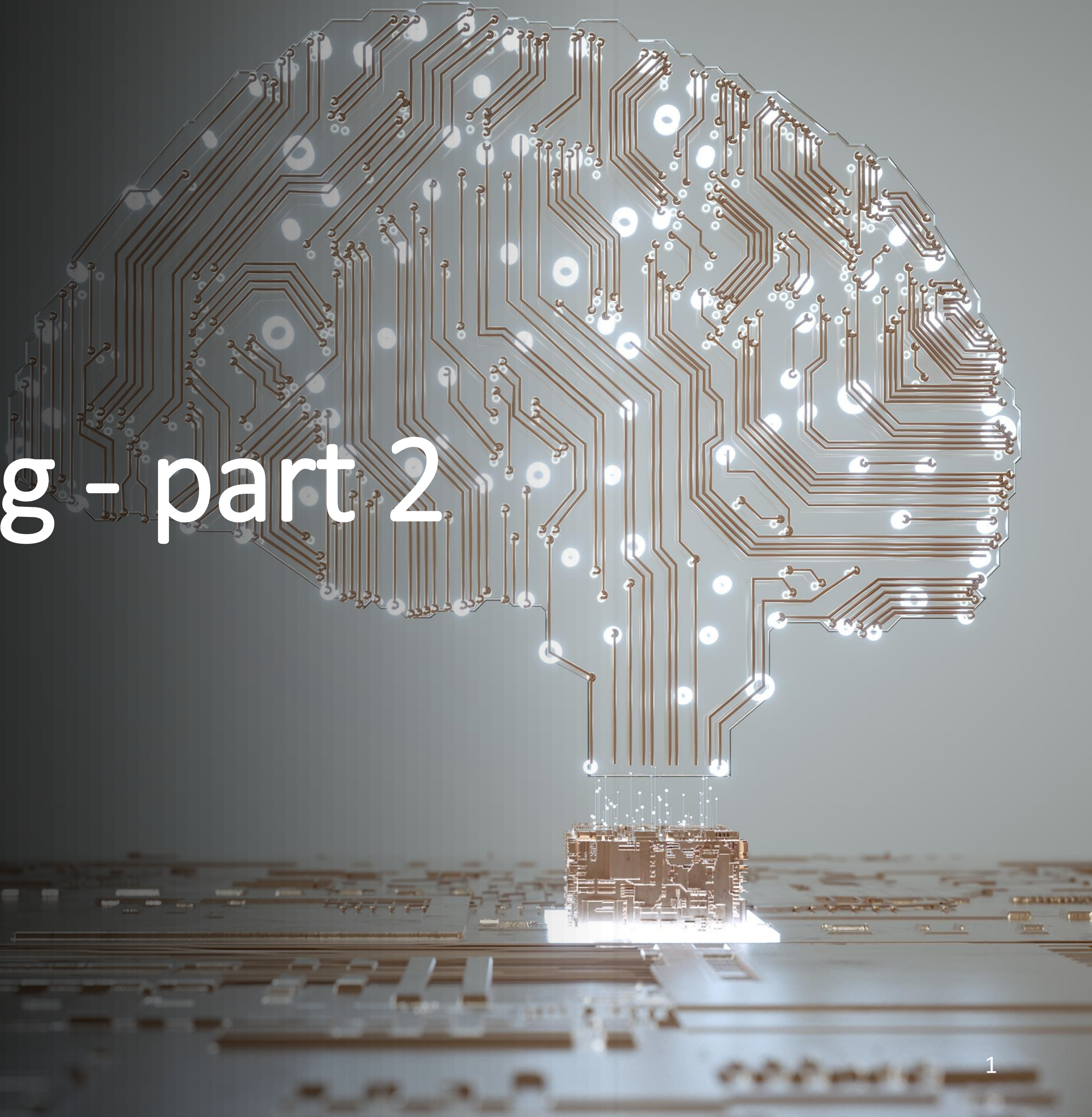


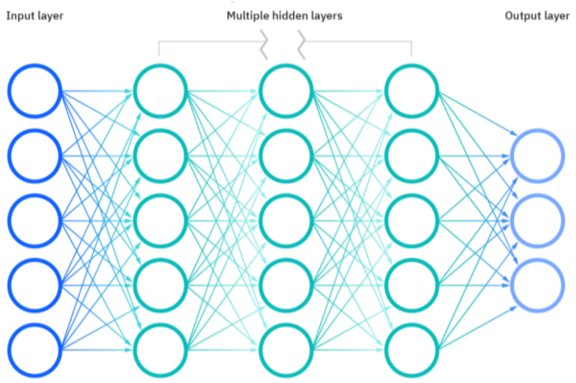
Machine Learning - part 2 Deep Learning

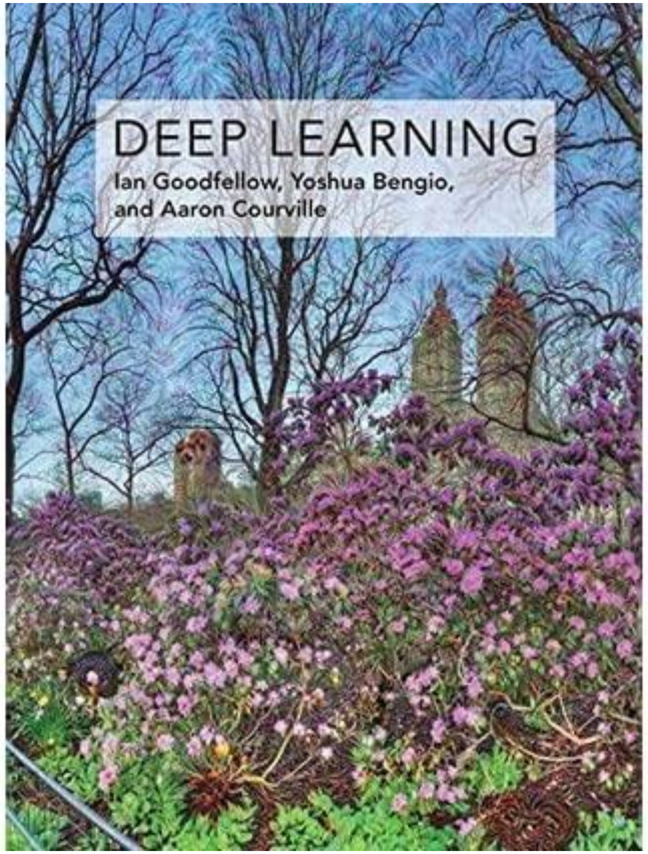
GGE 6505/GGE5405 Introduction to Big Data & Data Science

