COIT20277 Introduction to Artificial Intelligence

Week 6

Deep Learning

- AI, Machine Learning and Deep Learning
- Artificial Neural Networks

CQUniversity AUSTRALIA

BE WHAT YOU WANT TO BE
cqu.edu.au

Acknowledgement of Country

I respectfully acknowledge the Traditional Custodians of the land on which we live, work and learn. I pay my respects to the First Nations people and their Elders, past, present and future

Acknowledgment

The contents of this lecture have been adopted from the following references:

- Artificial Intelligence with Python (2nd edition), *Artasanchez and Joshi*, ISBN 978-1-83921-953-5
  - Chapter 19

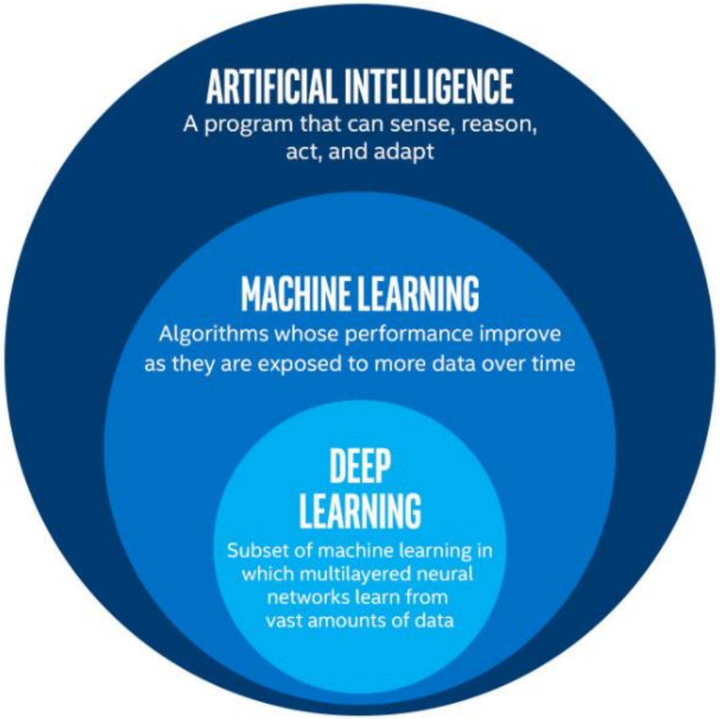
- Artificial Intelligence Programming with Python - From Zero to Hero, 2022, Perry Xiao, ISBN 978-1-119-82086-4:
  - Chapter 4

Topics

- AI vs ML vs Deep Learning
- What is Deep Learning?
- Types of Deep Neural Networks
- Artificial Neural Networks (ANNs)
- Training an ANN by Error Backpropagation
- Gradient Descent vs Stochastic Gradient Descent

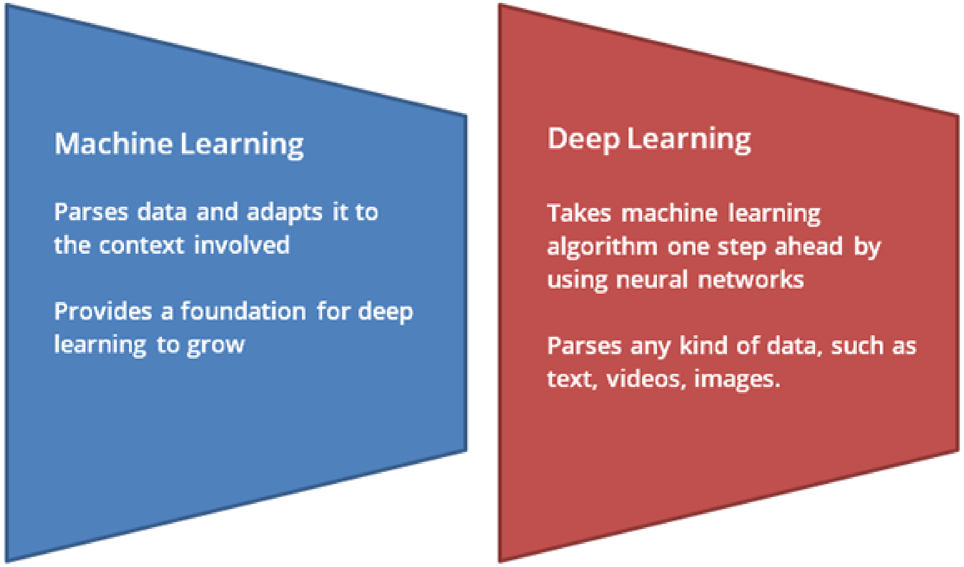
AI vs ML vs Deep Learning

- **Artificial Intelligence (AI)** is the overarching field of computer science focused on creating intelligent systems capable of simulating human-like intelligence.
- **Machine Learning (ML)** is a subset of AI that enables systems to learn from data and improve their performance over time without being explicitly programmed.
- **Deep Learning (DL)** is a specialized form of ML that uses deep neural networks to automatically learn representations of data.


ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Machine Learning vs Deep Learning

- Machine Learning requires human intervention to extract relevant features from the input data, while Deep Learning automatically learns the features from the raw data.
- Machine Learning uses simpler and shallower models, such as linear regression or decision trees, while Deep Learning uses complex and deeper models, such as neural networks or convolutional neural networks.



Machine Learning

Parses data and adapts it to the context involved

Provides a foundation for deep learning to grow

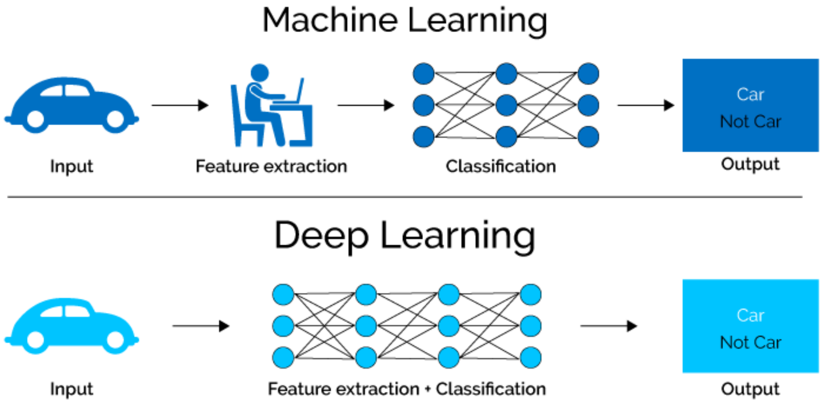
Deep Learning

Takes machine learning algorithm one step ahead by using neural networks

Parses any kind of data, such as text, videos, images.

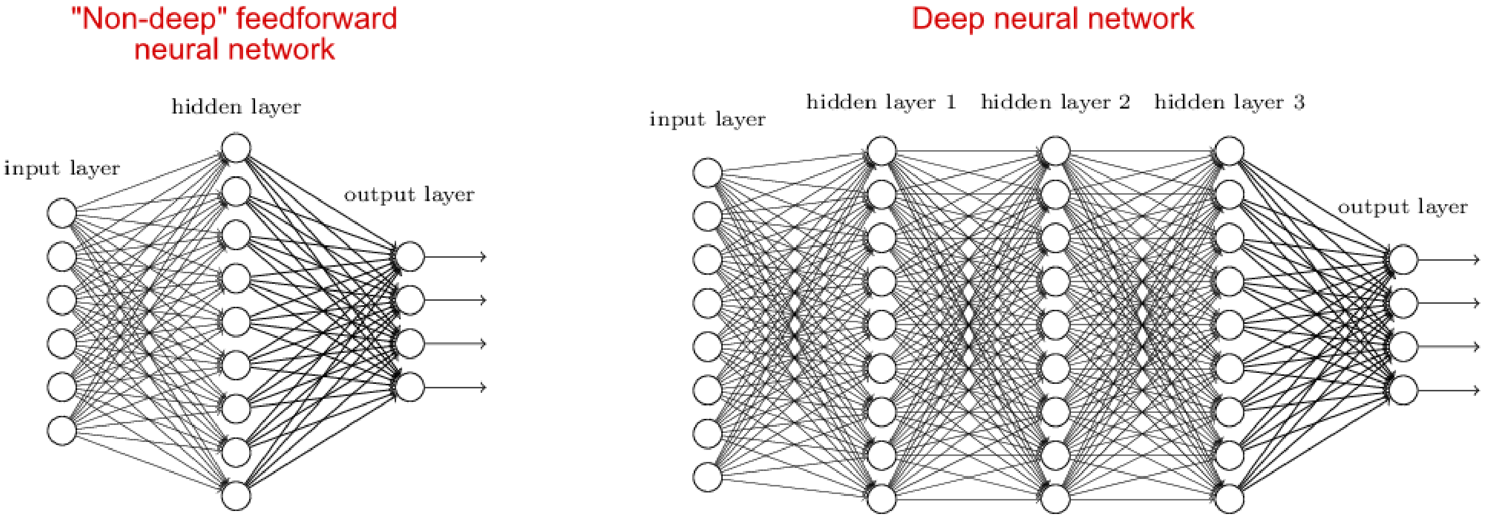
Example of Machine Learning vs Deep Learning

- The image shows an example of how Machine Learning and Deep Learning can classify an input image of a car.
- Machine Learning:
  - The input image is processed by a human expert who selects and extracts the features that are important for identifying a car, such as shape, color, or size.
  - The extracted features are then fed into a classification model, such as a neural network, that outputs a probability of the image being a car or not.
- Deep Learning:
  - The input image is directly fed into a classification model, such as a neural network, that has multiple layers of processing units that can learn the features from the raw data.
  - The model outputs a probability of the image being a car or not.



What is Deep Learning?

