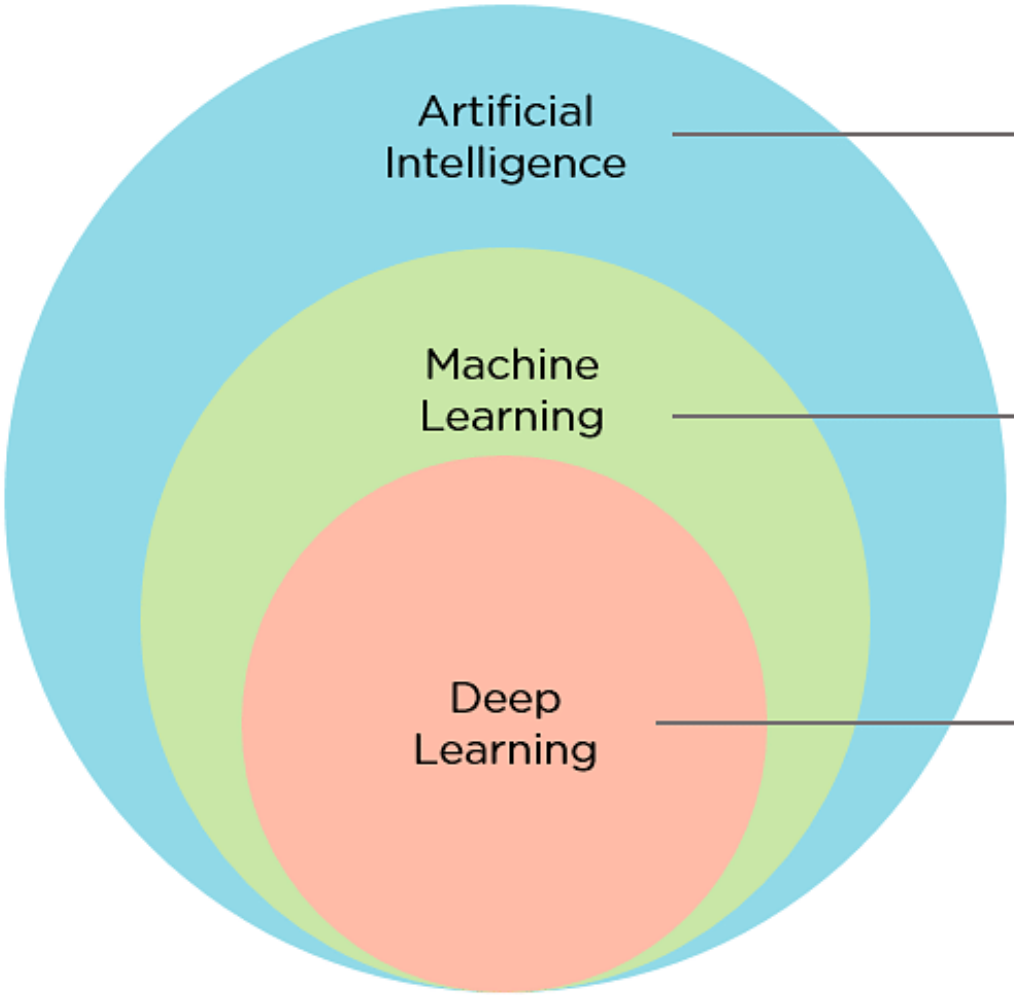

11001
DEEP
LEARNING
WWW

Agenda

- Introduction to Deep Learning
- What is Deep Learning?
- Basics of neural networks
- Introduction to Deep learning frameworks

Agenda

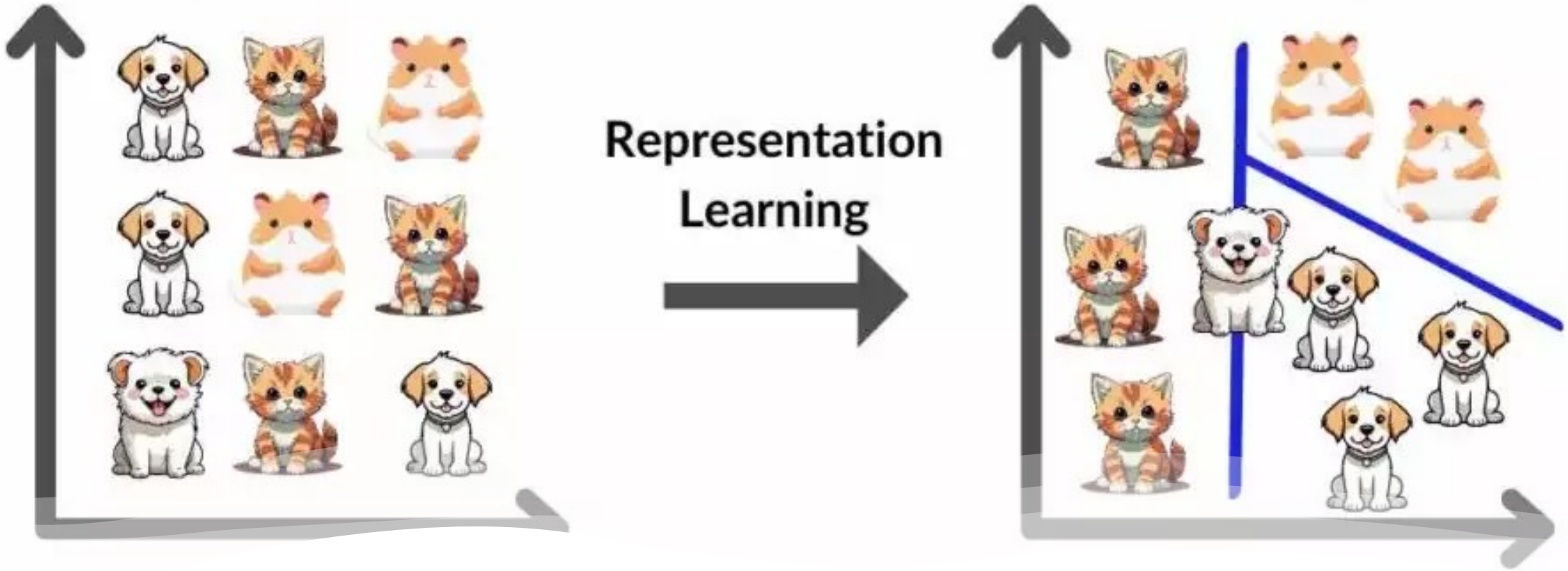
- TensorFlow Introduction
- Tensorflow Python Example
- Keras Introduction
- Keras Python Example
- **Summary and Conclusion**



Ability of a machine to imitate intelligent human behavior

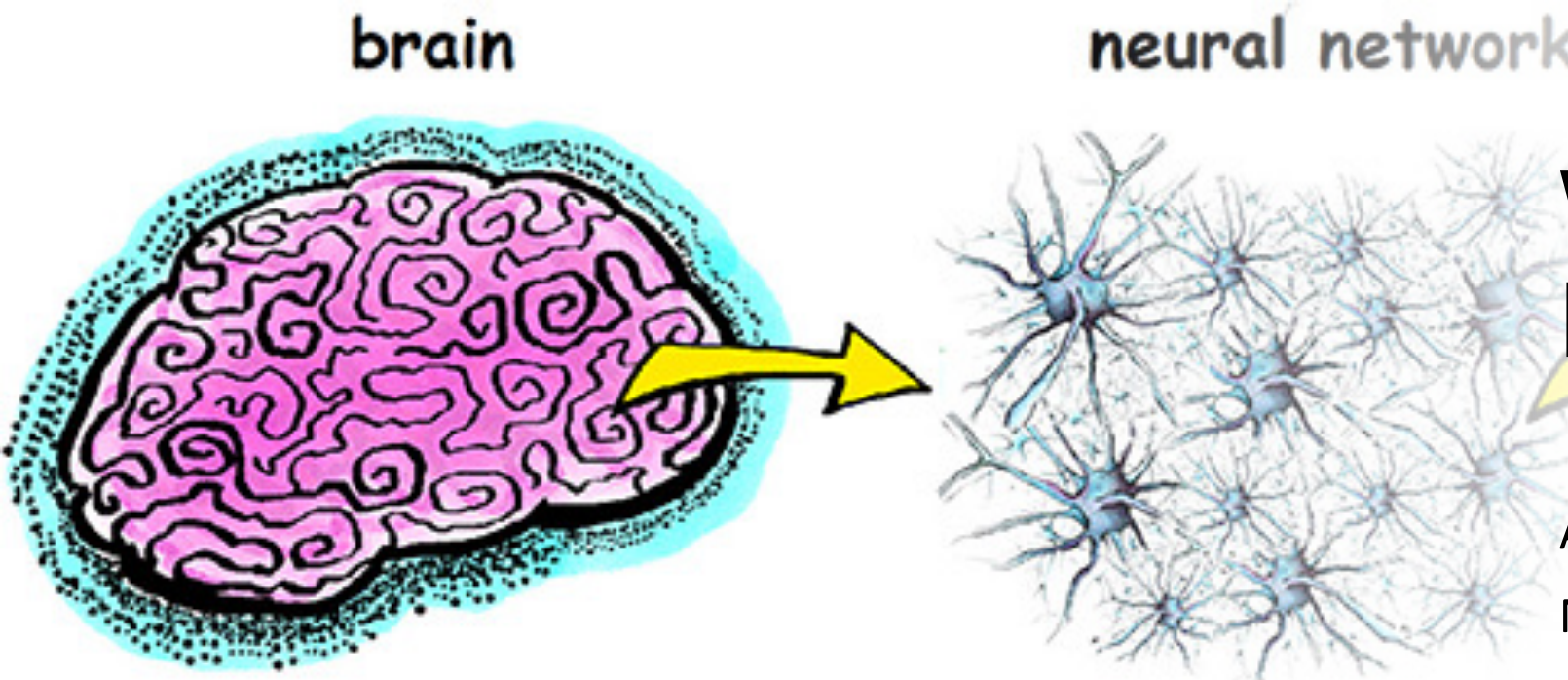
Application of AI that allows a system to automatically learn and improve from experience