Deep Learning Textbook

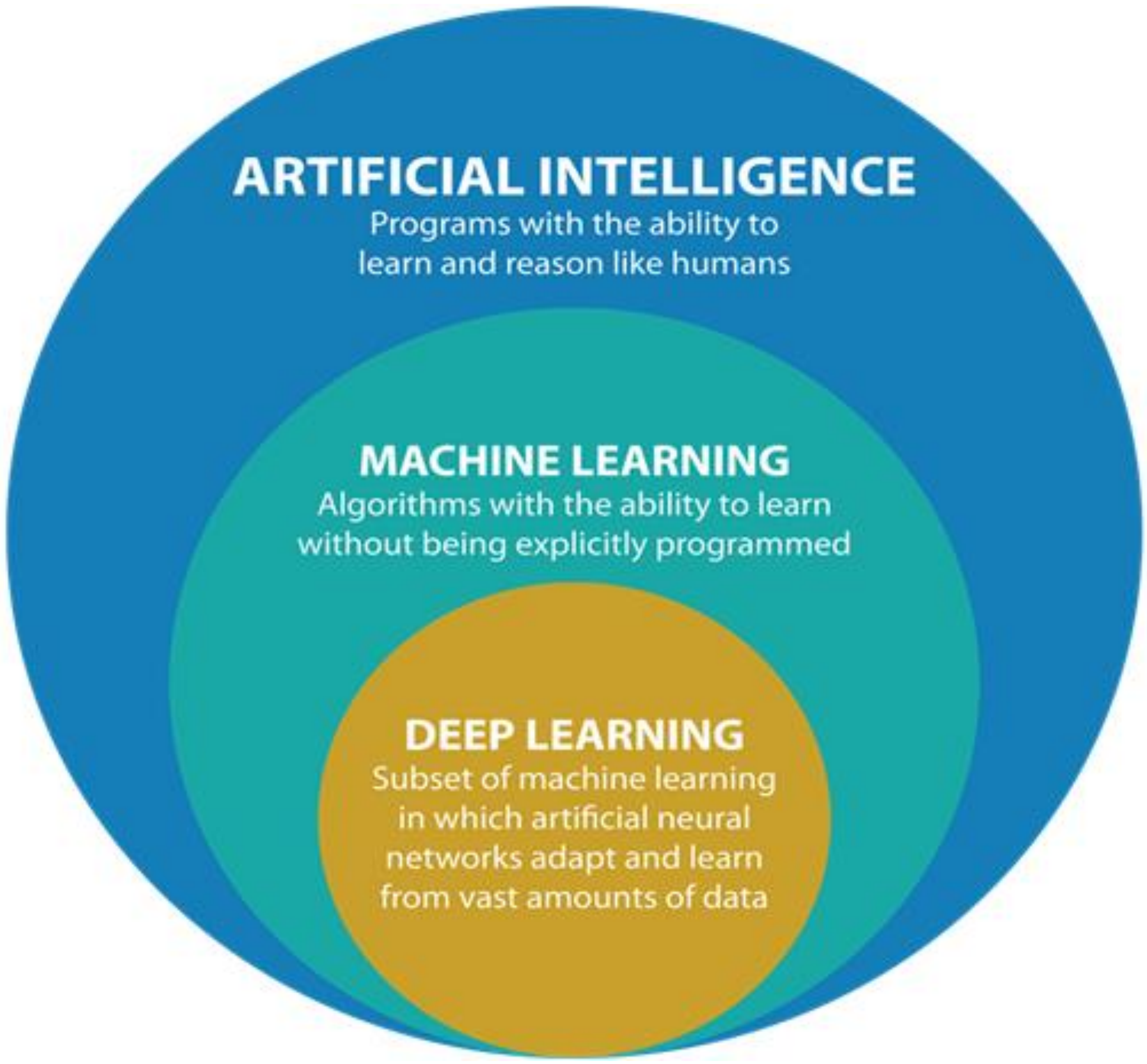
- [Deep Learning (deeplearningbook.org)]

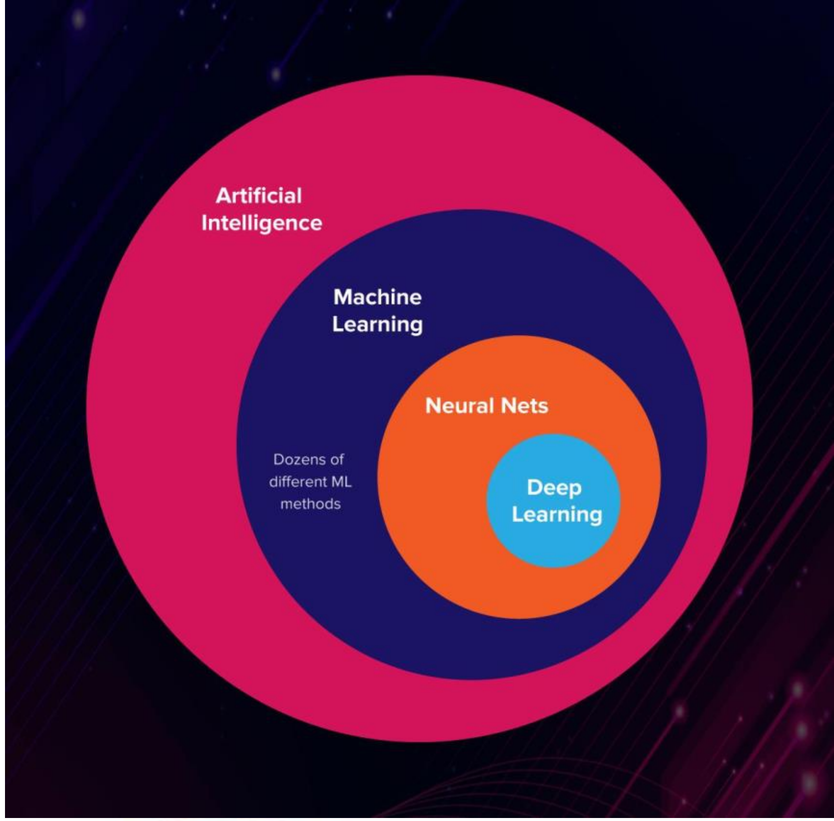
Goodfellow-et-al-2016, Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press,2016