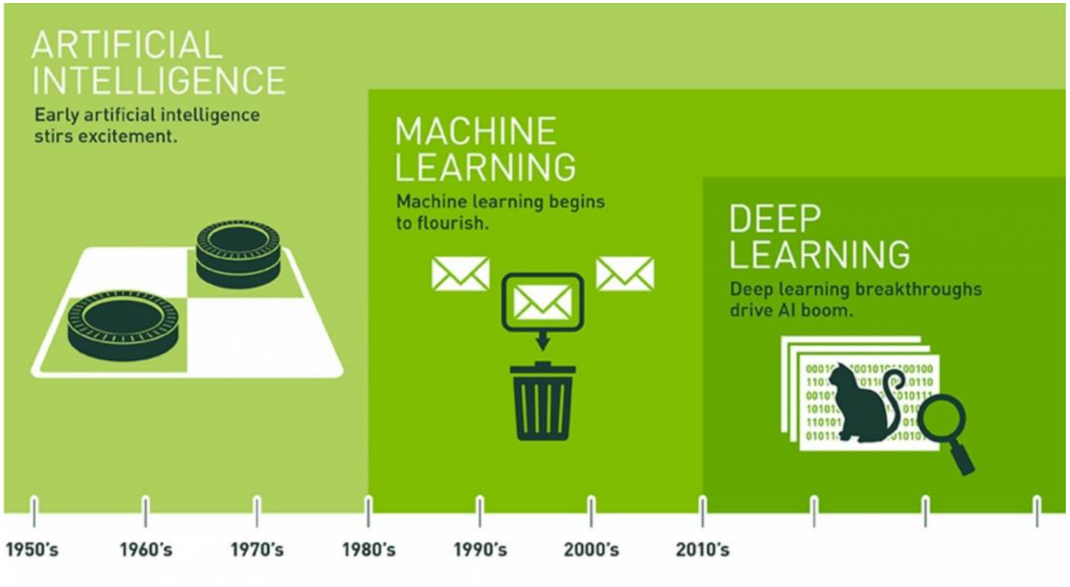
- Deep learning is a subset of machine learning that uses artificial neural networks with multiple hidden layers.
- Traditional neural networks typically have only one hidden layer, while deep learning neural networks can have many.
- This allows deep learning models to learn more complex patterns from data.



This Photo by Unknown Author is licensed under CC BY-SA

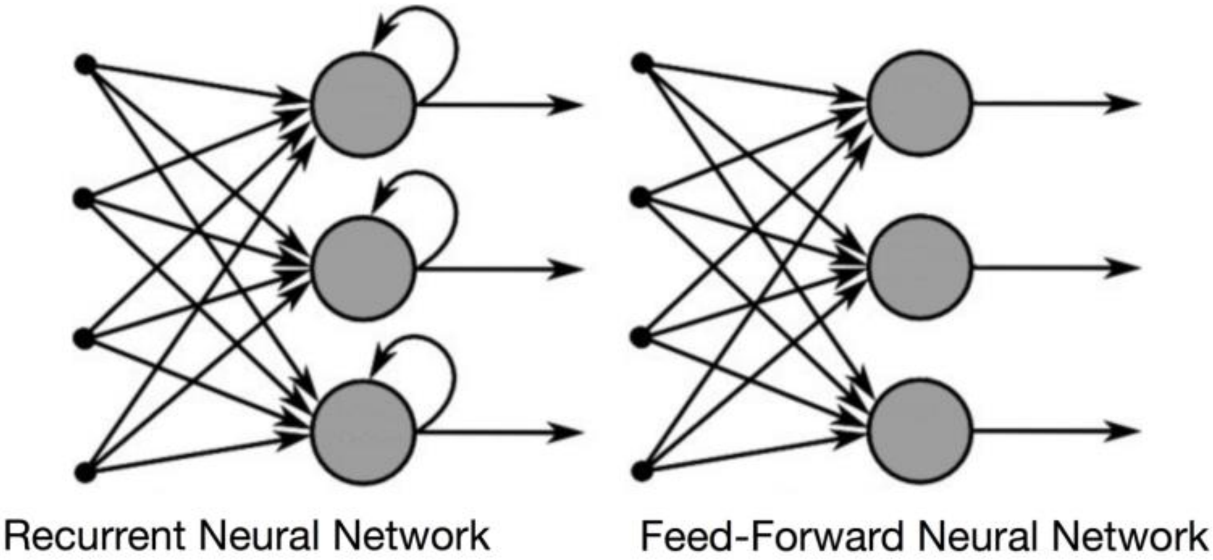
History of Deep Learning

- Deep learning can be traced back to the 1980s, but it only recently became practical due to advances in computing power.
- In 2012, a deep learning model called AlexNet won the ImageNet competition, a benchmark for image recognition. This win helped to popularize deep learning.
- Since then, deep learning has been used to achieve state-of-the-art results in many tasks, including image recognition, speech recognition, and natural language processing.


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Types of Deep Neural Networks

- There are two main types of deep learning neural networks: convolutional neural networks (CNNs) and recurrent neural networks (RNNs).
- CNNs are designed for processing images and other grid-like data.
- RNNs are designed for processing sequential data, such as text and time series.