Application of Machine Learning that uses complex algorithms and deep neural nets to train a model



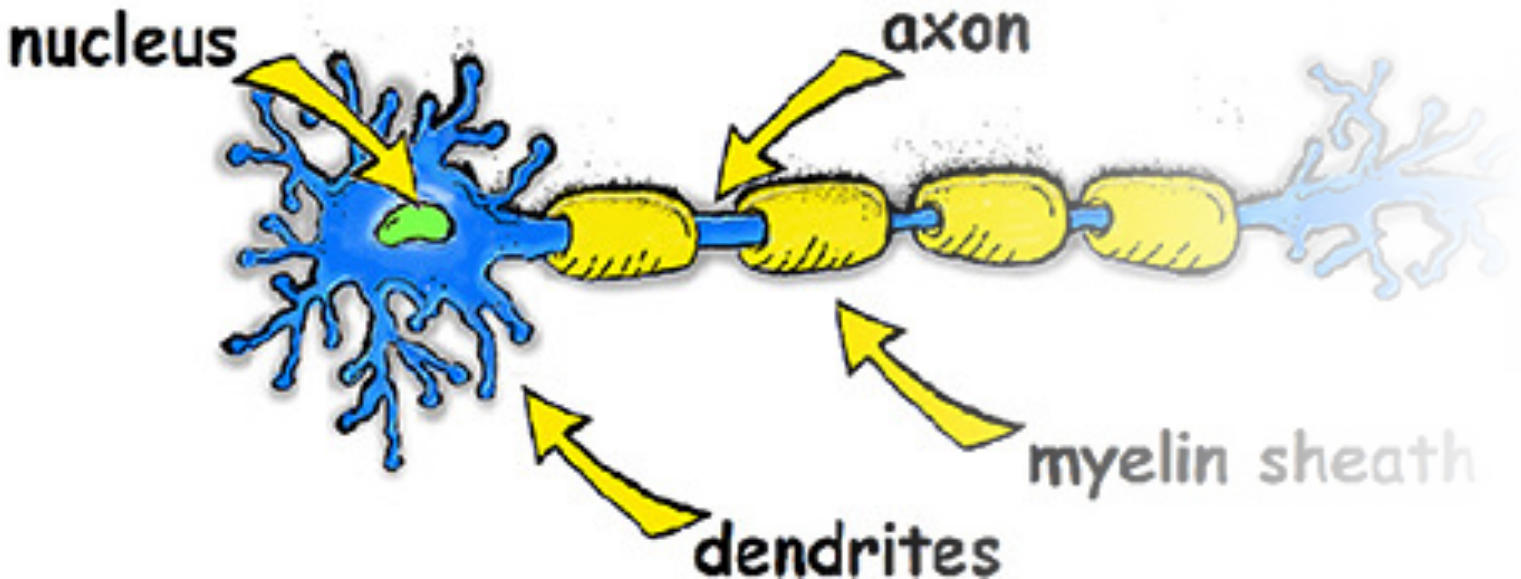
What is Deep Learning?

- Subset of machine learning
- Based on Artificial Neural Networks
- with representation learning.



What is Deep Learning?

A computer software Mimics Network of Neurons in a brain.



What is Deep Learning?

For Data Scientists

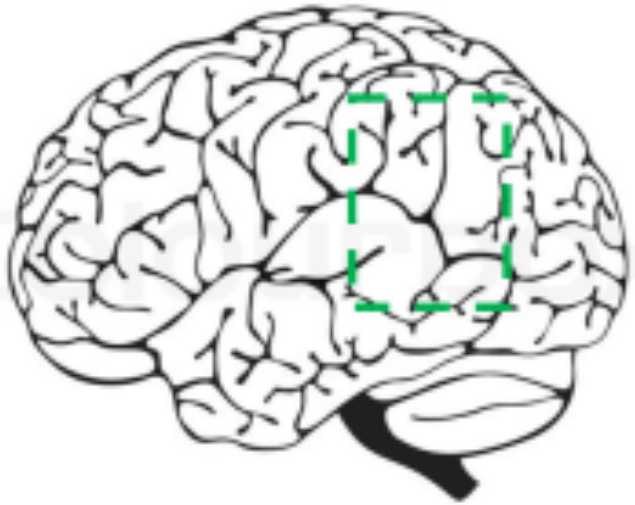
Deep learning is indispensable

Facilitates and accelerates

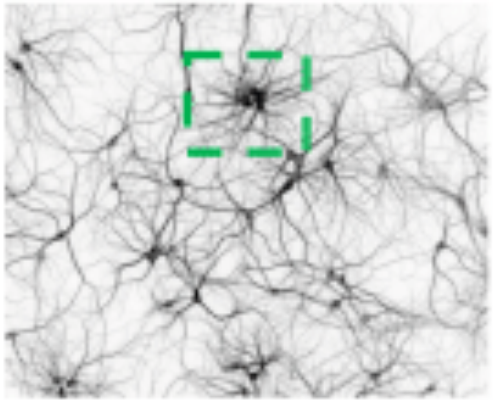
Collecting, Analyzing

Processing of large amounts of data.

What is Deep Learning?

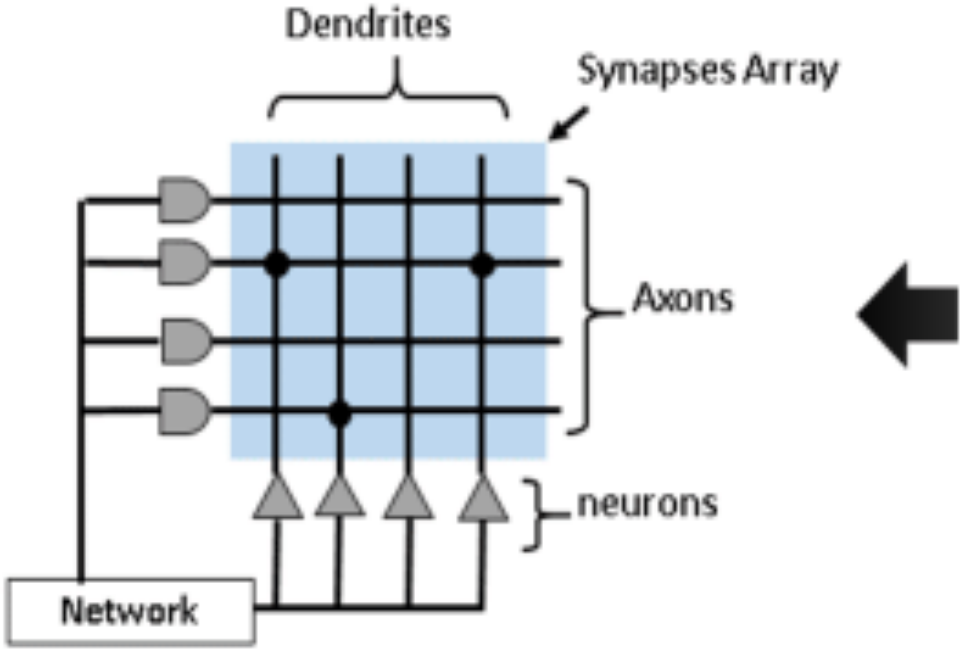


(a) Brain



(b) Neuron network

- Makes use of Deep Neural Networks.
- Learning can be supervised,
- Semi-supervised or
- Unsupervised.