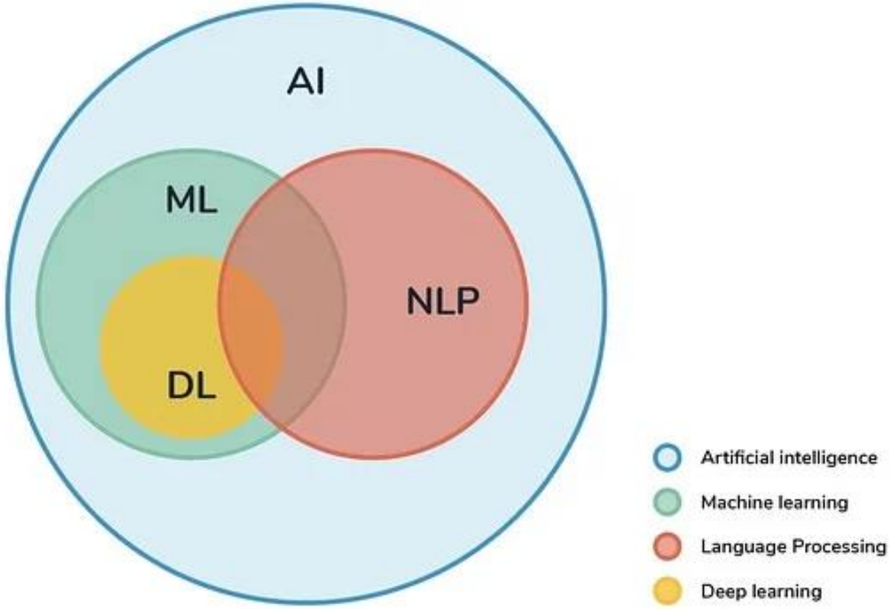




Machine learning vs AI vs Deep learning







Machine learning vs AI vs Deep learning

AI: Artificial intelligence is a field of computer science that makes a computer system that can mimic human intelligence

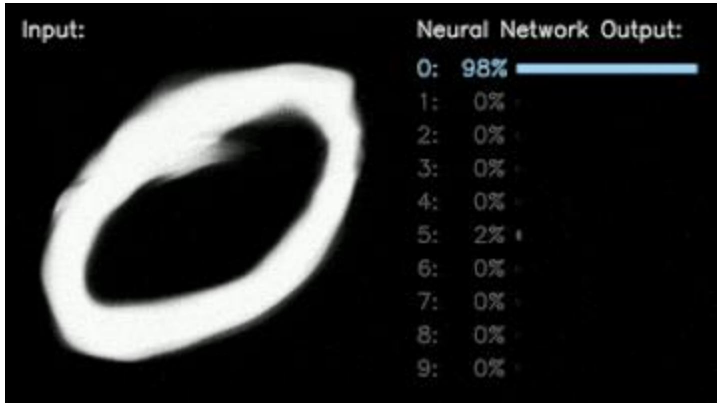
Machine learning: Machine learning enables a computer system to make predictions or take some decisions using historical data without being explicitly programmed

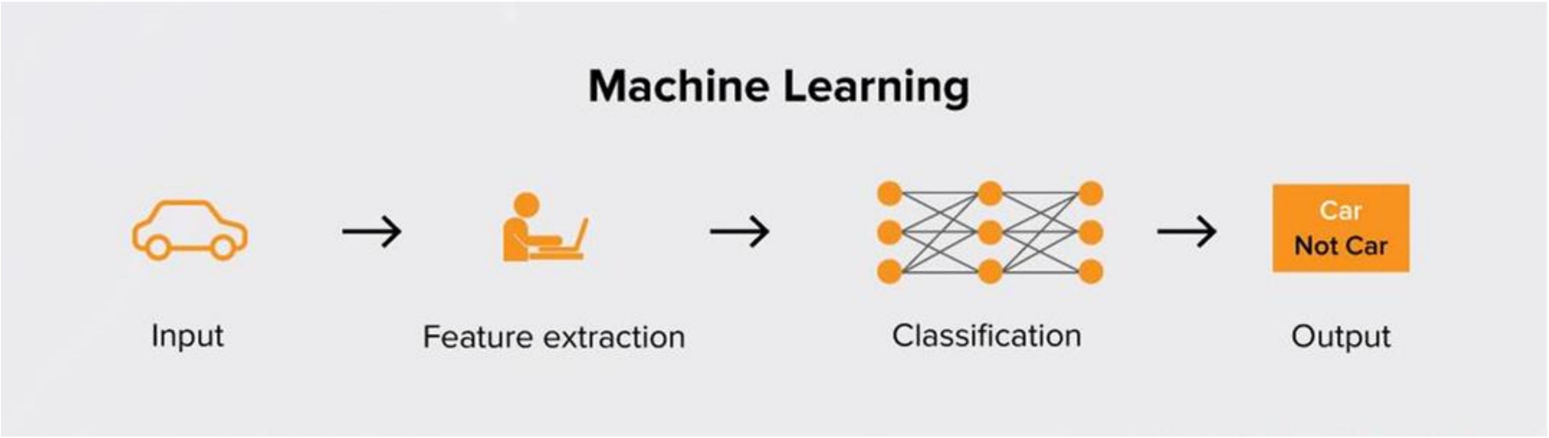
Deep Learning/ Neural Networks: The biggest advantage of Deep Learning is that we do not need to manually extract features from the image. The network learns to extract features while training. You just feed the image to the network (pixel values).

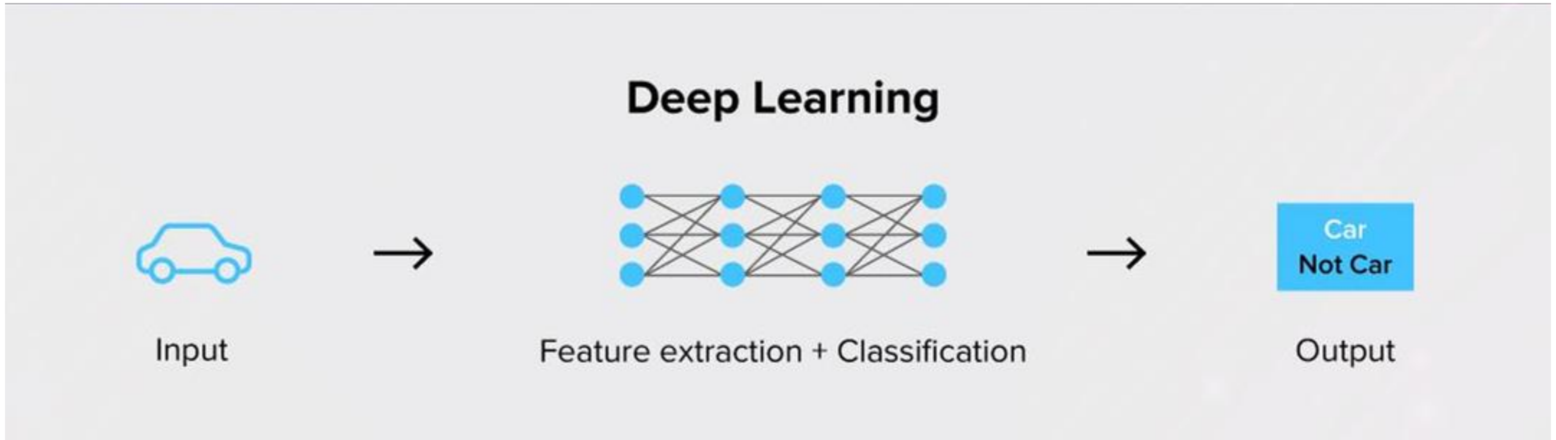
AI VS. MACHINE LEARNING VS. DEEP LEARNING
Artificial Intelligence: a program that can sense, reason, act and adapt.
Machine Learning: algorithms whose performance improve as they are exposed to more data over time.
Deep Learning: subset of machine learning in which multilayered neural networks learn from vast amounts of data.

