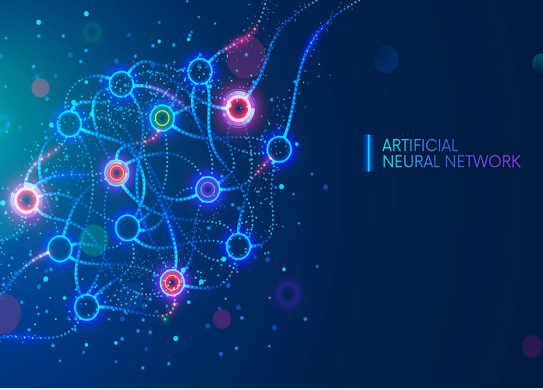




Neural Networks

Dr. Seena Joseph

Neural Network

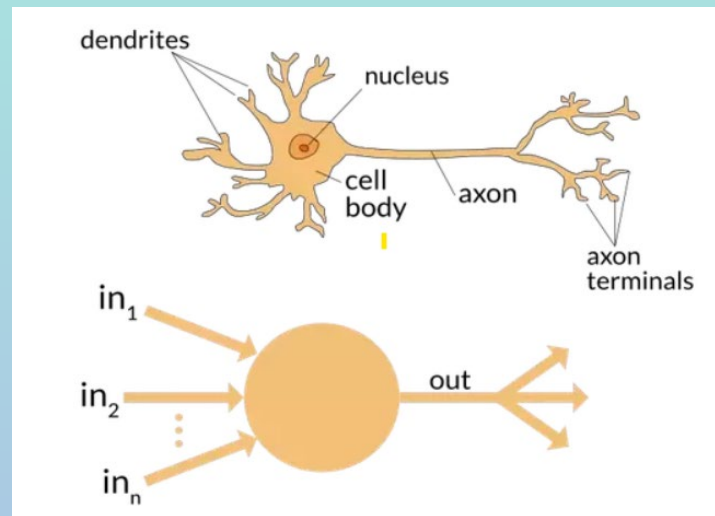
- Learning outcomes
 - Understanding the concept of Neural Networks including its basic structure and components.
 - Gain an understanding to build a feed-forward neural network for a classification task using Keras.

Neural Networks

- Introduction
- Key components of NN
- Steps in building the NN architecture
- Setting up the environment for building NN in Python
- Demonstration of the model implementation in Python – Case Study
- Conclusion

Neural Network

- Neural Network or **Artificial Neural Network (ANN)**: Computational model inspired by the function and structure of a biological neural network.



Neural Networks

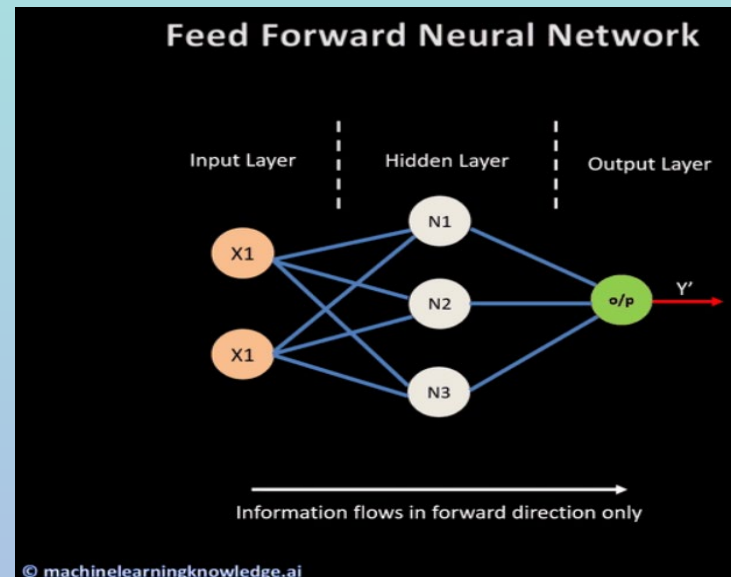
- *Learn it* from the data
 - Acquires knowledge and improves its performance by analysing and processing input data
- *Application* of NN
 - Image classification: Image recognition, object detection, facial recognition
 - Natural language processing: Language translation, text generation, chatbots
 - Speech recognition: voice assistance and transcription services
 - Recommendation system: streaming platforms, personalized content recommendation
 - Medical diagnosis
 - Financial forecasting
 - Autonomous Vehicles
 -

Neural networks

- Types of Neural Networks
 - Feed forward Neural Network (FNN)
 - Convolutional Neural Network (CNN)
 - Recurrent Neural Network (RNN)
 - Long Short Term Memory (LSTM)
 - Auto encoders
 - Generative Adversarial Network (GAN)
 - ...

Neural networks

- Feed Forward Neural Network
 - This is the most basic neural network architecture, the building block for most other neural networks.
 - Information flows in one direction, from the input layer to the output layer. The connections between nodes do not form a cycle.