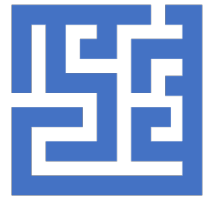


(e) Neuron Network

What is Deep Learning?

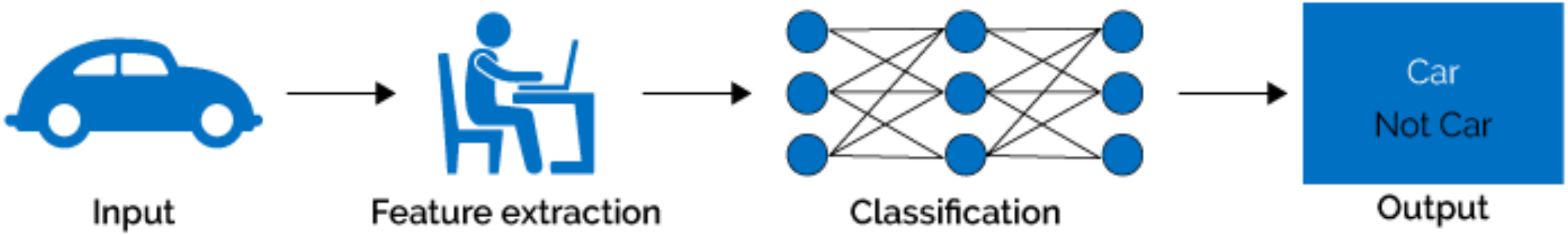
A subset of machine learning

Use of artificial neural networks



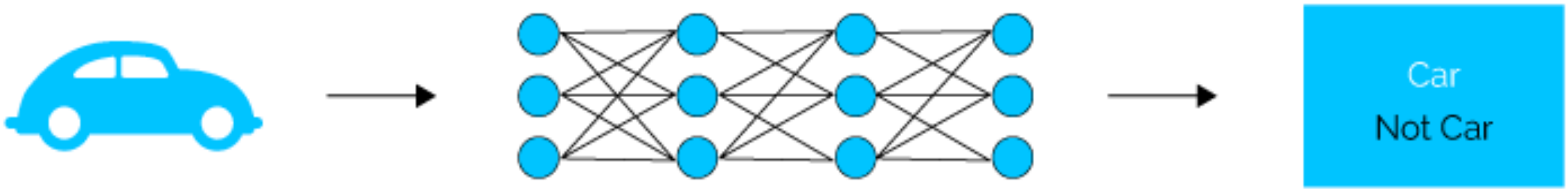
to model and solve complex problems.

Machine Learning



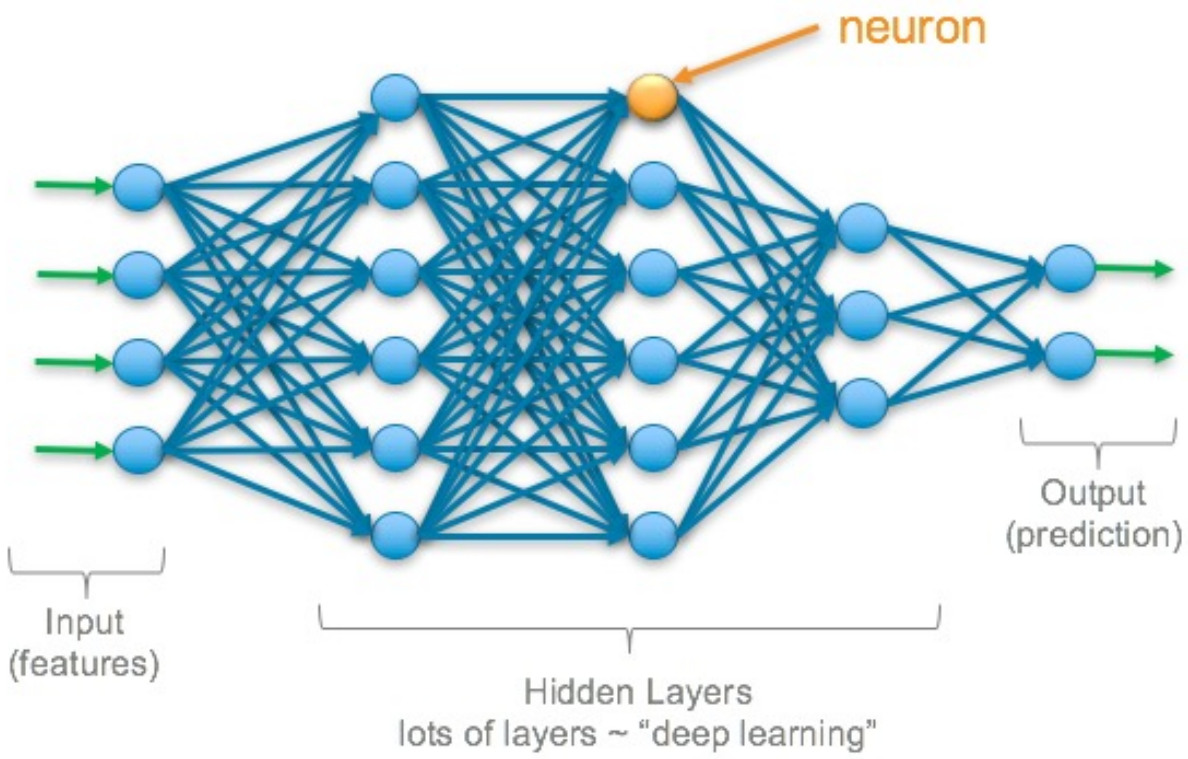
Input | Feature extraction | Classification | Output

Deep Learning



Input | Feature extraction + Classification | Output

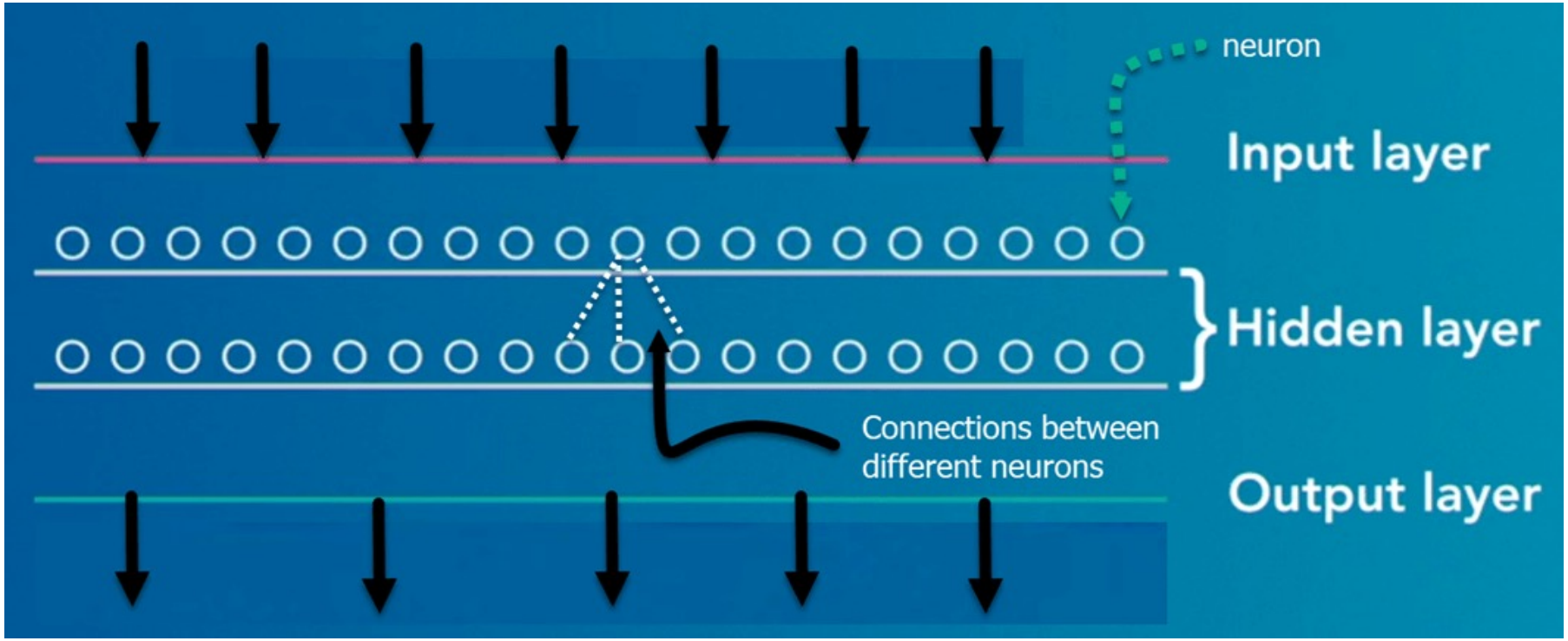
What is Deep Learning?