Applications of Deep Learning

- Deep learning is being used in a wide range of applications, including:
- **Image recognition**: Deep learning models can be used to recognize objects in images, such as faces, cars, and animals.
- **Speech recognition**: Deep learning models can be used to convert spoken language into text.
- **Natural language processing**: Deep learning models can be used to understand and generate human language.
- **Self-driving cars**: Deep learning models are being used to develop self-driving cars that can navigate roads without human input.
- **Healthcare**: Deep learning models are being used to analyze medical images and diagnose diseases.

Impact of Data Size on Algorithm Performance

- Traditional ML algorithms perform well with limited data but plateau with larger datasets.
- Conventional neural networks improve with more data but eventually reach a plateau.
- Deep learning networks continue to improve performance as data size increases.

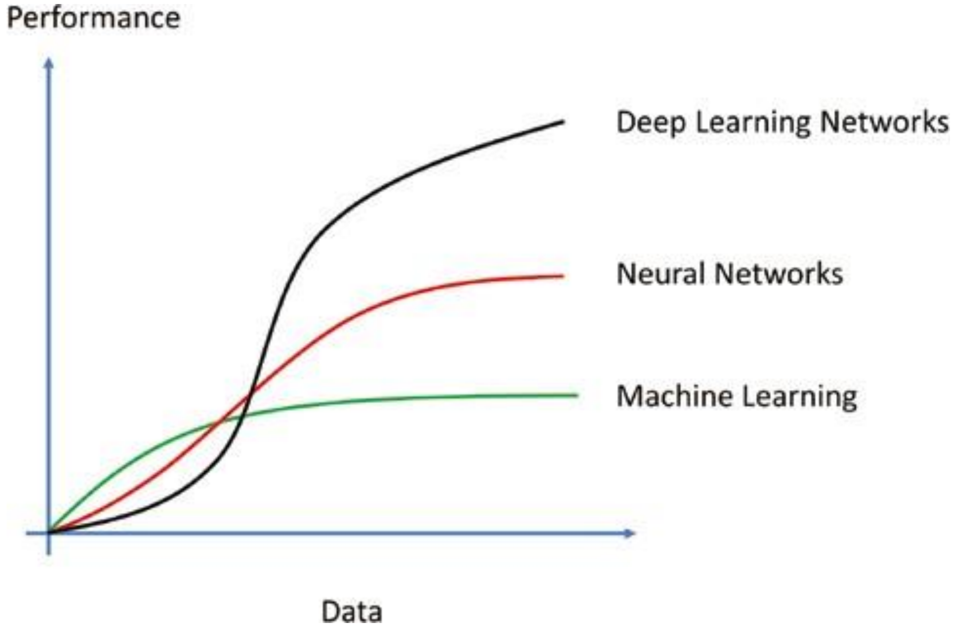
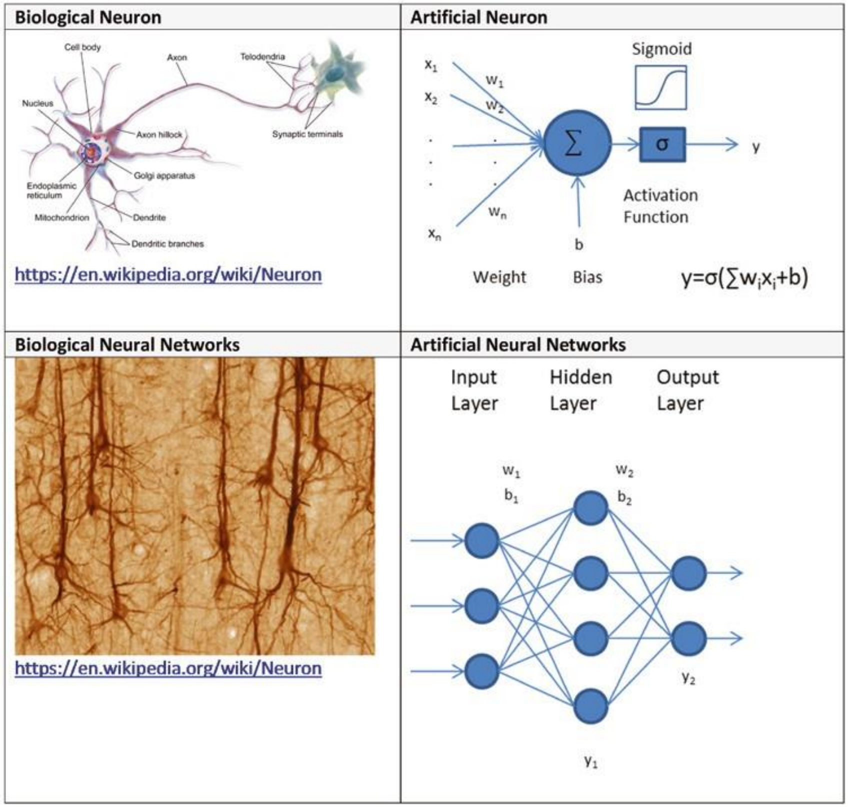


Figure 4.3 (Xiao, P., 2022): The typical performances of traditional machine learning algorithms, traditionalneural networks, and deep learning neural networks against the data

Artificial Neural Networks

- Computer simulations of biological neural networks in the human brain
- Developed by Warren McCulloch and Walter Pitts in 1943
- Consist of interconnected artificial neurons that take input signals, multiply them by weights, and pass them to an activation function for output
- Usually have three layers: input, output, and hidden
- Trained by adjusting the weights using backpropagation and gradient descent.