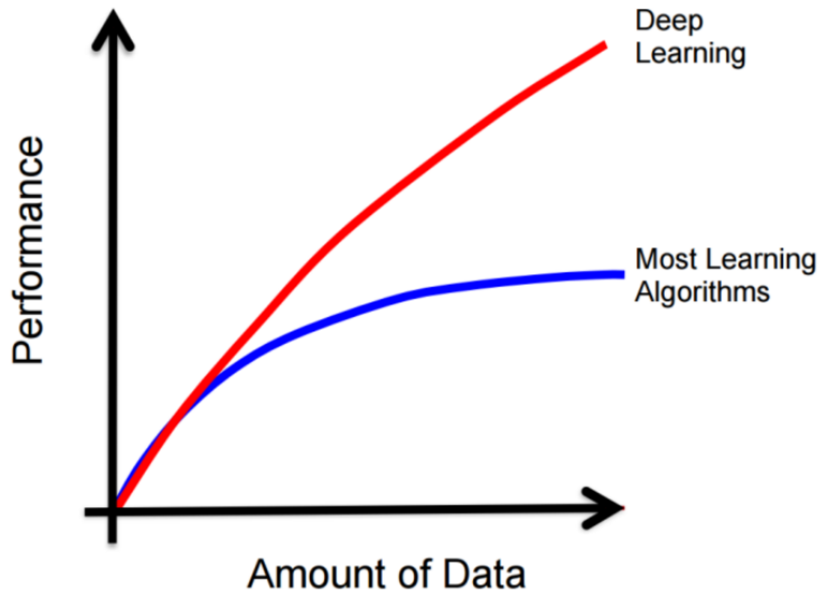
Exciting progress:

- Face recognition
- Image classification
- Speech recognition
- Text-to-speech generation
- Handwriting transcription
- Machine translation
- Medical diagnosis
- Cars: drivable area, lane keeping
- Digital assistants
- Ads, search, social recommendations
- Game playing with deep RL

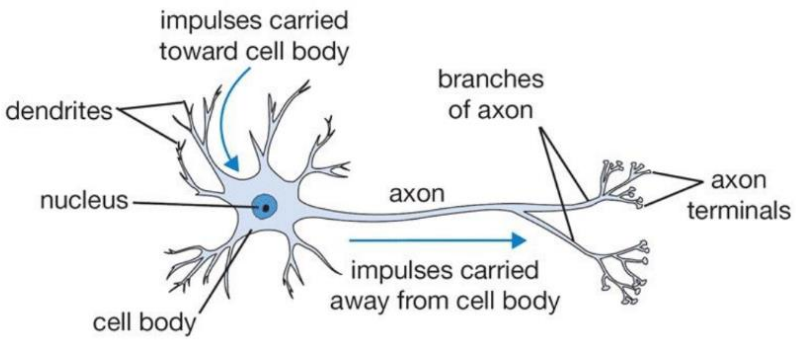



Machine Learning
Car
Not Car
Input
Feature extraction
Classification
Output

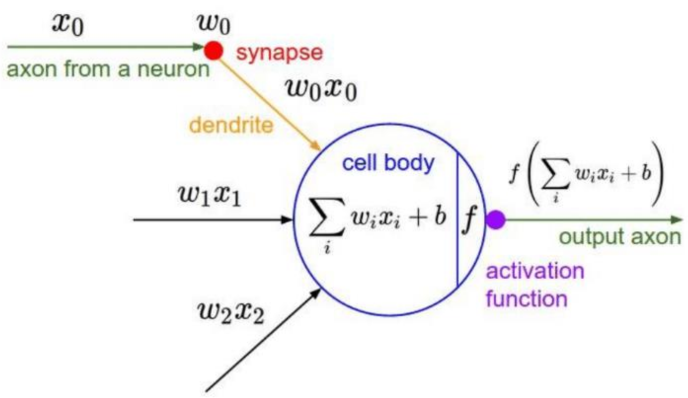

Deep Learning
Car
Not Car
Input
Feature extraction + Classification
Output


Deep
Learning
Most Learning
Algorithms
Performance
Amount of Data

Neuron: Biological Inspiration for Computation



- **Neuron:** computational building block for the brain

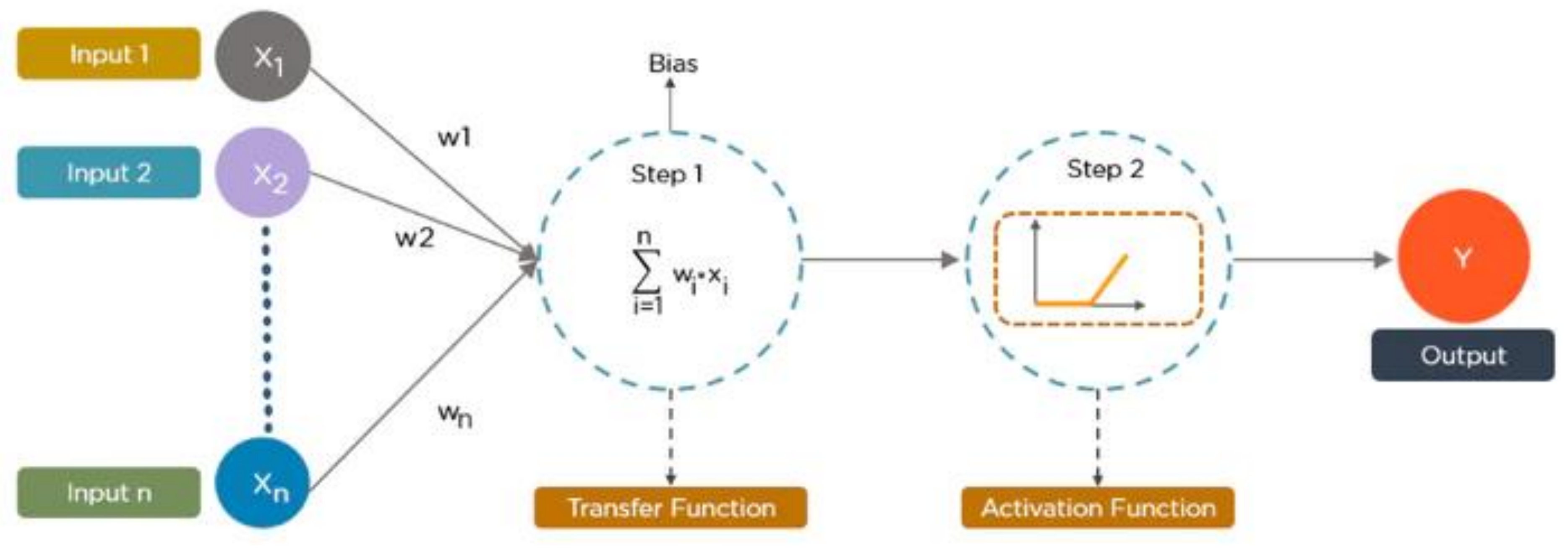


- **(Artificial) Neuron:** computational building block for the “neural network”