- DL Algorithms are constructed with connected layers.
- First Layer / Input Layer
- Last layer / Output Layer
- Middle Layer / Hidden Layers.

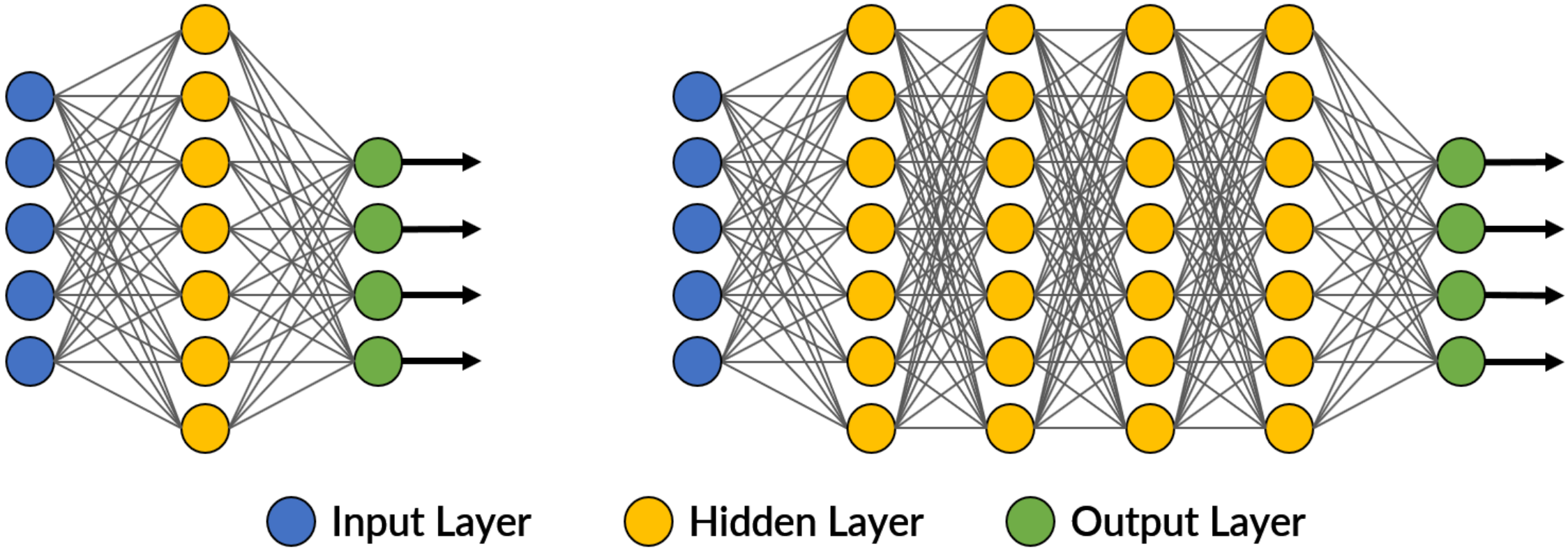
What is Deep Learning?

- The word deep means
- the network join neurons in
- more than two layers.

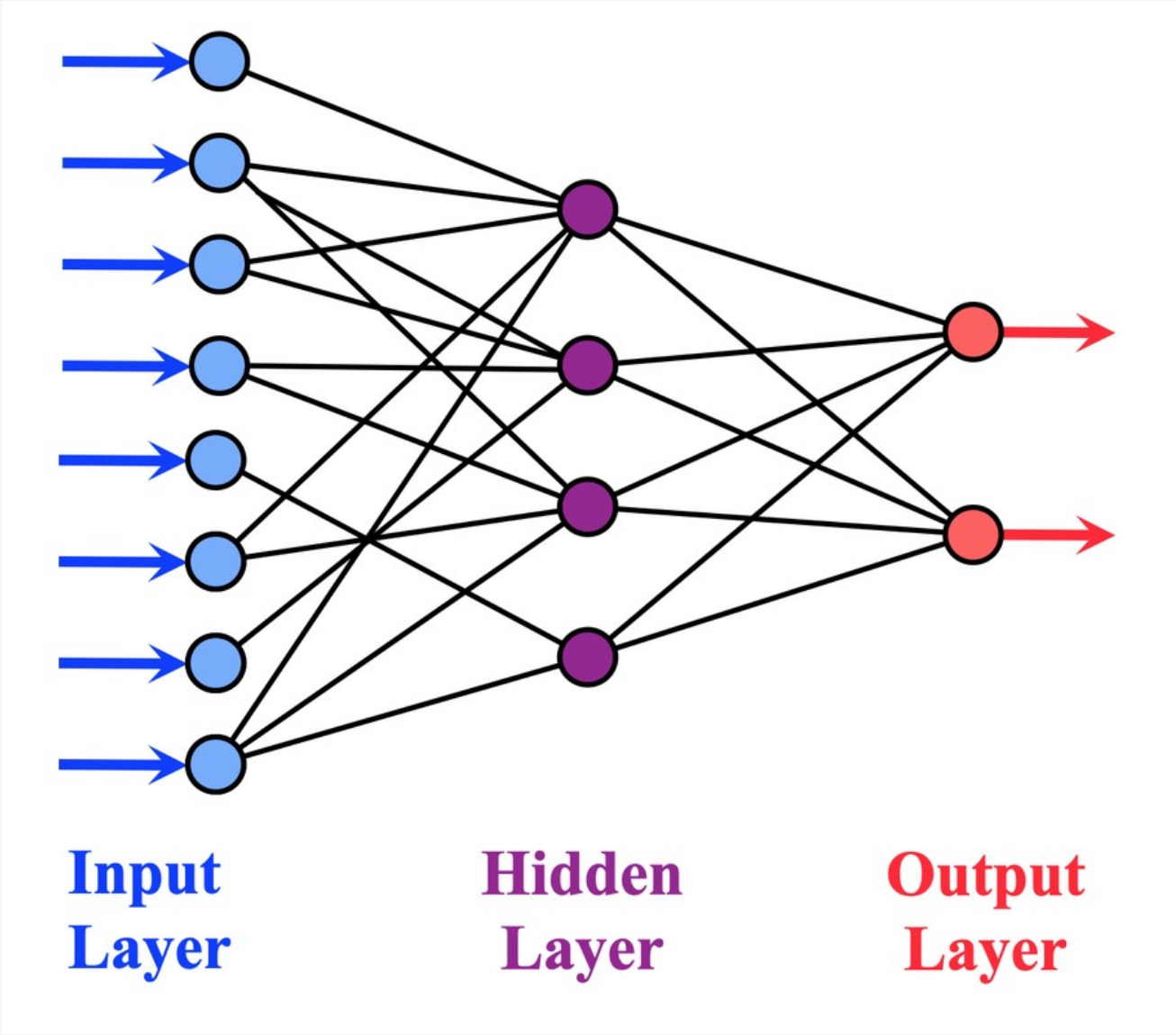


Simple Neural Network

Deep Learning Neural Network



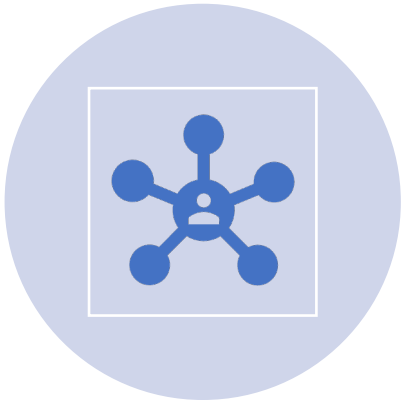
Input Layer Hidden Layer Output Layer



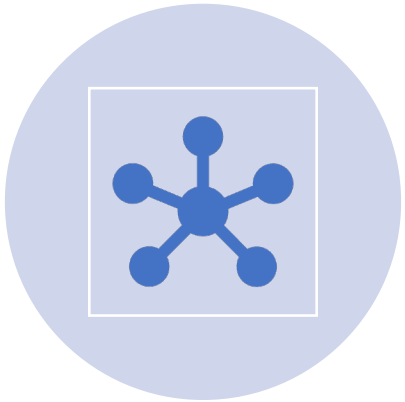
What is Deep Learning?