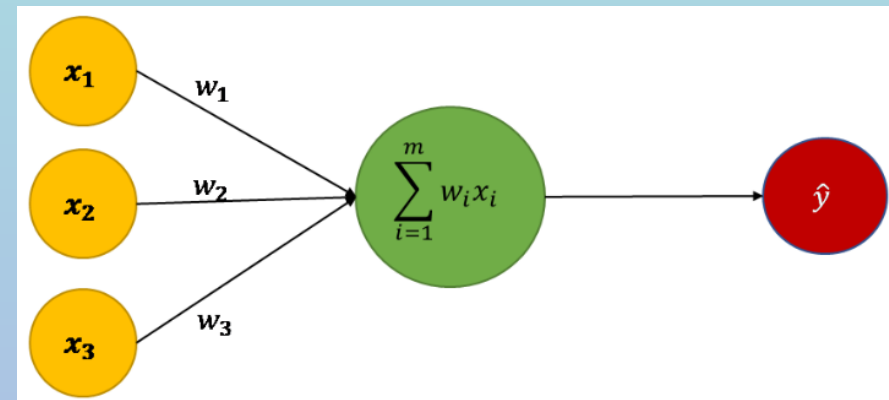
Building blocks of Neural networks

Neuron – Fundamental building block.

- Receive input signal -> multiply by weights -> Weighted sum of inputs -> apply activation function -> Output

A neuron in a neural network works in three steps:

- Multiply the input signal with the corresponding weights
- Obtaining a weighted sum of the inputs
 - $Z = x_1 * W_1 + x_2 * W_2 + x_3 * W_3$
- Apply an activation function (introduce non-linearity)



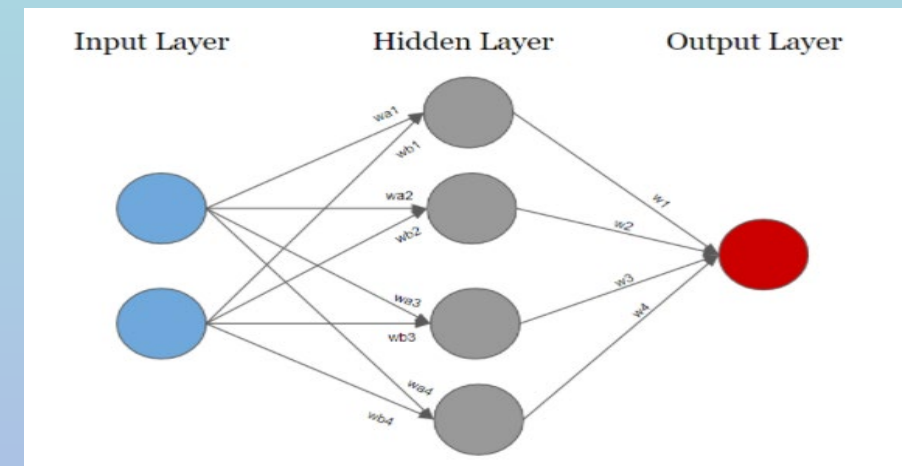
Building blocks of Neural networks

- **Layers**

- Input layer – Vector that describes the feature of the input data (passive)
- Hidden layer – Intermediate layer between the input and output layers
 - Processing and transforming the input data before producing the output
- Output layer – Produces final output

- **Weights** and **biases**

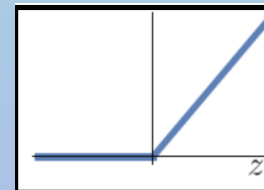
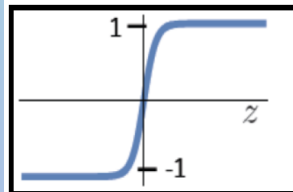
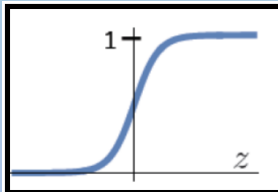
- Strength of the connection between the neuron
 - Numerical Value
 - Determines the impact of input on the output of the connected neuron
-
- Biases are additional parameters
 - Each neuron (Except the input layer neuron) has an associated bias
 - Helps in producing a non—zero output when input features are zero