Key Difference:

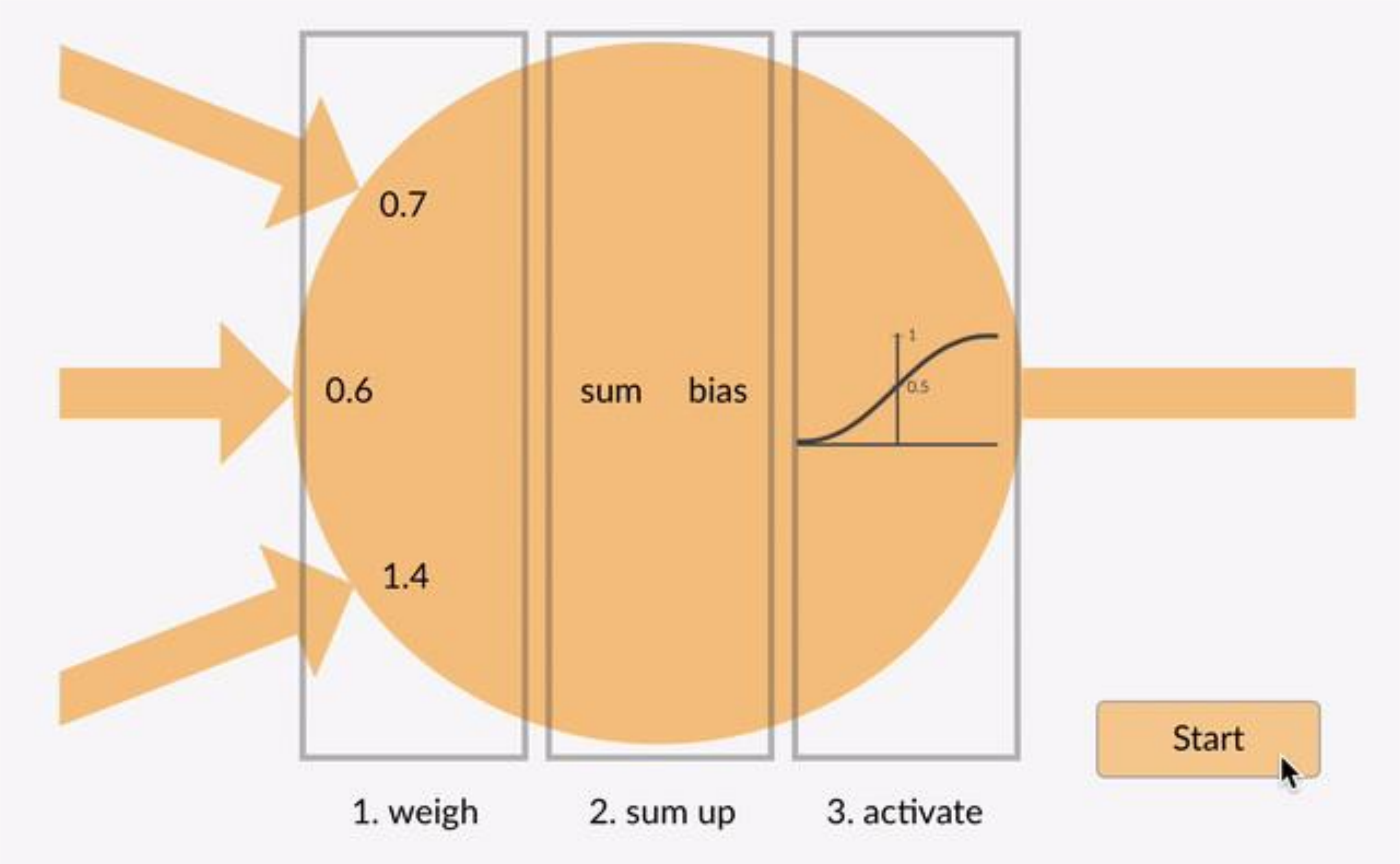
- **Parameters:** Human brains have ~10,000,000 times synapses than artificial neural networks.
- **Topology:** Human brains have no “layers”. **Async:** The human brain works asynchronously, ANNs work synchronously.
- **Learning algorithm**: ANNs use gradient descent for learning. We don’t know what human brains use
- **Power consumption:** Biological neural networks use very little power compared to artificial networks
- **Stages:** Biological networks usually never stop learning. ANNs first train then test.