- NN Consumes
- Large amounts of input data
- Operates them through
- Multiple layers

What is Deep Learning?

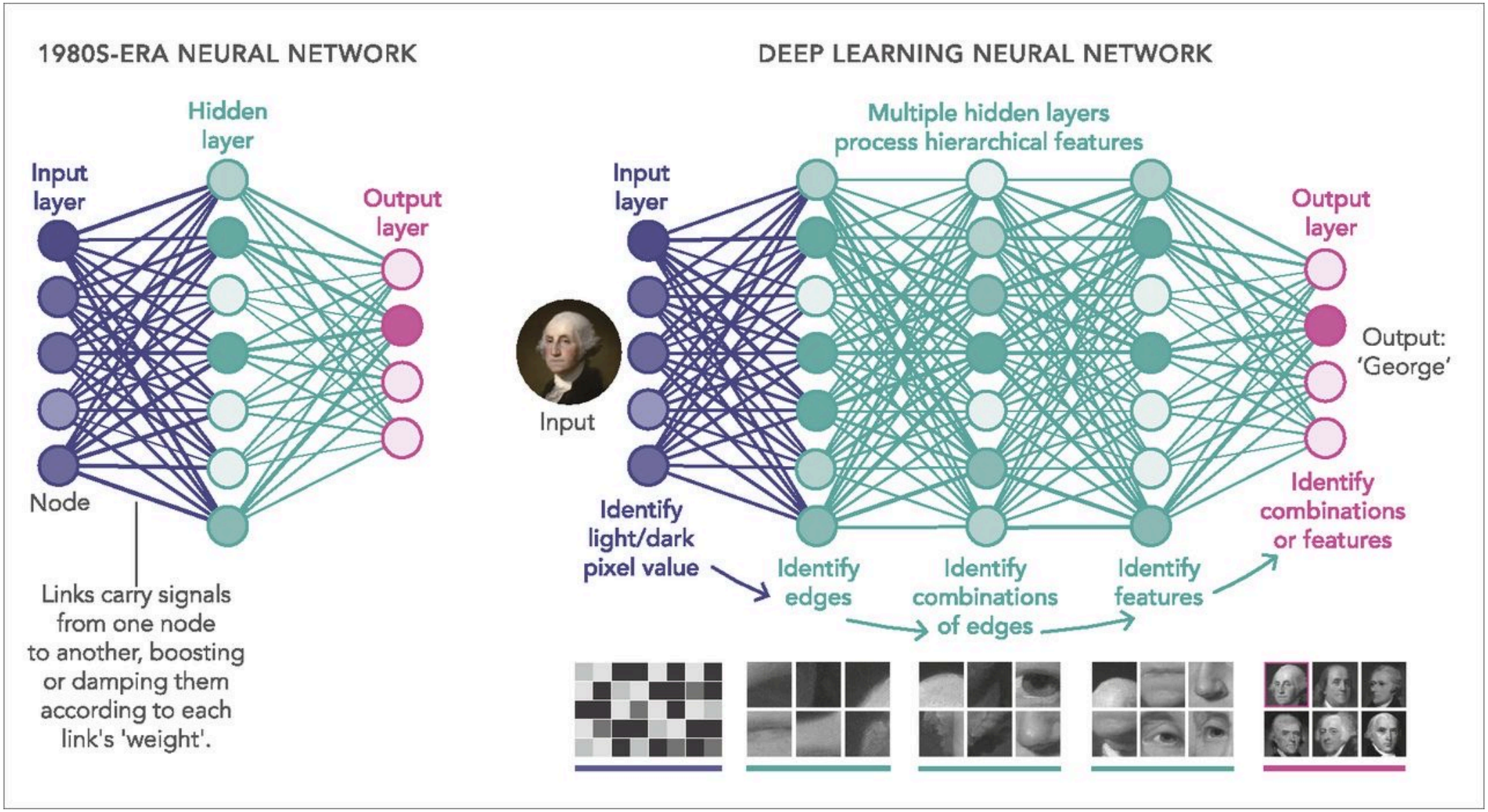


THE NETWORK CAN LEARN

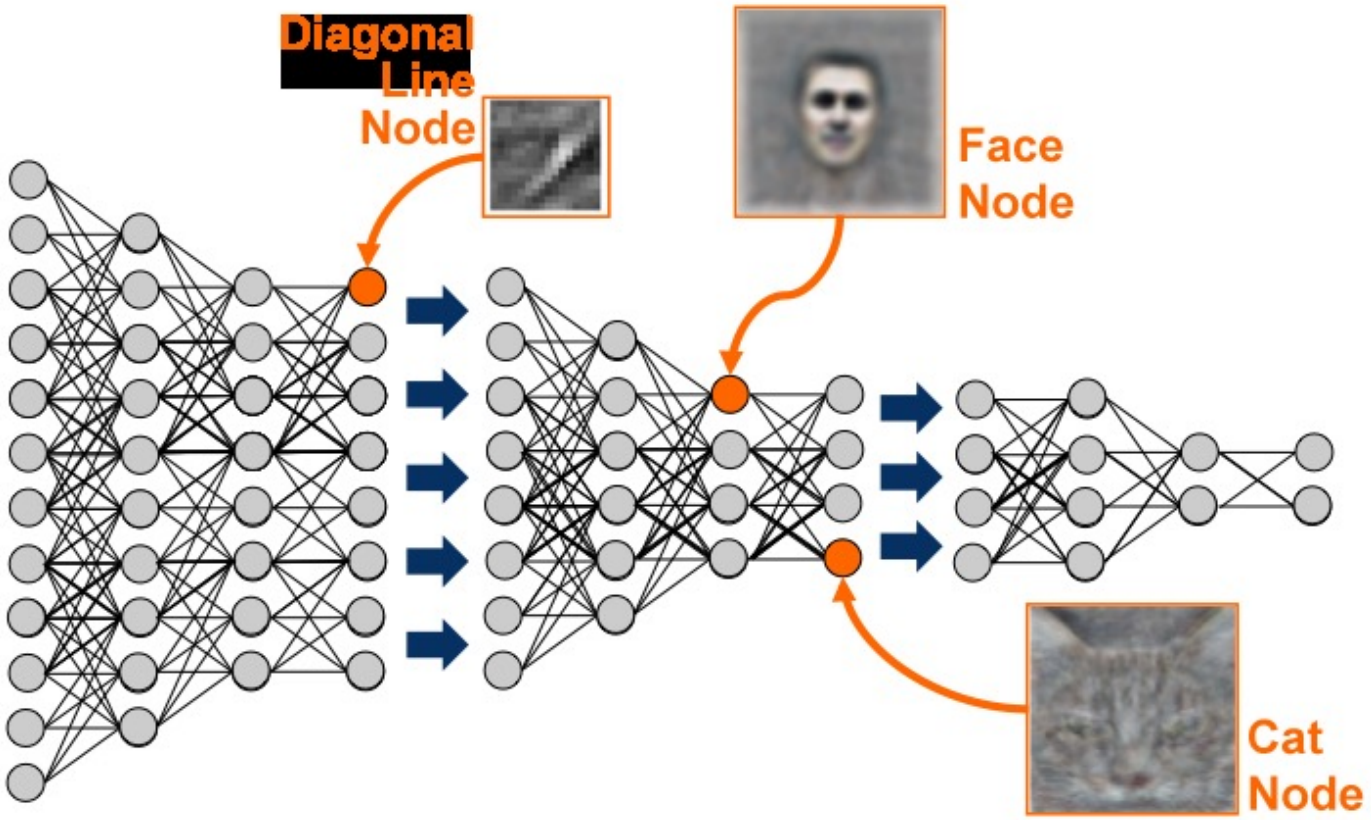


INCREASINGLY COMPLEX FEATURES

OF THE DATA AT EACH LAYER.