Biological Neuron

https://en.wikipedia.org/wiki/Neuron

Artificial Neuron

Biological Neural Networks

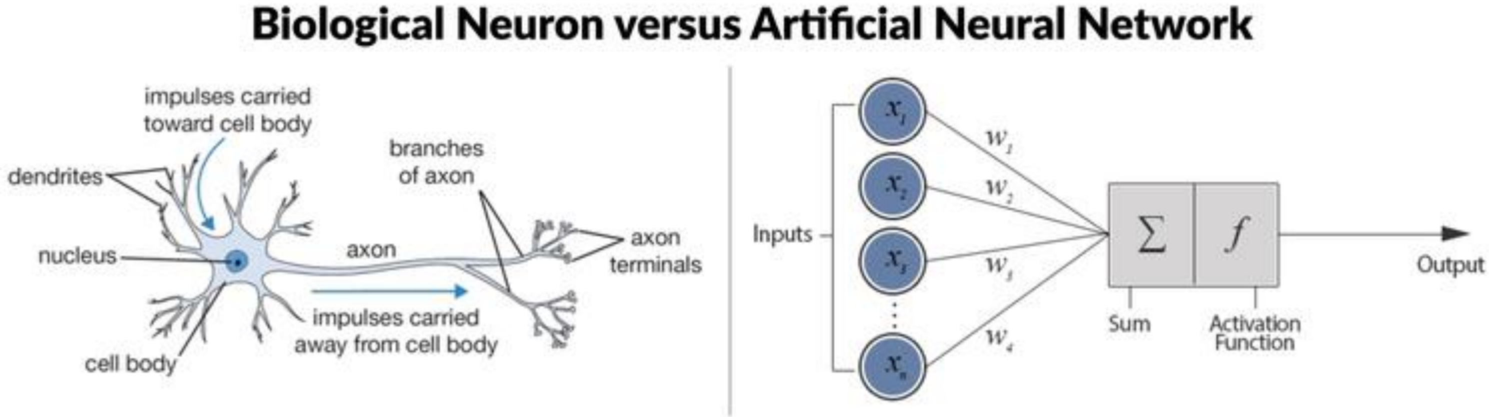
https://en.wikipedia.org/wiki/Neuron

Artificial Neural Networks

Figure 4.4 (Xiao, P., 2022): Biological neuron versus artificial neuron, and biological neuron networks versus artificial neuron networks

Biological Neuron Networks vs Artificial Neuron Networks

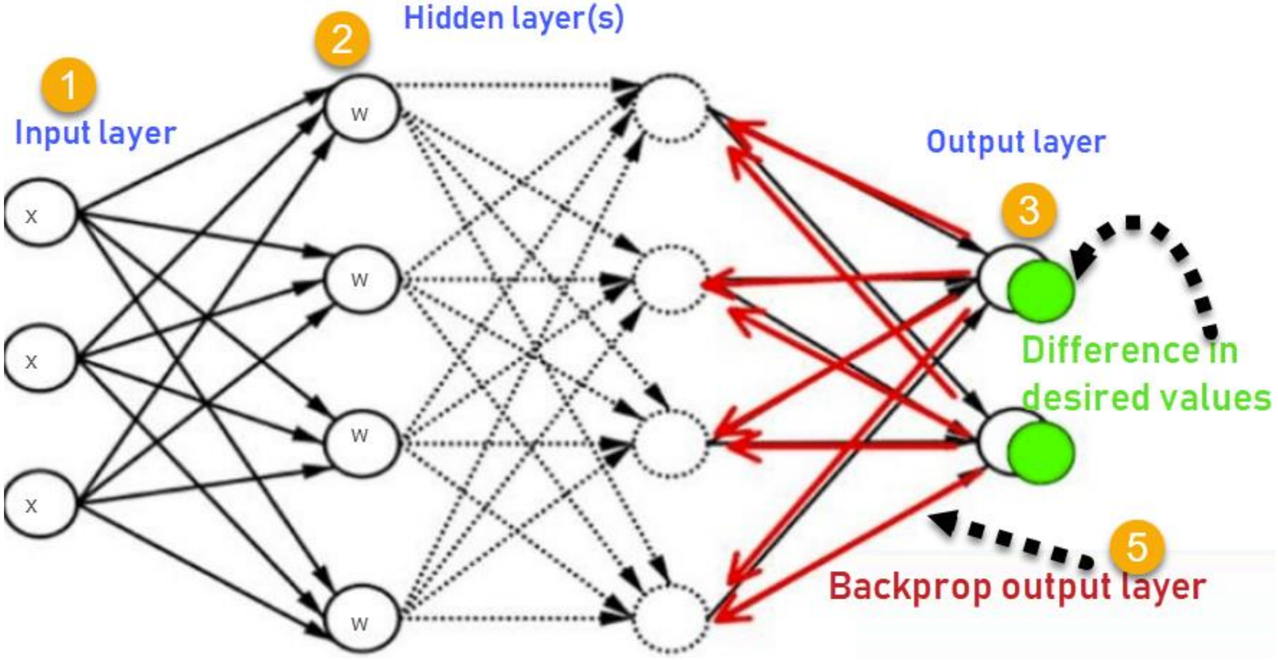
- Biological neuron networks: composed of billions of neurons that form complex patterns and circuits.
- Artificial neuron networks: composed of a finite number of neurons that form simple layers and connections.
- Both can learn from data and perform tasks, but differ in the scale and flexibility of the networks.



