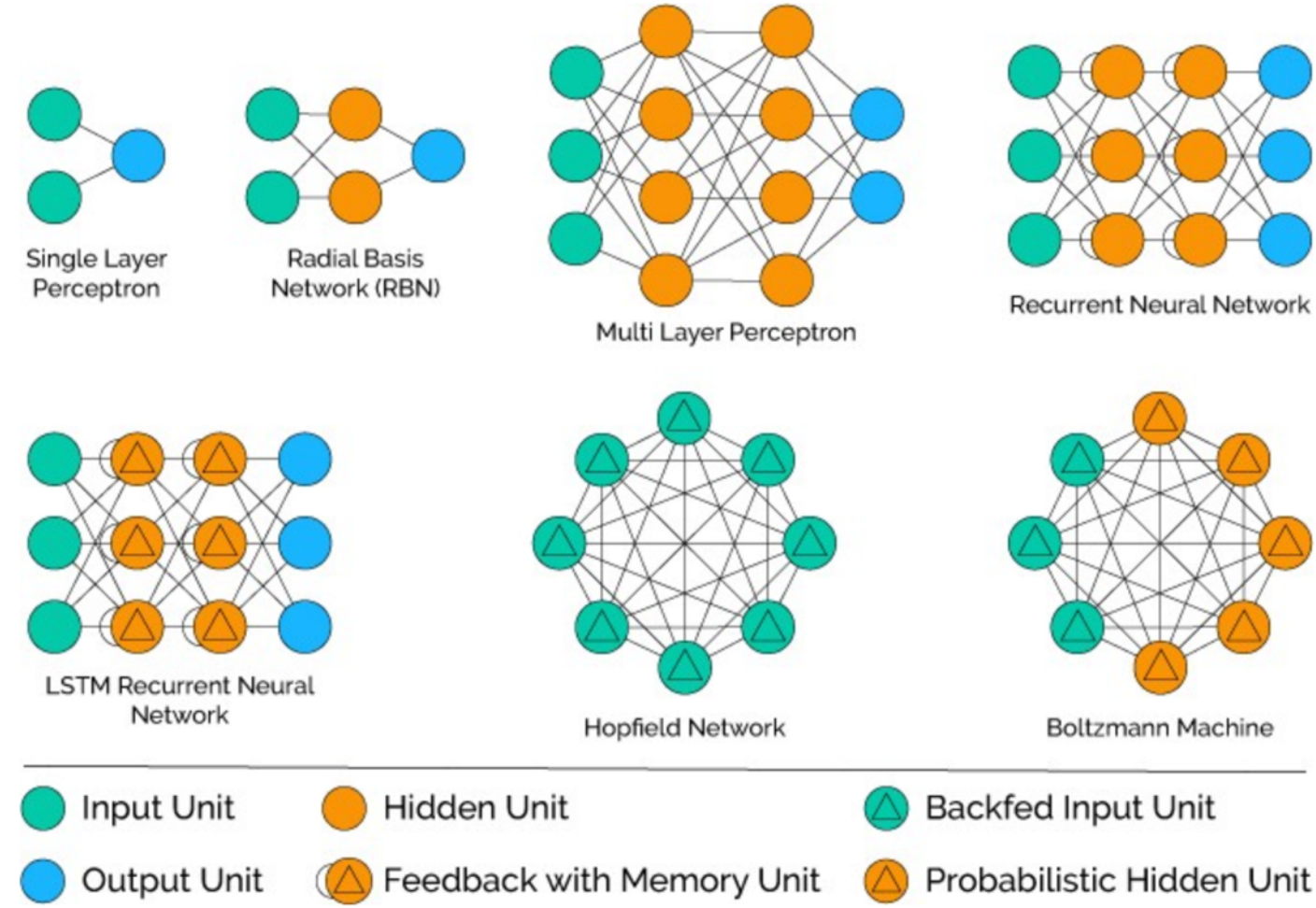
- Artificial neural network (ANN) is program designed to solve any problem by trying to mimic the structure and the function of **human nervous system**.
- ANN is based in simulated neurons joined together in a variety of ways to form a network.
- ANN resembles the human brain in the following two ways:
  - An ANN acquires knowledge through **learning**.
  - An ANN's knowledge is stored within the interconnection strength know as **synaptic weight**.