1980S-ERA NEURAL NETWORK
Hidden layer
Input layer
Output layer
Node
Links carry signals from one node to another, boosting or damping them according to each link's 'weight'.
DEEP LEARNING NEURAL NETWORK
Multiple hidden layers process hierarchical features
Input layer
Output layer
Input
Output: 'George'
Identify light/dark pixel value
Identify edges
Identify combinations of edges
Identify features
Identify combinations or features

Deep Neural Network



- Provides state-of-the-art
- accuracy in many tasks,
- from object detection
- to speech recognition

Deep Learning Process

DEEP LEARNING IS A CLASS OF MACHINE LEARNING ALGORITHMS THAT USES MULTIPLE LAYERS TO PROGRESSIVELY EXTRACT HIGHER-LEVEL FEATURES FROM THE RAW INPUT.

STEP 1

Understand the Problem

STEP 2

Identify Data

STEP 3

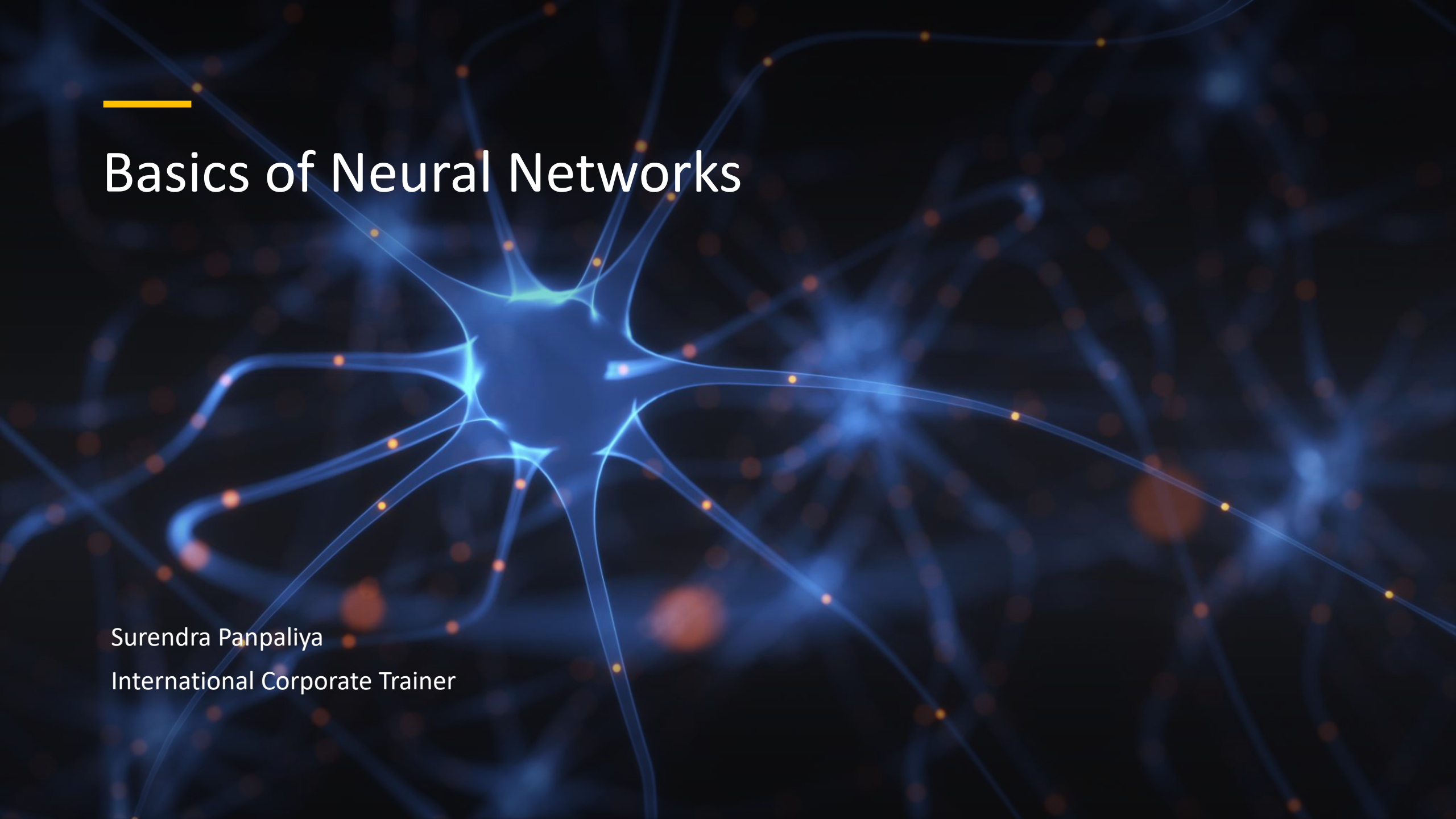
Select Deep Learning ALgorithms

STEP 4

Training the Model

STEP 5

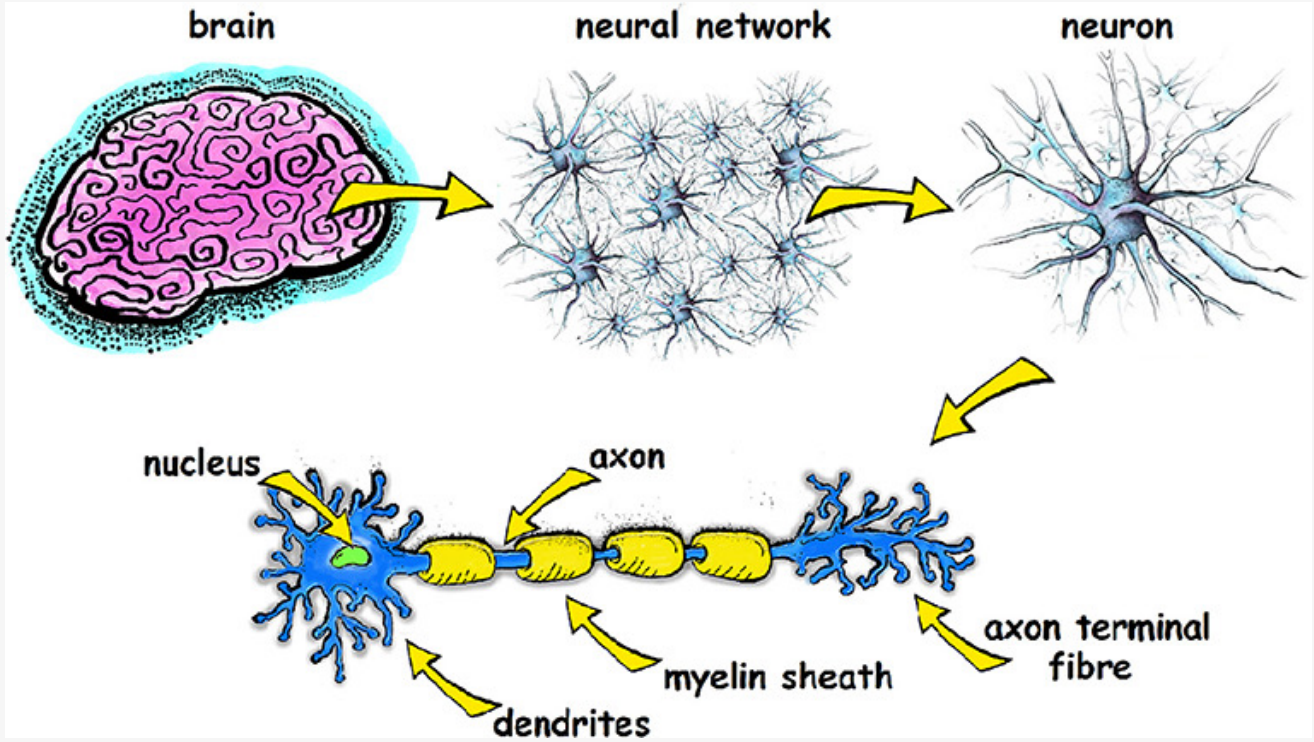
Test the Model



Basics of Neural Networks

Surendra Panpaliya

International Corporate Trainer



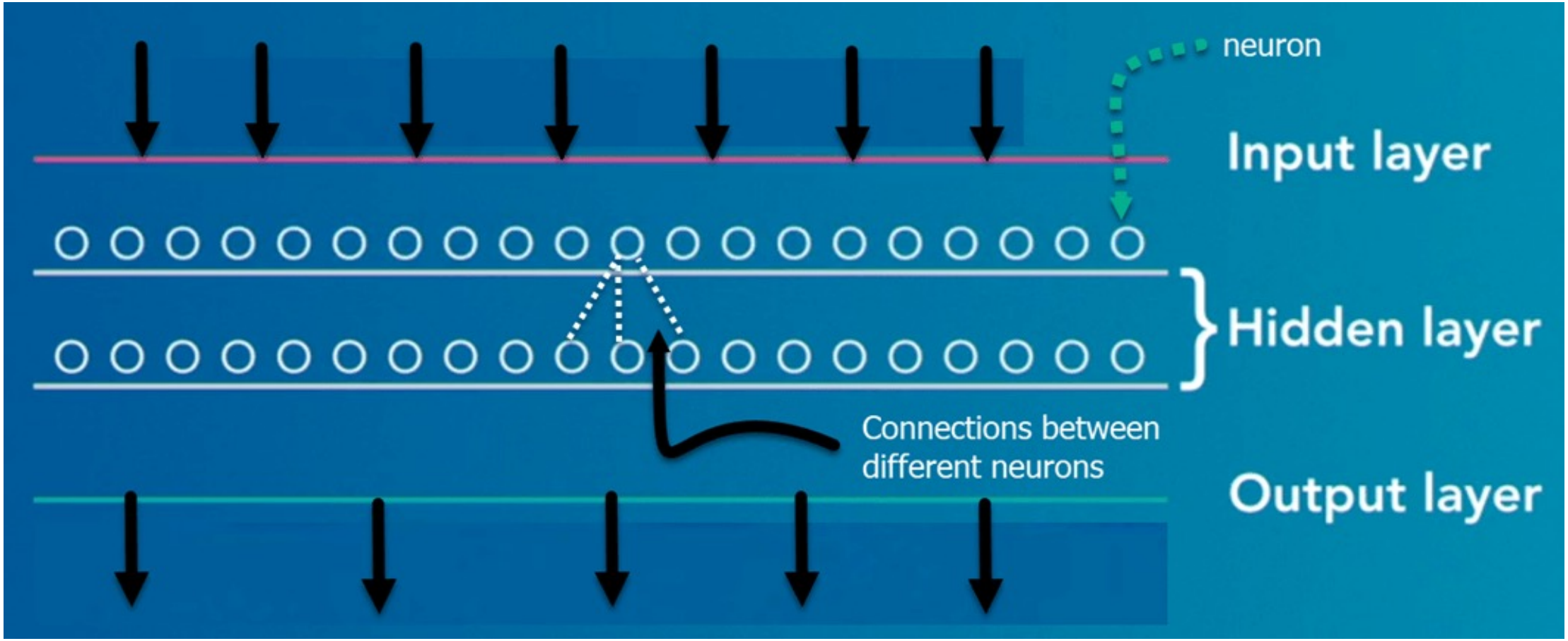
1. Neurons

Each neuron receives input,

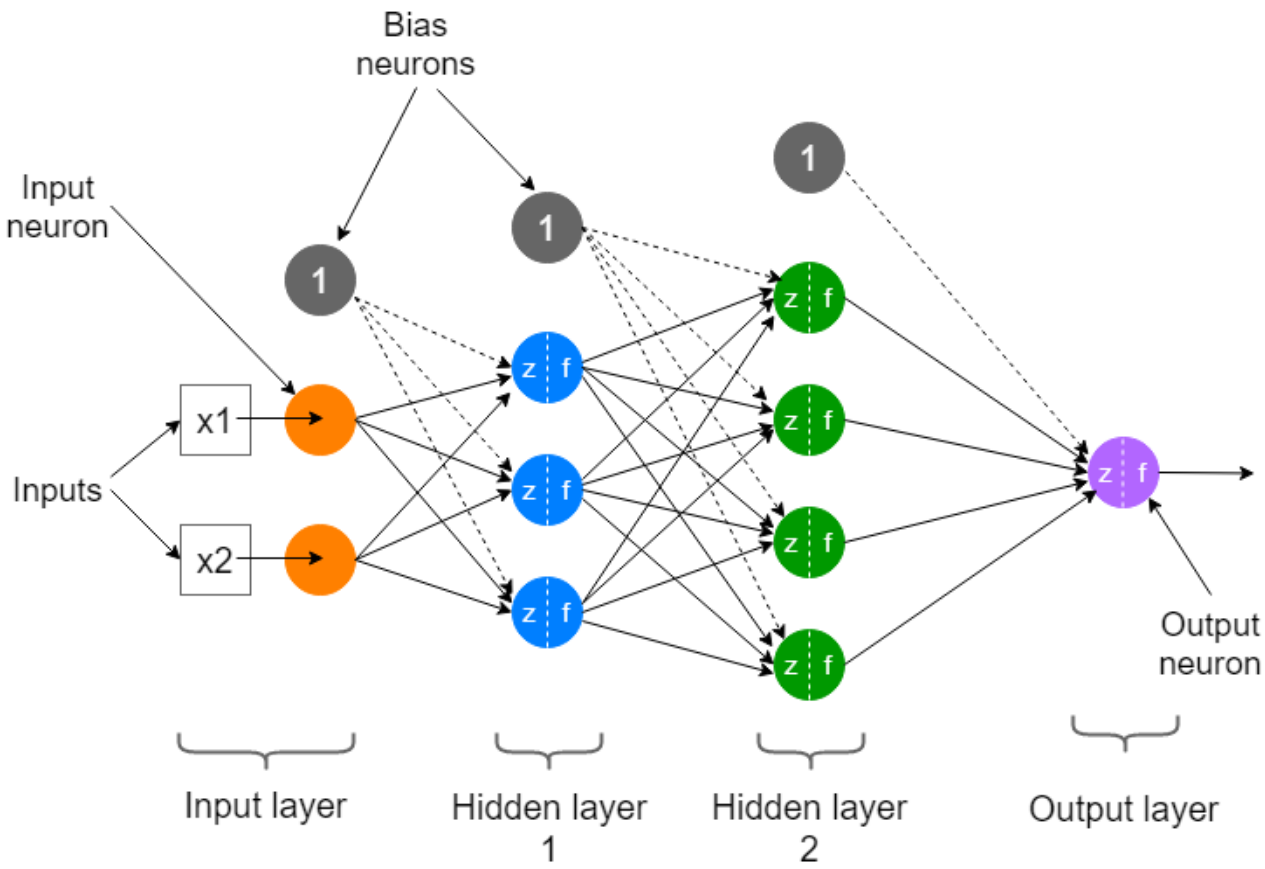
Performs a computation

Produces an output.

2. Layers


neuron
Input layer
Hidden layer
Connections between
different neurons
Output layer

2. Layers

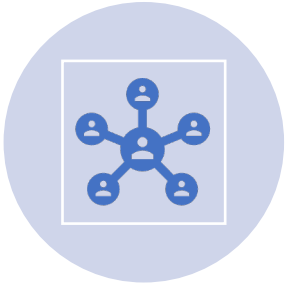


- Neurons in hidden layers
- Perform computations
- Learn representations from the data.
- Produces the final output of the network.

3. Connections and Weights

Neurons are connected to neurons

in the next layer



through connections,

each associated with a weight.

3. Connections and Weights