This Photo by Unknown Author is licensed under CC BY-SA

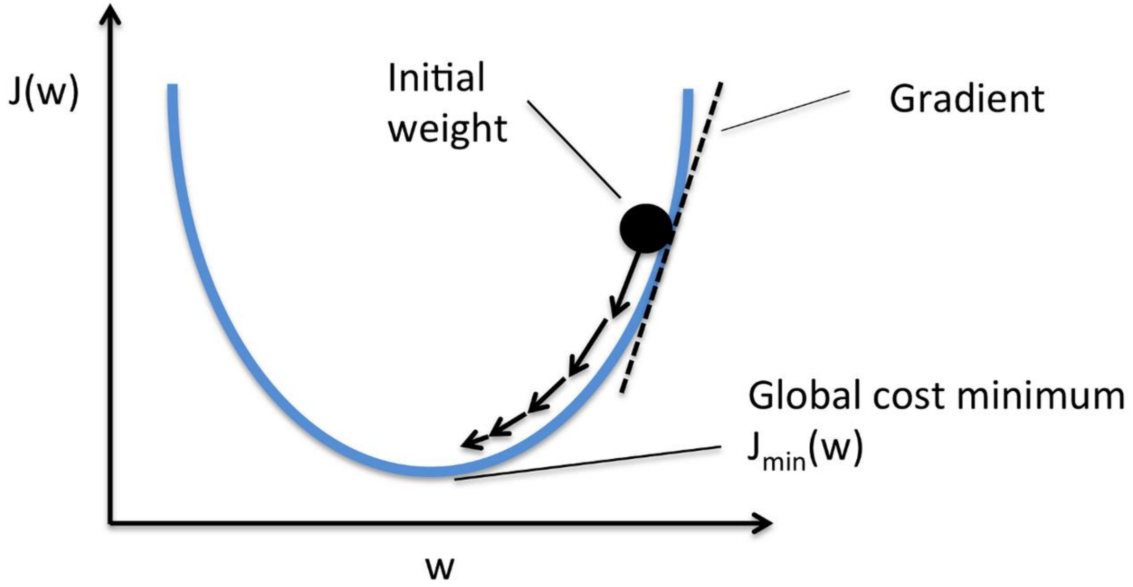
Training an ANN by Error Backpropagation

- Initialize the weights randomly.
- Feed the network a set of training data with inputs and outputs.
- Compare the computed output with the desired output using a loss function.
- Return the errors to the network and adjust the weights accordingly.
- Repeat until the weights are stabilized.
- This is an optimization problem that can be solved by gradient descent.



Gradient Descent

- An iterative optimization algorithm to find a local minimum of a function.
- Take repeated steps in the opposite direction of the gradient, which gives the direction of steepest descent.
- This is like going down from the top of a mountain and finding a way to explore the bottom.



Standard vs Stochastic Gradient Descent

- Standard gradient descent follows the path with steepest descent and tends to get stuck at a local minimum.
- Stochastic gradient descent adds randomness to the path and converges faster.
- As a result, stochastic gradient descent reaches the global minimum more often than standard gradient descent.