https://www.youtube.com/watch?v=aircAruvnKk&t=2s

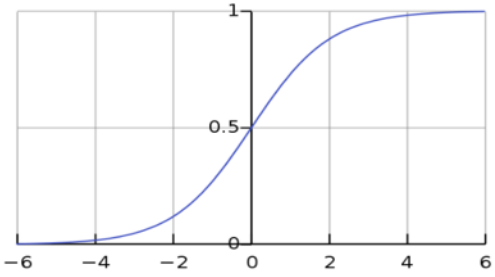
ARTIFICIAL NEURAL NETWORK



ARTIFICIAL NEURAL NETWORK



EXAMPLE 1:



$$f(x) = \frac{1}{1 + e^{-x}}$$

8

14

3

0.3

0.5

0.1

0.8

0.4

0.4

0.9

0.2

$f(x) = x$

Bias=1

Bias=1

LEARNING

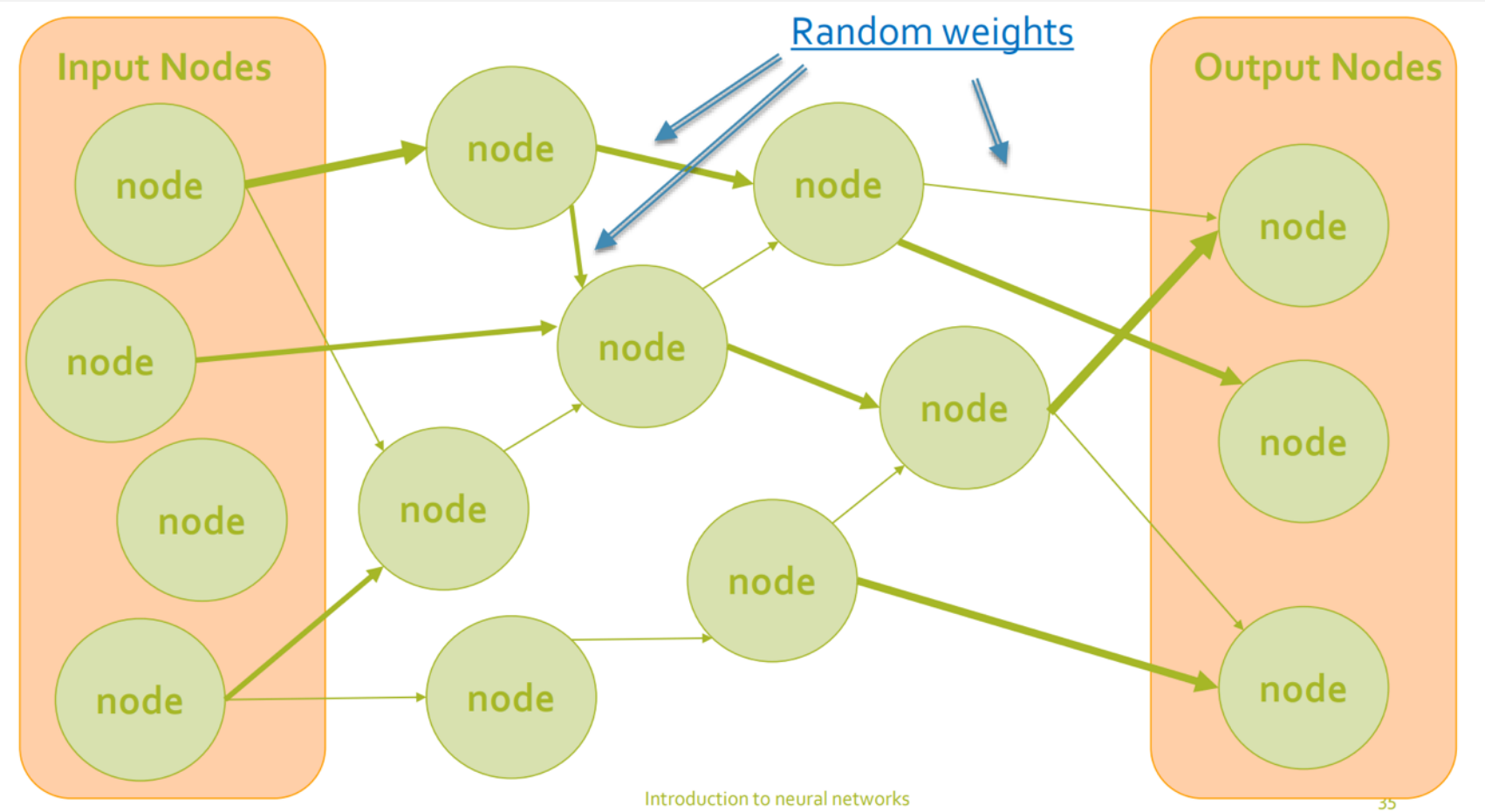
- How does the learning occur?
- What are possible **mathematical** models of learning?
- In ANN, learning refers to the method of modifying the **weight of connections** between the nodes of a specific network .
- The learning ability of a neural network is determined by its **architecture** and by the **algorithmic method** chosen for training.