The weights represent

the strength of the connections

Learned during the training process.

- **ReLU (Rectified Linear Unit):** $f(x) = \max(0, x)$
- **Sigmoid:** $f(x) = \frac{1}{1+e^{-x}}$
- **Tanh:** $f(x) = \frac{e^{2x}-1}{e^{2x}+1}$

4. Activation Function

Each neuron has an activation function

Determines its output

Based on the weighted sum of its inputs.

5. Feedforward Process

Input data is passed through the network

Computations are performed layer by layer

Until the final output is produced.

This process transforms input data into a prediction.

6. Loss Function

Measures difference between

the predicted output

the actual target.

Quantifies the error of

the model's predictions

7. Backpropagation

- An optimization algorithm
- used to train neural networks.
- involves adjusting the weights
- based on the error (loss)
- calculated during the feedforward process.

7. Backpropagation

This process is iteratively performed

to minimize the loss and

improve the model's performance.

8. Training Data

Neural networks require

Labelled training data

To learn and generalize patterns.

Consists of input features

Corresponding target labels.

9. Gradient Descent

An optimization algorithm

used in the backpropagation process

9. Gradient Descent

It adjusts weights in the direction that

minimizes the loss function,

gradually converging towards

the optimal set of weights.

10. Epochs and Batch Size

Training is typically done in epochs,

where one epoch is a complete pass

through the entire training dataset.

10. Epochs and Batch Size

Batch size determines

the number of samples processed

before updating the model's weights.