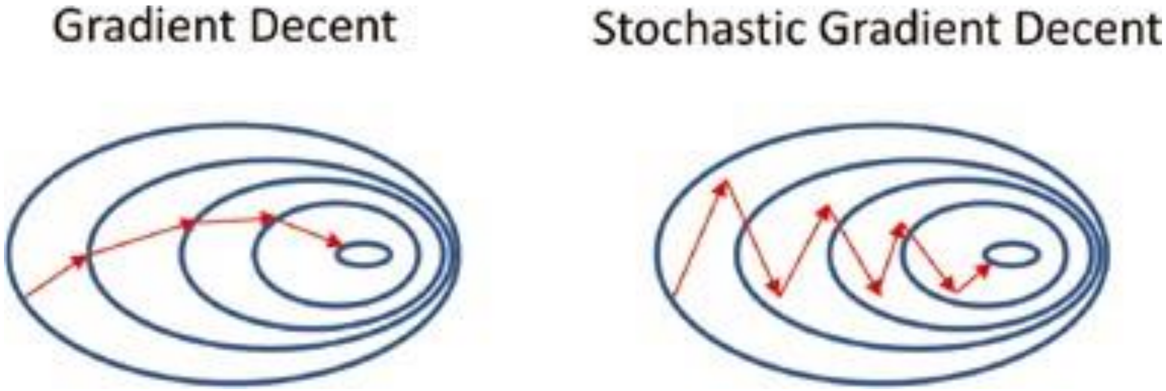
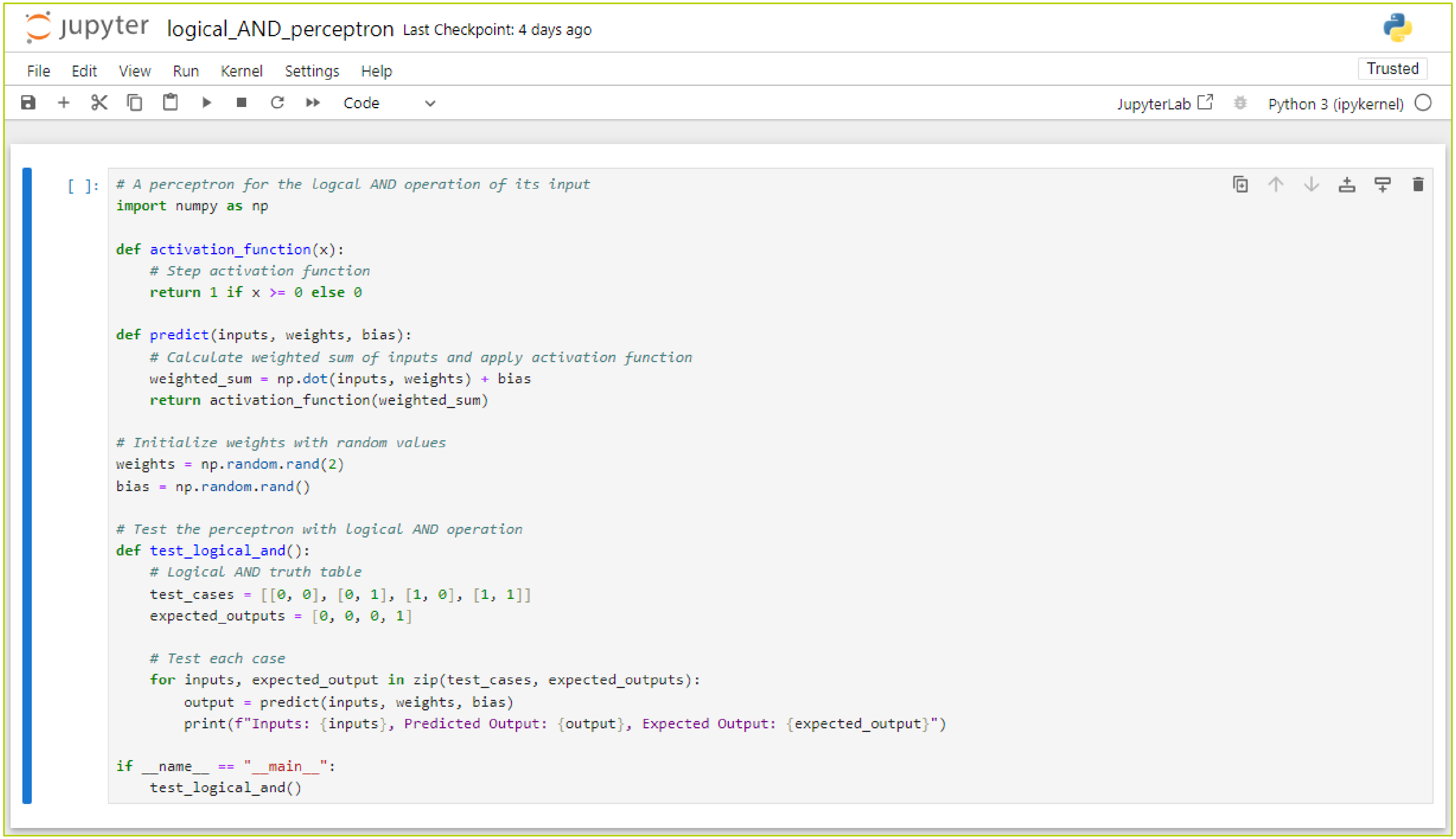