BACKPROPAGATION ALGORITHM

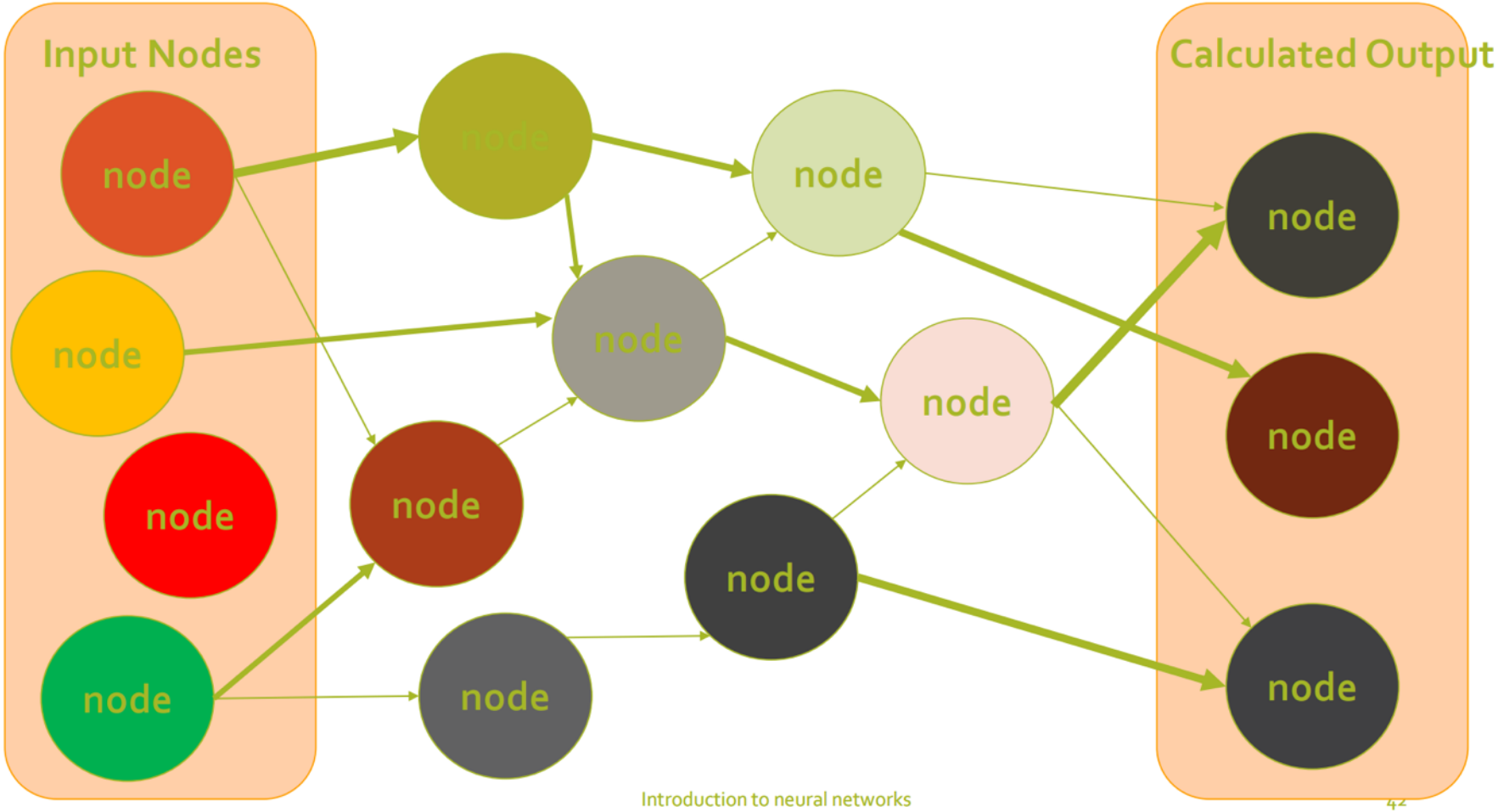
- The backpropagation algorithm (Rumelhart and McClelland, 1986) is used in layered feed-forward ANN.
- Back propagation is a multi-layer feed forward, **supervise** learning network based on **gradient descent** learning rule.
- The idea of the backpropagation algorithm is to reduce the error (difference between actual and expected results) by learning from the **training data**.