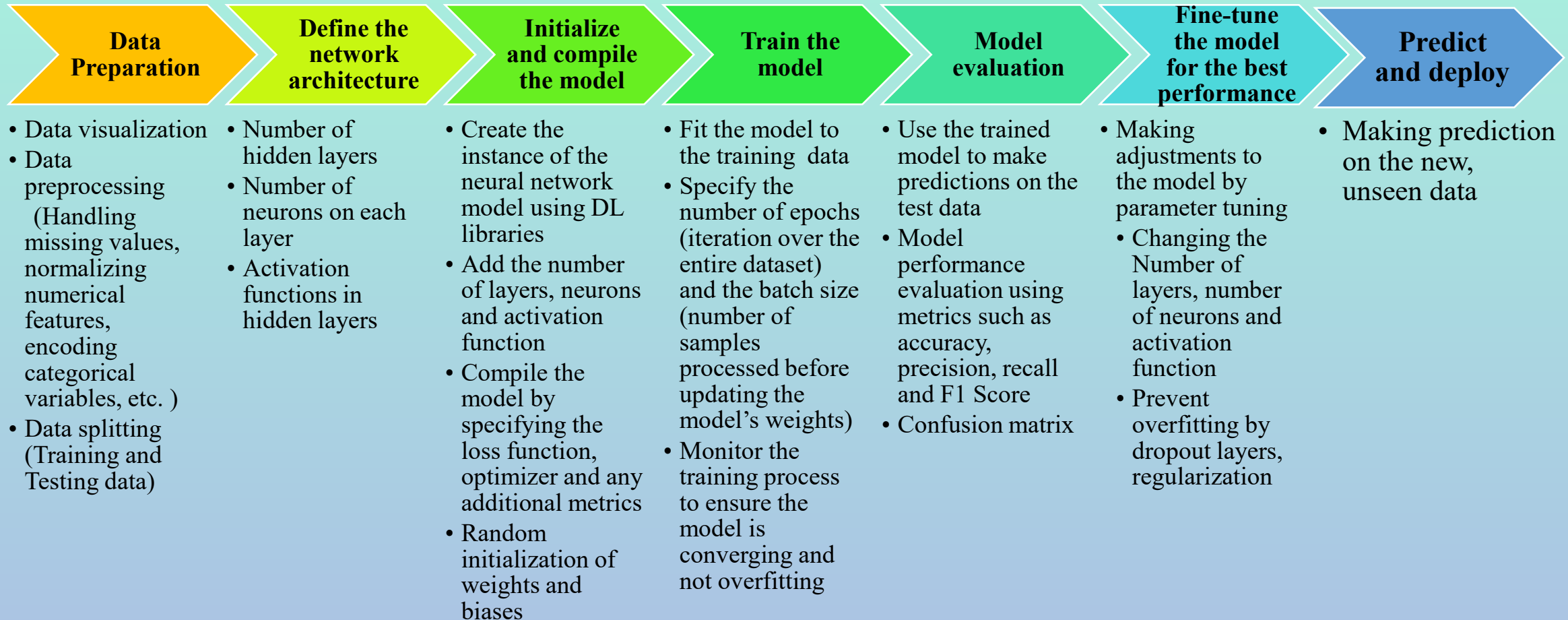
Building blocks of Neural networks

- **Activation function**

- Introduce nonlinearity, and determine the output of the neuron or layers based on the weighted sum of inputs.
- Common types of activation function
 - Sigmoid Function – Most commonly used in the **output layer** for **binary classification**
 - Output Range **(0,1)**
 - Tanh (Tangent Hyperbolic Function) – Mathematically shifted version of Sigmoid
 - Preferred over sigmoid in **hidden layers**
 - Output Range **(-1,1)**
 - ReLU (Rectified Linear Unit) – Mainly used in hidden layers
 - Output Range **(0, ∞)**
 - Softmax - Most commonly used in the output layer for **multiclass classification**
 - Output Range **(0,1)**



Building the NN architecture



Setting up the environment for building an NN in Python

- Why Python
 - Easy-to-Understand Syntax
 - Rich Ecosystem of Libraries and Frameworks (Specifically designed for ML and DL tasks)
 - Numerical computing capabilities
- Install
 - Numpy: Numerical computation
 - Pandas: Data manipulation and analysis
 - Seaborn: data visualization
 - TensorFlow: a Deep learning framework
 - Keras: A high-level neural networks API integrated within TensorFlow
 - scikit-learn: A comprehensive machine learning library that includes various tools for preprocessing data, model selection, and evaluation.

Case-study: Predicting Chances of admission

Context

- The world is fast evolving and constantly seeking the best expertise and experience among people. This inspires people all over the world to excel in their work and pursue higher education degrees that will help them improve their skills and knowledge. As a result, the number of students applying for Master's programs has significantly increased.
- The present admission dataset was developed to **forecast admissions** to the University of ABC. It was created to assist students in narrowing down their university choices based on their profiles. The expected production offers them a good sense of their chances of acceptance.
- **Objective:** to build a neural network-based classification model to estimate a student's chances of admission to ABC University.

Case-study: Predicting Chances of admission

- Implementation in Python
- Predicting Admission Chance - Jupyter Notebook

Conclusion

- We gained a basic understanding of a NN
- We have learned how to build a feed-forward neural network for a classification task using Keras.
- We have seen different hyper-parameters and how they affect the network.
- We were able to develop a model with an accuracy of 95% to achieve the objective stated for the given context.
- I strongly advise you to experiment with the other hyper-parameters to see how they affect the model.

Acknowledgement

- Data set and Information gathered – MIT IDSS Course on Data Science and ML