Figure 4.5: The paths of gradient descent and stochastic gradient descent

A Single Neuron Program

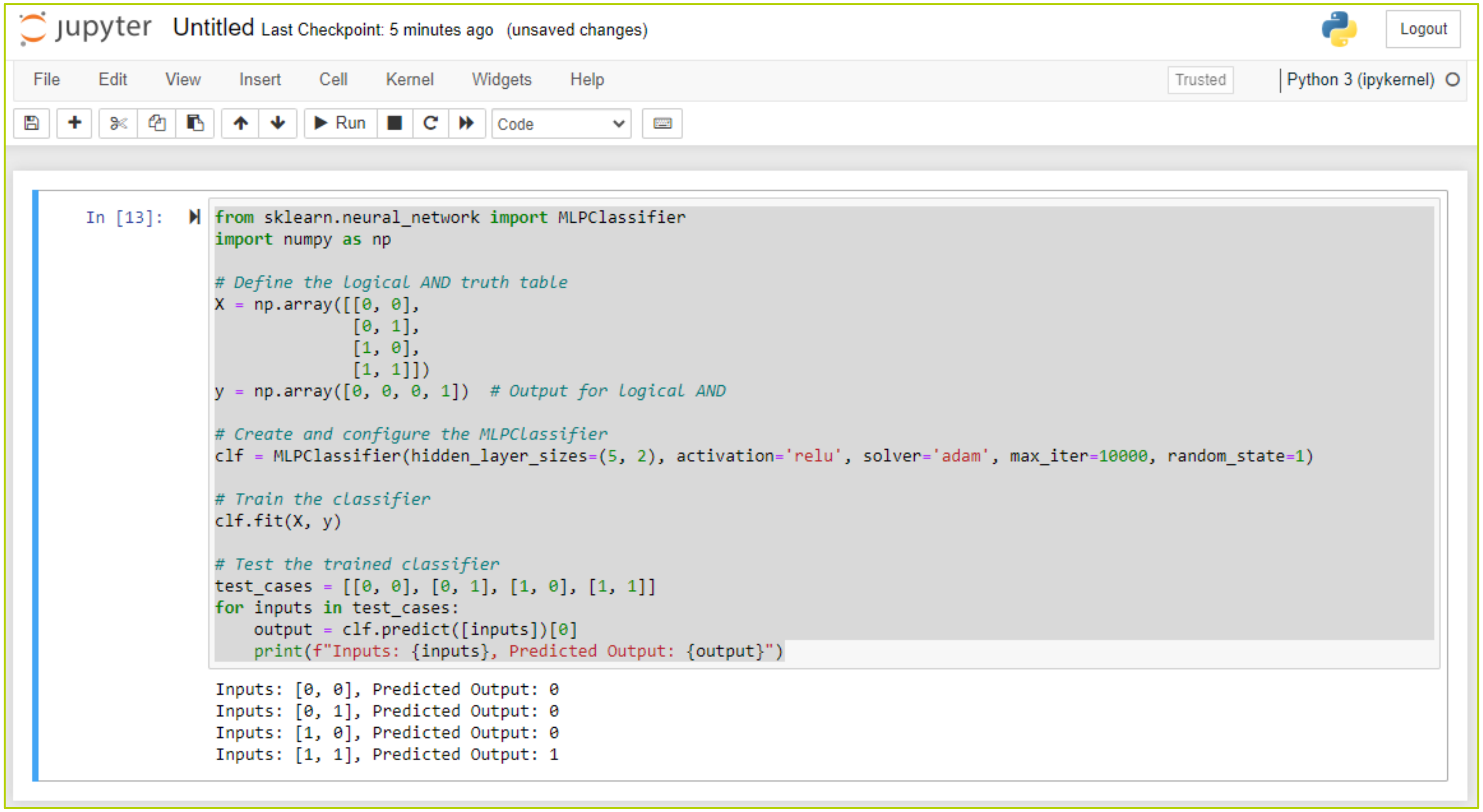
- In ANN research, a single neuron network is also called ***a perceptron***.



```python
# A perceptron for the logcal AND operation of its input
import numpy as np

def activation_function(x):
    # Step activation function
    return 1 if x >= 0 else 0

def predict(inputs, weights, bias):
    # Calculate weighted sum of inputs and apply activation function
    weighted_sum = np.dot(inputs, weights) + bias
    return activation_function(weighted_sum)

# Initialize weights with random values
weights = np.random.rand(2)
bias = np.random.rand()

# Test the perceptron with logical AND operation
def test_logical_and():
    # Logical AND truth table
    test_cases = [[0, 0], [0, 1], [1, 0], [1, 1]]
    expected_outputs = [0, 0, 0, 1]

    # Test each case
    for inputs, expected_output in zip(test_cases, expected_outputs):
        output = predict(inputs, weights, bias)
        print(f"Inputs: {inputs}, Predicted Output: {output}, Expected Output: {expected_output}")

if __name__ == "__main__":
    test_logical_and()
```

The Multiple Layer Neural Network

- This Multiple layer NN implements a logical AND operation.
- It has 2 input and 1 output value, with 2 hidden layers, the first has 5 neurons and the second has 2 neurons. The NN is fully connected.



```python
from sklearn.neural_network import MLPClassifier
import numpy as np

# Define the logical AND truth table
X = np.array([[0, 0],
              [0, 1],
              [1, 0],
              [1, 1]])
y = np.array([0, 0, 0, 1])  # Output for logical AND

# Create and configure the MLPClassifier
clf = MLPClassifier(hidden_layer_sizes=(5, 2), activation='relu', solver='adam', max_iter=10000, random_state=1)

# Train the classifier
clf.fit(X, y)

# Test the trained classifier
test_cases = [[0, 0], [0, 1], [1, 0], [1, 1]]
for inputs in test_cases:
    output = clf.predict([inputs])[0]
    print(f"Inputs: {inputs}, Predicted Output: {output}")
```

```
Inputs: [0, 0], Predicted Output: 0
Inputs: [0, 1], Predicted Output: 0
Inputs: [1, 0], Predicted Output: 0
Inputs: [1, 1], Predicted Output: 1
```

THANK YOU

TIME FOR DISCUSSION & QUESTIONS