Neural Network



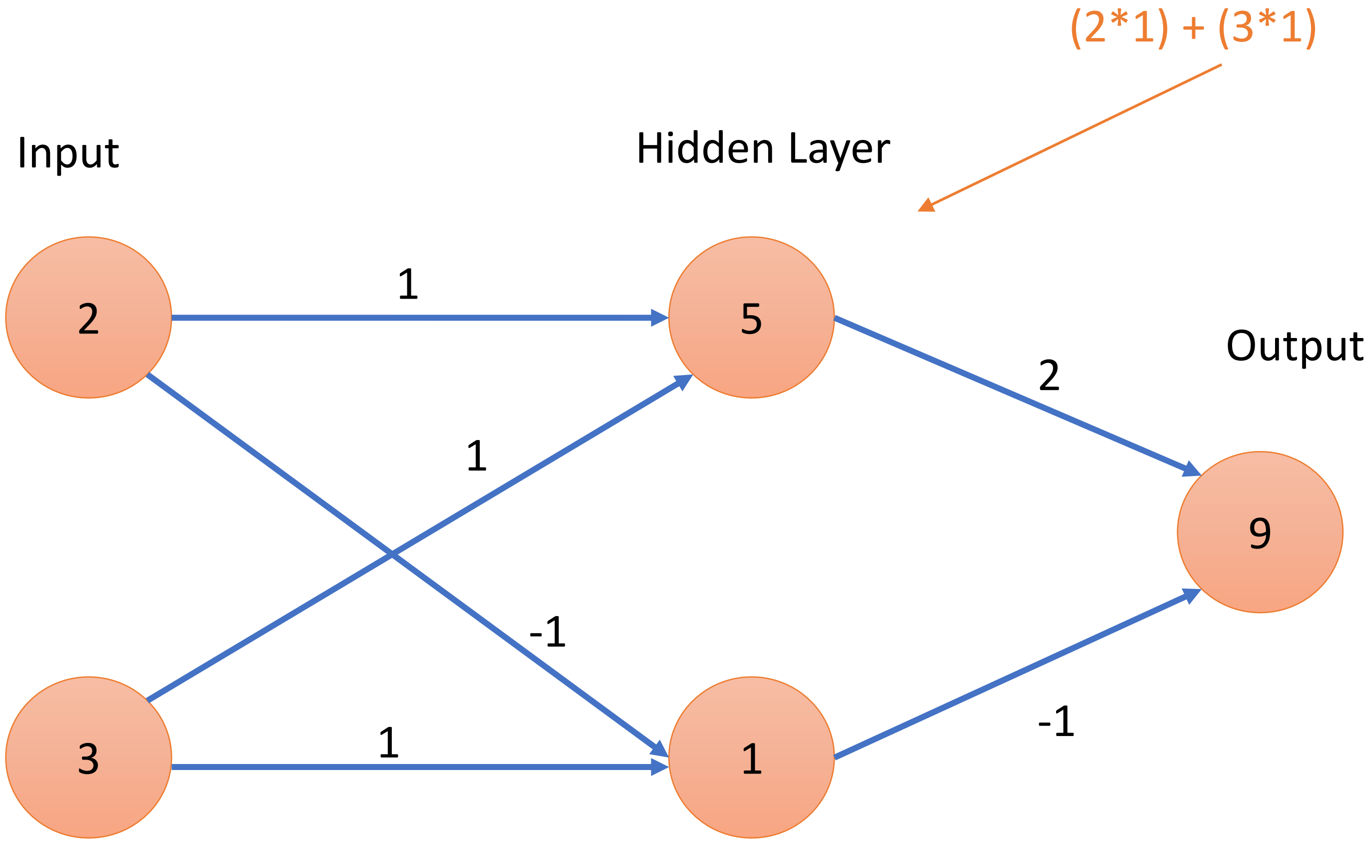
Top 10 Deep Learning Algorithms You Should Know in 2022 (simplilearn.com)

Neural Network can be considered a class of algorithms, mechanisms, or models of Machine Learning that take their inspiration from a human brain since they can have simple to highly complex ‘neural’ networks. Having these layers helps a machine perform computations that would not be possible using just the standard ML algorithms.



[Activation Functions in Neural Networks | by akhil anand | Artificial Intelligence in Plain English](https://)

Forward Propagation



Key concepts:

Activation Function

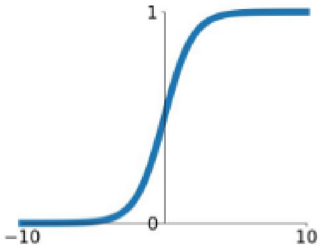
Activation functions in general are used to convert linear outputs of a neuron into [nonlinear outputs](), ensuring that a neural network can learn nonlinear behavior.

```python
import math

def sigmoid(x):
  return 1 / (1 + math.exp(-x))
```

Sigmoid

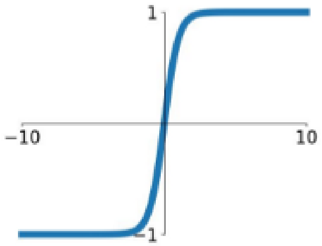
$$\sigma(x) = \frac{1}{1+e^{-x}}$$




```python
def tanh(x):
  return (math.exp(x) - math.exp(-x)) / (math.exp(x) + math.exp(-x))
```

tanh

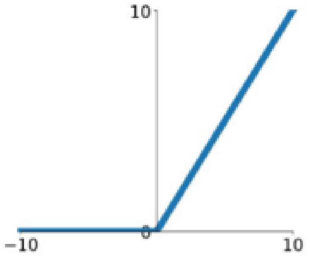
$$\tanh(x)$$



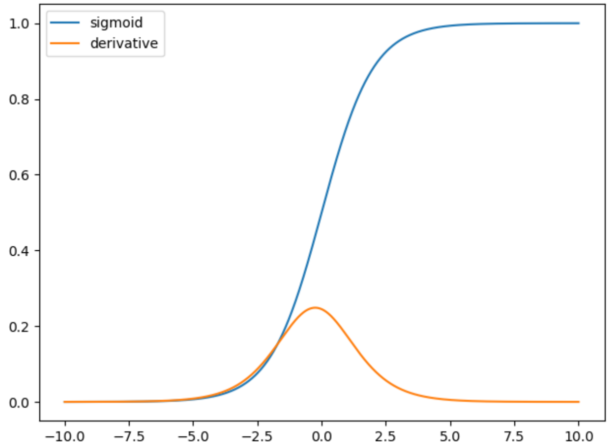

```python
def relu(x):
    return max(0,x)
```

ReLU

$$\max(0, x)$$



Sigmoid function



Advantages:

- It introduces non linearity into dataset
- It can be differentiated which will be resulted into backpropagation in neural network.
- Normalize the data by confining it in a range of [0,1].

Disadvantages:

- Resulted into Vanishing Gradient problem.
- Shows Non-Zero centric behavior.
- Due to presence of exponential function it becomes computationally too much Expensive.

Mathematical Representation

- f(x)=1÷(1+ exp(-x))

Hypertangent function(tanh)

Advantages:

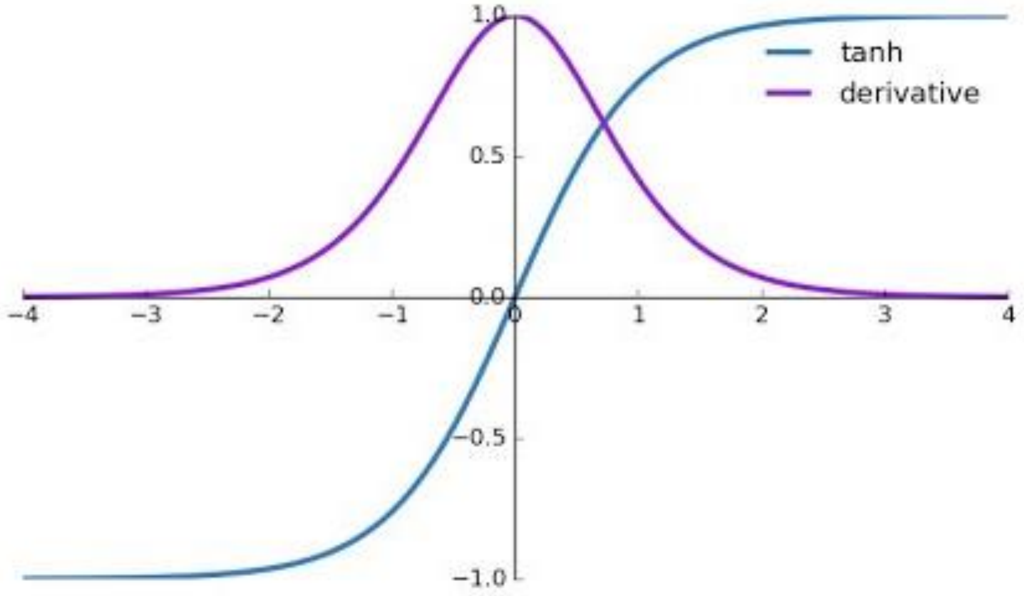
- Zero centric Function
- finds new weight and biases with every back propagation because it’s derivative lies in a range [0,1]
- Better than sigmoid function.