https://www.youtube.com/watch?v=Ilg3gGewQ5U

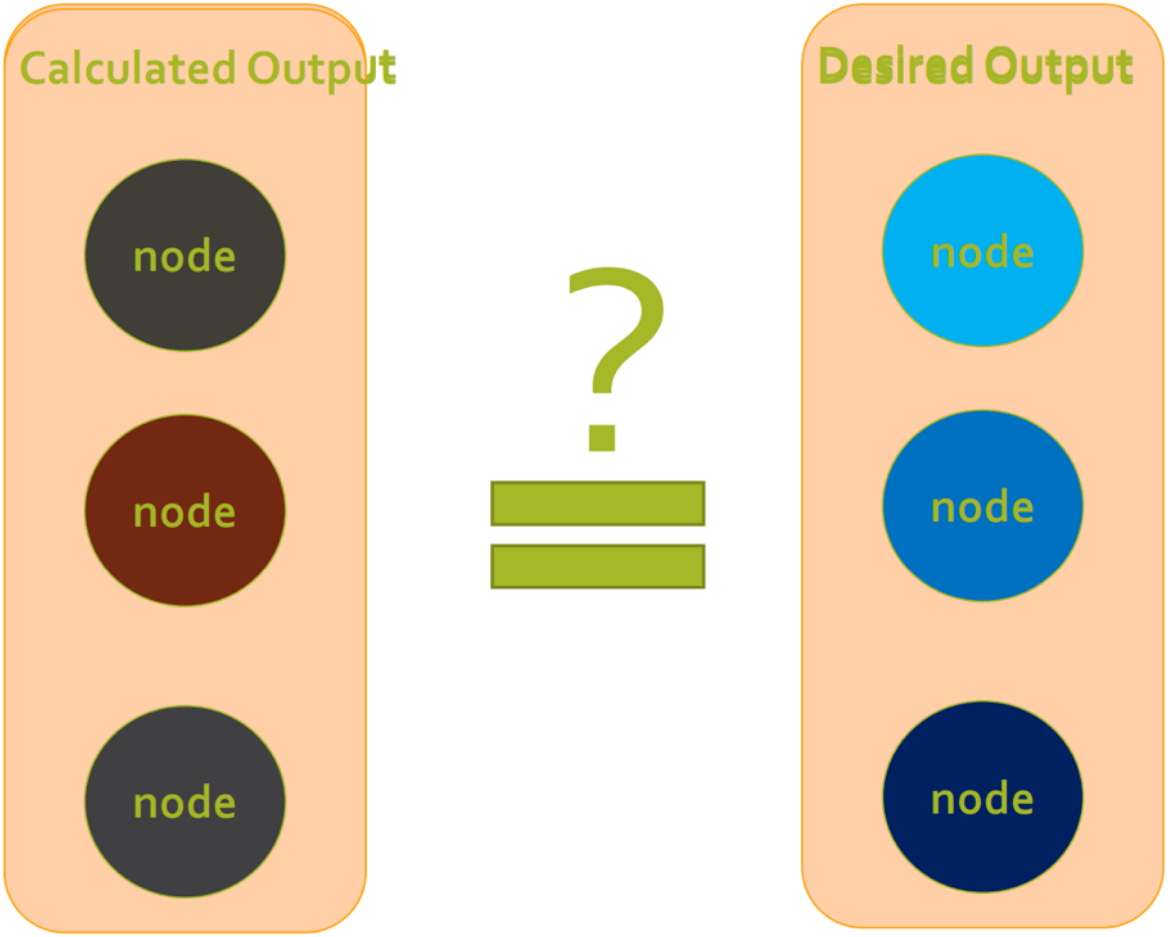
BACKPROPAGATION ALGORITHM



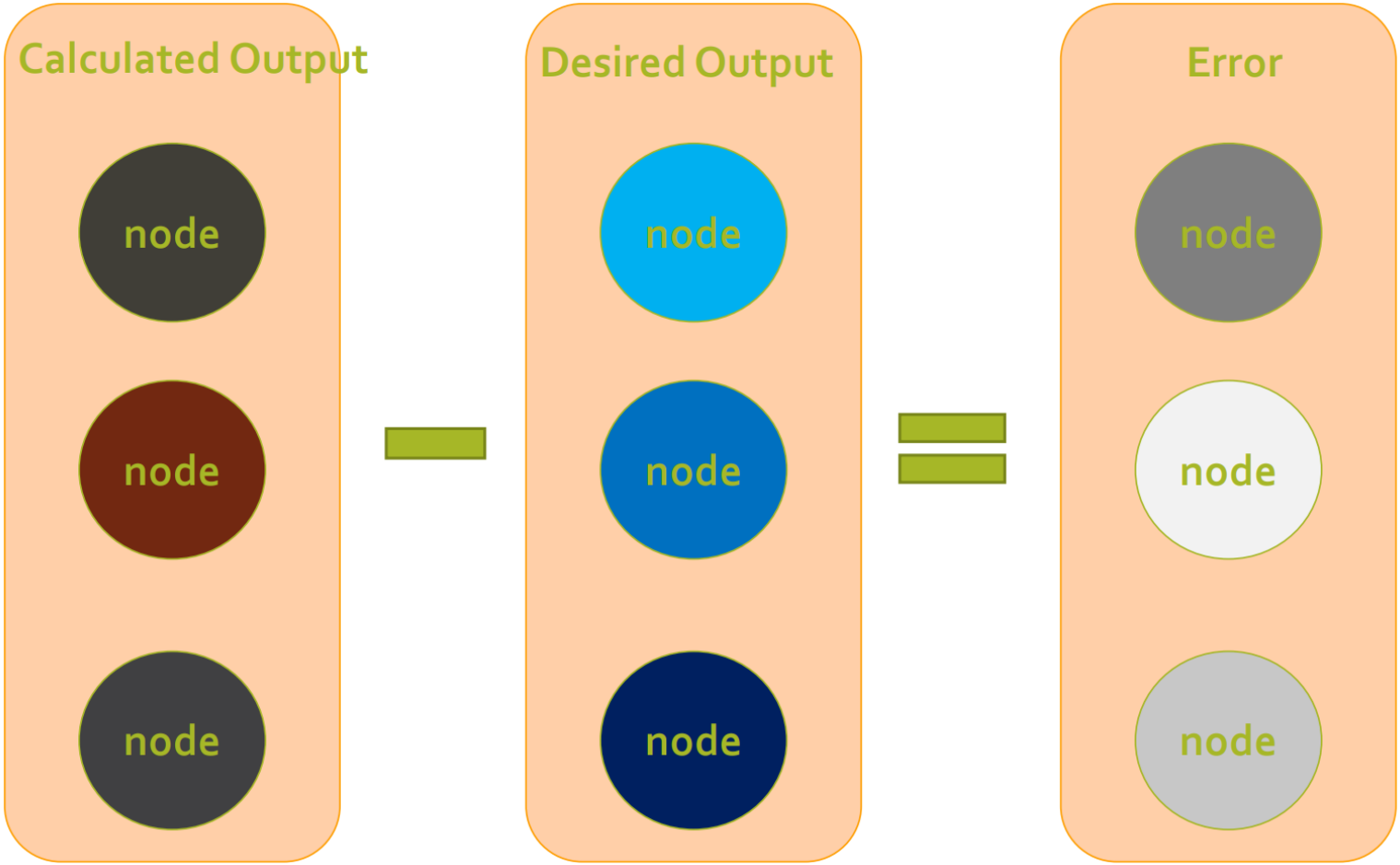
BACKPROPAGATION ALGORITHM



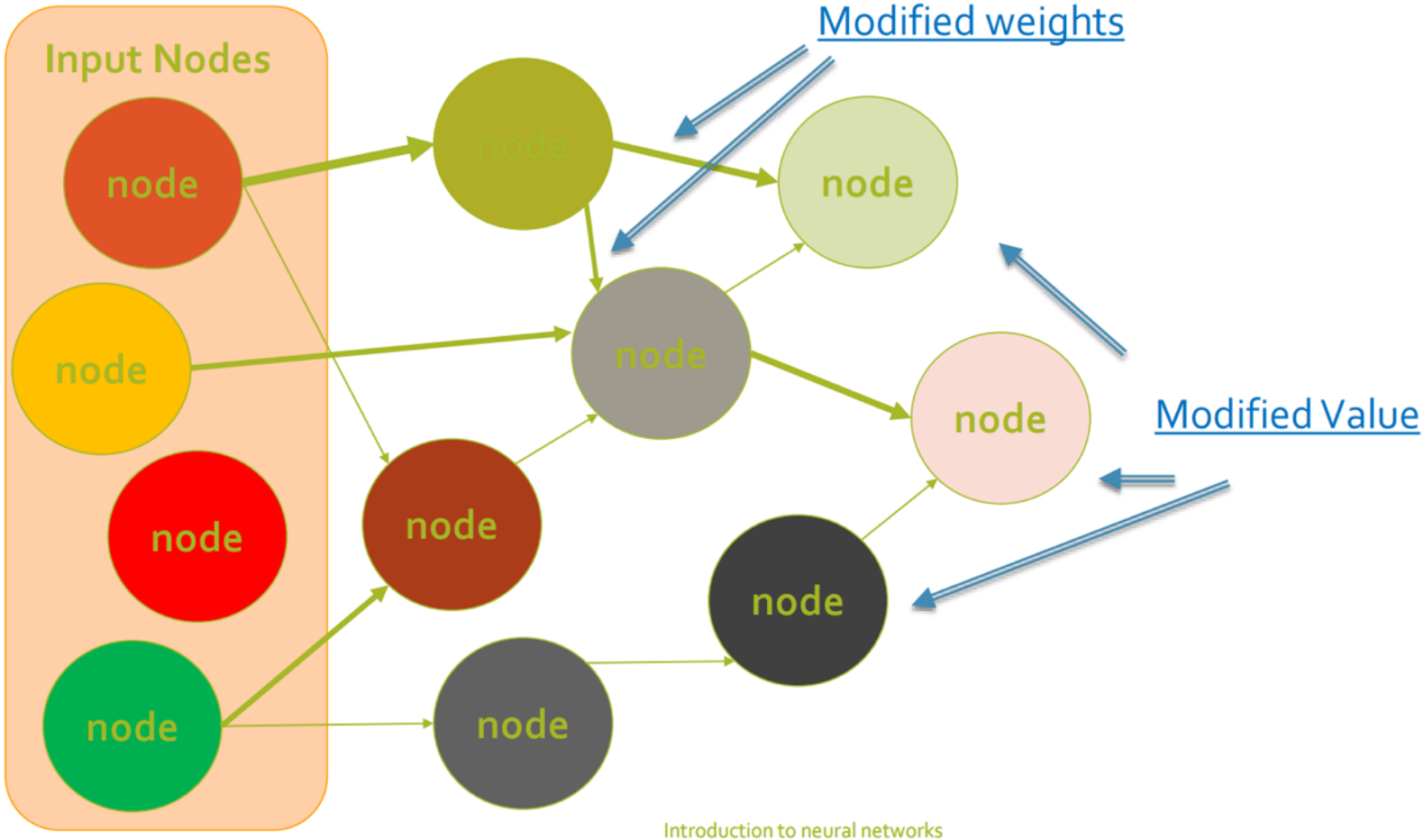
BACKPROPAGATION ALGORITHM



BACKPROPAGATION ALGORITHM



BACKPROPAGATION ALGORITHM



BACKPROPAGATION ALGORITHM

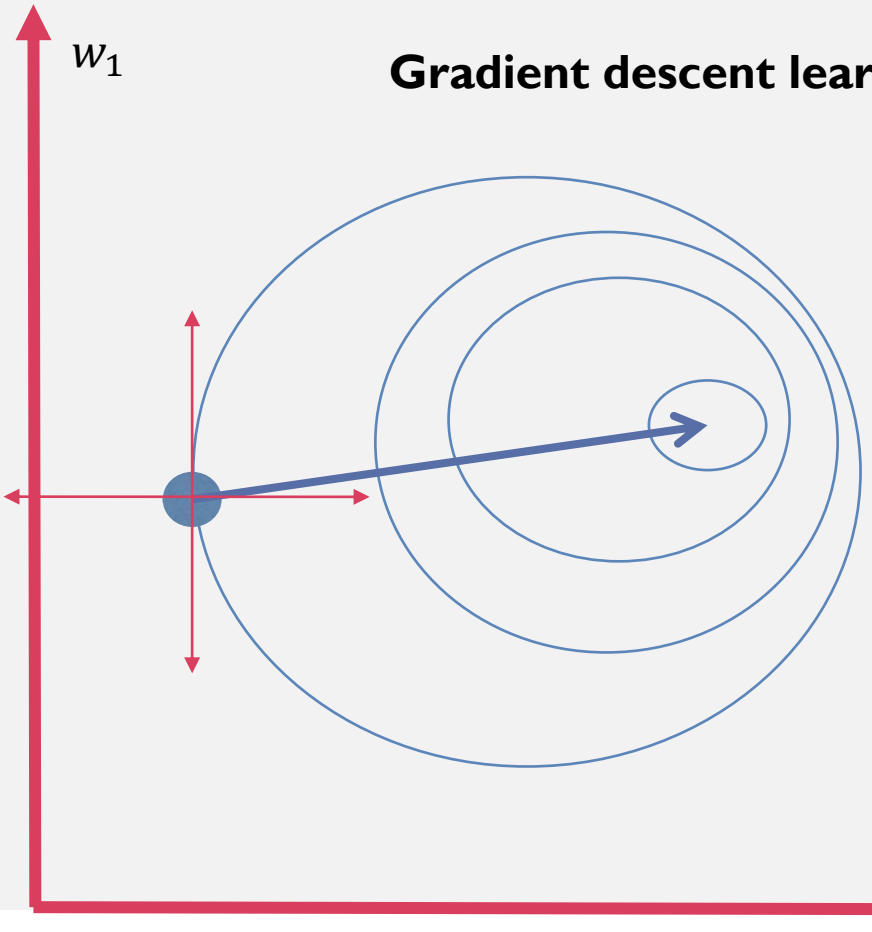
- Cost function, $C = \frac{1}{n}\sum_{i=1}^{n} C_i$
- Cost function for individual example, $C_i = \frac{1}{2}\left\|y_i - a_i^L\right\|^2$

L denotes the number of layers in the network.

a_i^L is the vector of activations output from the network of example i.