11. Hyperparameters

- Such as the learning rate
- the number of hidden layers
- the number of neurons in each layer,
- are set before training
- influence the model's learning process.

12. Overfitting and Regularization

Neural networks may overfit the training data,

capturing noise and reducing generalization.

Techniques like dropout and

regularization are used

to mitigate overfitting.

13. Validation and Testing

The model's performance is assessed

on a separate validation set during training

final evaluation is done

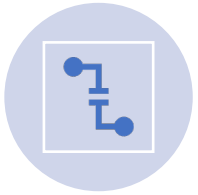
on a test set not seen during training.

Summary of Neural Networks

Neural networks can take

Various architectures

feedforward, convolutional, and



recurrent networks,

depending on the nature of

the data and the problem at hand

Summary of Neural Networks

Understanding Neural networks

crucial for working with

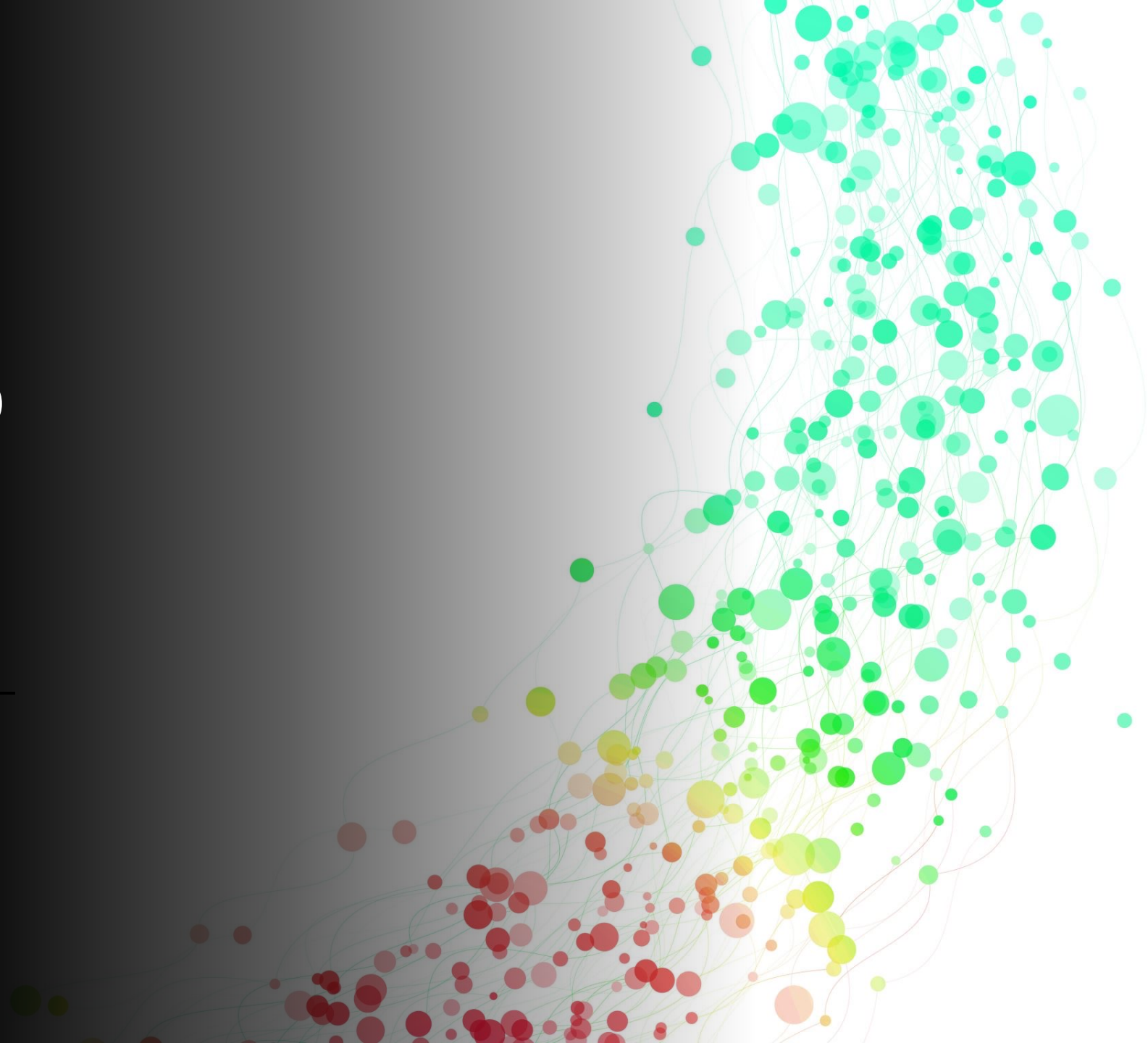
Deep learning models

developing solutions for a

wide range of applications.

Introduction to Deep learning frameworks

Surendra Panpaliya



Deep learning frameworks

Software libraries or tools

Provide a set of pre-built functions

Modules to simplify the development,

Training, Deployment of

Deep learning models

Deep learning frameworks

These frameworks offer an abstraction layer

for working with neural networks,

making it easier for researchers and developers

to implement complex architectures and

algorithms without having

to build everything from scratch.

TensorFlow

Developed by the Google Brain team

TensorFlow is an open-source

machine learning library that

supports both deep learning and

traditional machine learning.

TensorFlow Features

Flexible and scalable for a wide range of applications.

TensorFlow offers a high-level API

Called Keras for easy model building.

TensorFlow Lite enables

deployment on mobile and edge devices.

TensorFlow Use Cases

Image and speech recognition.

Natural language processing.

Reinforcement learning.

Website: TensorFlow

2. PyTorch

An open-source deep learning framework

developed by Facebook's AI Research lab (FAIR).

It has gained popularity

for its dynamic computation graph

ease of use.

2. PyTorch Features

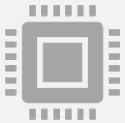
Dynamic computation graph allows for more intuitive model building and debugging.

PyTorch Lightning provides a lightweight PyTorch wrapper for high-level abstractions

Extensive community support and a strong ecosystem.

2. PyTorch Use Cases

Natural language processing.



Computer vision.

Generative models.

Website: PyTorch

3. Keras

Initially developed as a high-level API

for building neural networks

on top of other frameworks,

Keras is now integrated as

the official high-level API for TensorFlow.

3. Keras Features

Simple and user-friendly syntax

for building

experimenting with models.

3. Keras Features

- Supports Convolutional,
- Recurrent, and
- Dense networks.
- Easy model deployment
- using TensorFlow.

3. Keras Use Cases

Rapid prototyping of deep learning models.

Beginner-friendly deep learning projects.

Website: Keras

4. Apache MXNet

An open-source deep learning framework

supports both symbolic and

imperative programming.

Designed for flexibility and efficiency

4. Apache MXNet Features

Supports symbolic and imperative programming models.

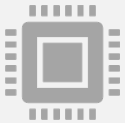
Efficient for distributed computing and multi-GPU training.

Gluon API provides a concise and dynamic approach

to building models.

4. MXNet Use Cases

Natural language processing.



Computer vision.

Recommendation systems.

Website: Apache MXNet

5. Caffe

Deep learning framework developed by the Berkeley Vision and

Learning Center.

Known for its speed and efficiency,

Especially in Convolutional Neural Networks (CNNs)

5. Caffe Features

Fast and efficient for image classification tasks.

Supports GPU acceleration.

Pre-trained models available in the Caffe Model Zoo.

5. Caffe Use Cases

Image classification.

Object detection.

Website: Caffe

6. Theano (Deprecated):

- Early deep learning framework
- Developed by the Montreal Institute for Learning Algorithms
- It is no longer actively maintained

6. Theano (Deprecated) Features

Efficient computation of mathematical expressions.

GPU acceleration support.

Integration with NumPy.

6. Theano (Deprecated) Use Cases

Early deep learning research and experimentation.

Website: Theano

Summary and Conclusion

Choosing a deep learning framework

depends on factors such as

personal preference,

project requirements, and

community support.

Summary and Conclusion

TensorFlow and PyTorch

The most popular

Widely used frameworks,

Each with its strengths

Unique features.

Summary and Conclusion

- Frameworks offer
- Compatibility with popular hardware accelerators like
- Graphics Processing Unit (GPU)
- Tensor Processing Units (TPUs)

Happy Learning !!

Thank You for your patience ☺

Happy to Connect !!☺

Email: Surendra@gktcs.com
Mobile: 9975072320