Disadvantages:

- Vanishing Gradient problem.

Mathematical Representation

f(x)= (exp(x)-exp(-x)) ÷ (exp(x)+exp(-x)



ReLU (Rectified Linear Activation)

ReLu overcomes the Vanishing Gradient problem which is found in Sigmoid and tanh function

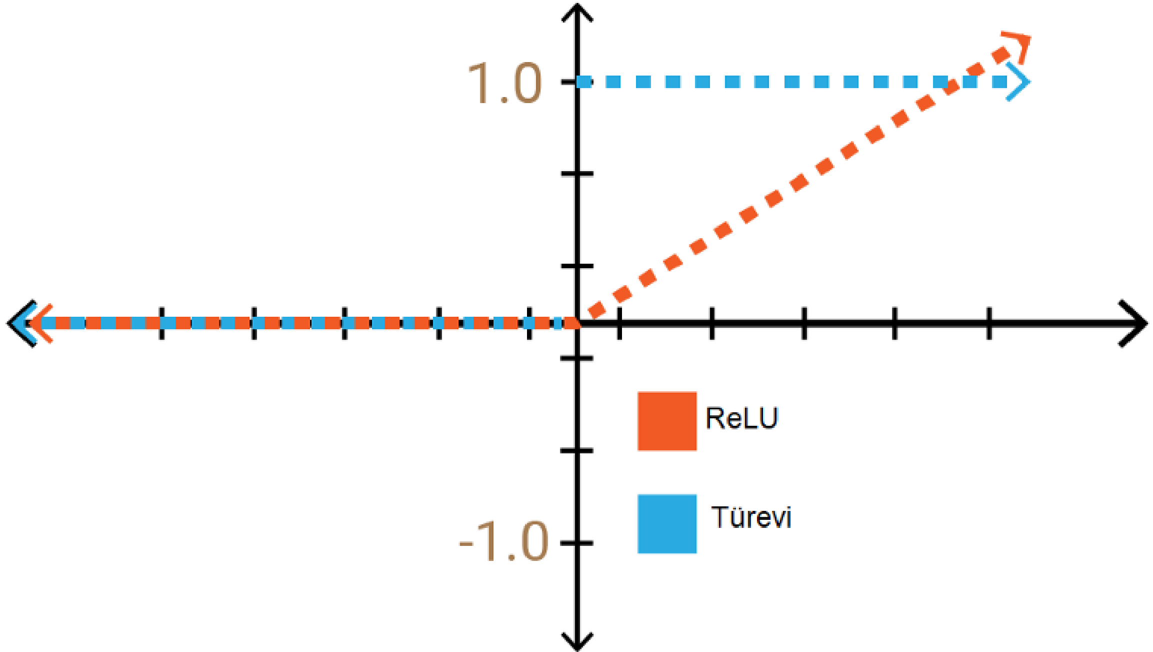
Advantages:

- It doesn't activate all the neuron at the same time

Disadvantages:

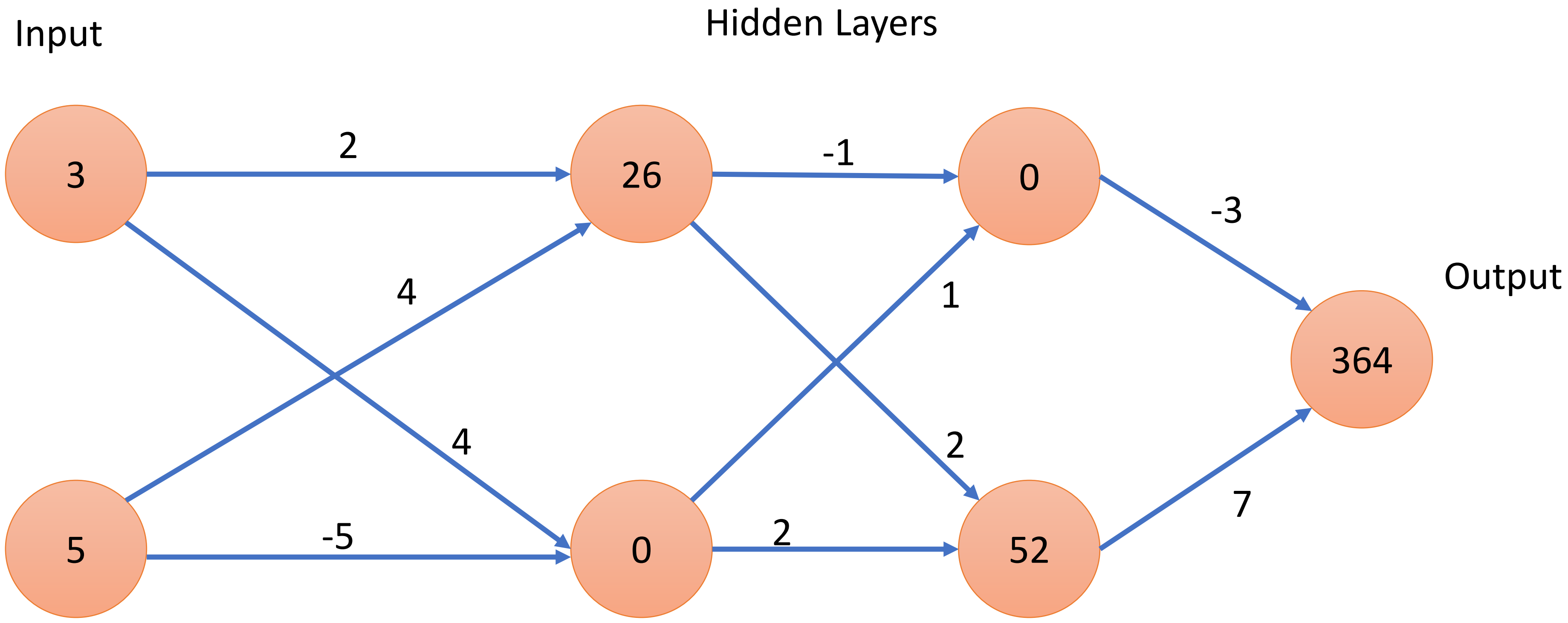
- If the value of training examples will be negative (x<0) it will not be able to find any gradient hence resulted in dying Perceptron problem.