y_i is the desired output of example i.

BACKPROPAGATION ALGORITHM



- $\Delta\bar{w} = r\left(\frac{\partial C}{\partial w_1} i + \frac{\partial C}{\partial w_2} \mathrm{j}\right)$

r denotes **learning rate**.

https://www.youtube.com/watch?v=IHZwWFHWa-w

DETERMINING THE NUMBER OF HIDDEN LAYERS

- <= 2 hidden layers

Num Hidden Layers	Result
none	Only capable of representing linear separable functions or decisions.
1	Can approximate any function that contains a continuous mapping from one finite space to another.
2	Can represent an arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy.
>2	Additional layers can learn complex representations (sort of automatic feature engineering) for layer layers.

https://www.heatonresearch.com/2017/06/01/hidden-layers.html

DETERMINING THE NUMBER OF NEURONS IN THE HIDDEN LAYERS

- No method to find the most optimum num of neurons in the hidden layers
- Only be done by trial
- Rule of thumbs:
  - The number of hidden neurons should be **between the size of the input layer and the size of the output layer**.
  - The number of hidden neurons should be **2/3 the size of the input layer, plus the size of the output layer**.
  - The number of hidden neurons should be **less than twice the size of the input layer**.

https://www.heatonresearch.com/2017/06/01/hidden-layers.html

BACKPROPAGATION ALGORITHM

- https://teachablemachine.withgoogle.com/

DIFFERENT USES OF ANN

- **Classification**—ANN can be trained to **classify given pattern** or data set into predefined class.
- **Regression**—ANN can be trained to **produce outputs** that are expected from given input (e.g. stock market prediction).
- **Clustering**—ANN can be used to identify a **special feature** of the data and classify them into **different categories** without any **prior knowledge** of the data.

APPLICATIONS

- Image recognition
- Speech recognition
- Medical diagnosis
- Text mining
- Self-driving car
- Etc.

PRO AND CONS

- Pros
  - ANN is a powerful data-driven, self-adaptive, flexible computational tool.
  - ANN has the capability of capturing **nonlinear** and complex underlying characteristics of any physical process (e.g. damage detection) with a high degree of accuracy.
- Cons
  - However, they are generally regarded to behave as "**black-box**" systems.
  - Due to this, the user cannot explain how learning from input data was

EXAMPLE: PREDICTING THE SURVIVAL OF TITANIC PASSENGERS BY DEEP LEARNING (//SAMPLES/DATA/TITANIC)

- **RMS Titanic**
- *The RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early morning hours of 15 April 1912, after it collided with an iceberg during its maiden voyage from Southampton to New York City. There were an estimated 2,224 passengers and crew aboard the ship, and more than 1,500 died, making it one of the deadliest commercial peacetime maritime disasters in modern history. The RMS Titanic was the largest ship afloat at the time it entered service and was the second of three Olympic-class ocean liners operated by the White Star Line. The Titanic was built by the Harland and Wolff shipyard in Belfast. Thomas Andrews, her architect, died in the disaster.*

ANALYSIS STEPS:

1. Data Preparation & Cleaning
   - Select attributes
   - Deal with missing values
2. Data Visulization & Analysis
   - Transfer Numerical to Categorial data, if necessary
   - Set Label
3. Deep Learning Training:
   - Split Validation
   - Num. of layers/ Num. of nodes/ learning rate
4. Testing&Evaluation:
   - Confusion Matrix

SELECT NODES PER HIDDEN LAYERS:

Optimize Parameters (Grid) (100 rows, 4 columns)

iteration	Extract Macro_layer2.example_index	Extract Macro_layer1.example_index	accuracy ↓
96	6	10	0.840
35	5	4	0.827
75	5	8	0.824
94	4	10	0.824
83	3	9	0.821
37	7	4	0.821
32	2	4	0.812
10	10	1	0.808
97	7	10	0.808
26	6	3	0.805
70	10	7	0.805
2	2	1	0.805
84	4	9	0.805
56	6	6	0.805
18	8	2	0.805

HW10: APPLY DEEP LEARNING TO PREDICT IMDB_SCORE (LOW OR HIGH) ON IMDB

- movie_title : Title of the Movie
- duration: Duration in minutes
- director_name : Name of the Director of the Movie.
- director_facebook_likes : Number of likes of the Director on his Facebook Page.
- color: Film colorization. 'Black and White' or 'Color'
- genres: Film categorization like 'Animation', 'Comedy', 'Romance', 'Horror', 'Sci-Fi', 'Action', 'Family'
- actor_1_name: Primary actor starring in the movie
- actor_1_facebook_likes : Number of likes of the Actor_1 on his/her Facebook Page.
- actor_2_name: Other actor starring in the movie
- actor_2_facebook_likes : Number of likes of the Actor_2 on his/her Facebook Page.
- actor_3_name: Other actor starring in the movie
- actor_3_facebook_likes : Number of likes of the Actor_3 on his/her Facebook Page.
- num_critic_for_reviews : Number of critical reviews on imdb
- num_voted_users: Number of people who voted for the movie
- cast_total_facebook_likes: Total number of facebook likes of the entire cast of the movie.
- language : English, Arabic, Chinese, French, German, Danish, Italian, Japanese etc
- country: Country where the movie is produced.
- gross: Gross earnings of the movie in Dollars
- budget: Budget of the movie in Dollars
- title_year: The year in which the movie is released (1916:2016)
- imdb_score: IMDB Score of the movie on IMDB
- movie_facebook_likes: Number of Facebook likes in the movie page.

H10: SOLUTION STEPS

1. Data Preparation & Cleaning
   - Select attributes
   - Deal with missing values
2. Data Visulization & Analysis
   - Transfer Numerical to Categorial data, if necessary
   - Set Label
   - Select Attributes (takes more time for big attribute)
3. Deep Learning Training:
   - Split data or Cross Validation
   - Num. of layers/ Num. of nodes/ learning rate
4. Testing&Evaluation:
   - Confusion Matrix

REFERENCES

- McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics. 5 (4): 115–133.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. in Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations (eds Rumelhart, D. E. & McClelland, J. L.) 318–362 (MIT, Cambridge, 1986).
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533.
- https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/