$$\sigma(x) = \begin{cases} max(0, x) & , x >= 0 \\ 0 & , x < 0 \end{cases}$$



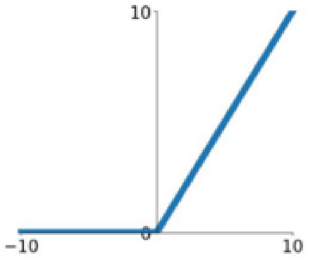
Activation Functions in Neural Networks | by akhil anand | Artificial Intelligence in Plain English

Multiple Hidden Layers

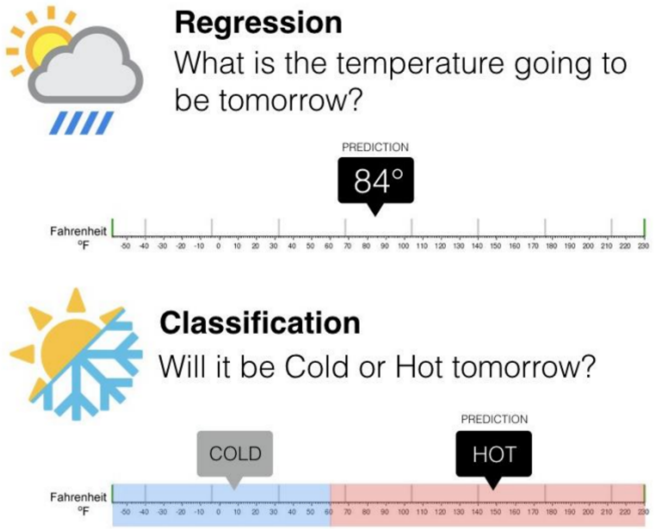


ReLU

$\max(0, x)$



Loss Functions



- Loss function quantifies gap between prediction and ground truth
- For regression:
  - Mean Squared Error (MSE)
- For classification:
  - Cross Entropy Loss

Mean Squared Error

Prediction

$$MSE = \frac{1}{N}\sum(t_i - s_i)^2$$

Ground Truth

Cross Entropy Loss

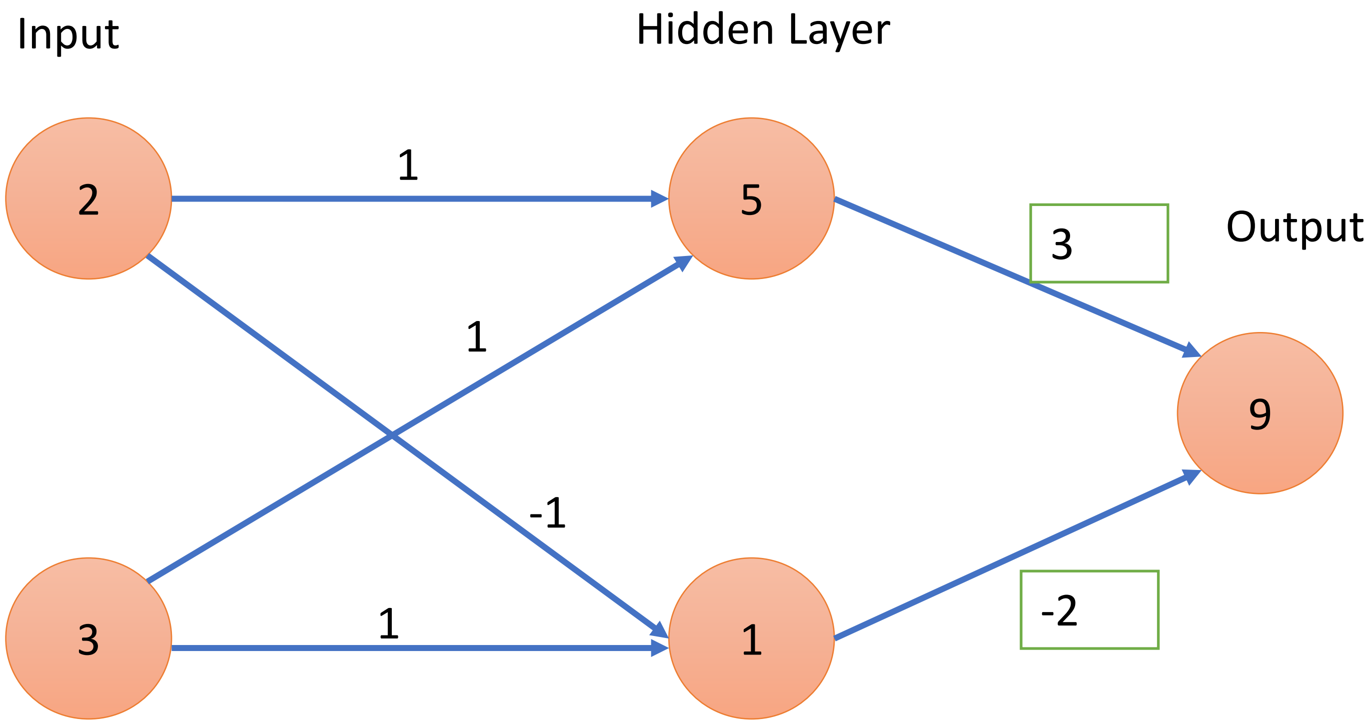
Classes

Prediction

$$CE = -\sum_{i}^{C} t_i log(s_i)$$

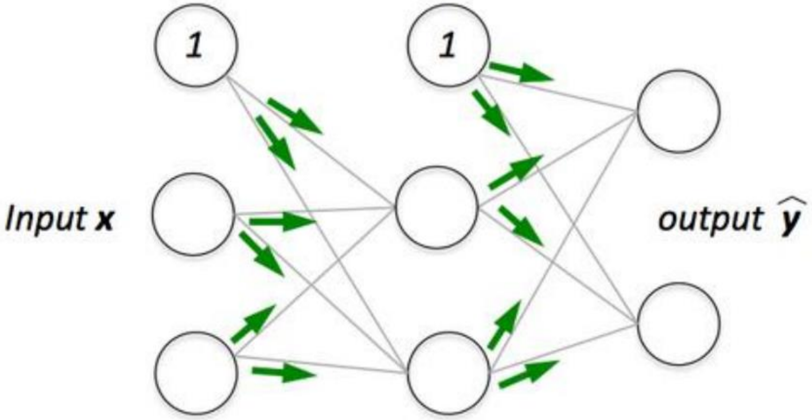
Ground Truth {0,1}

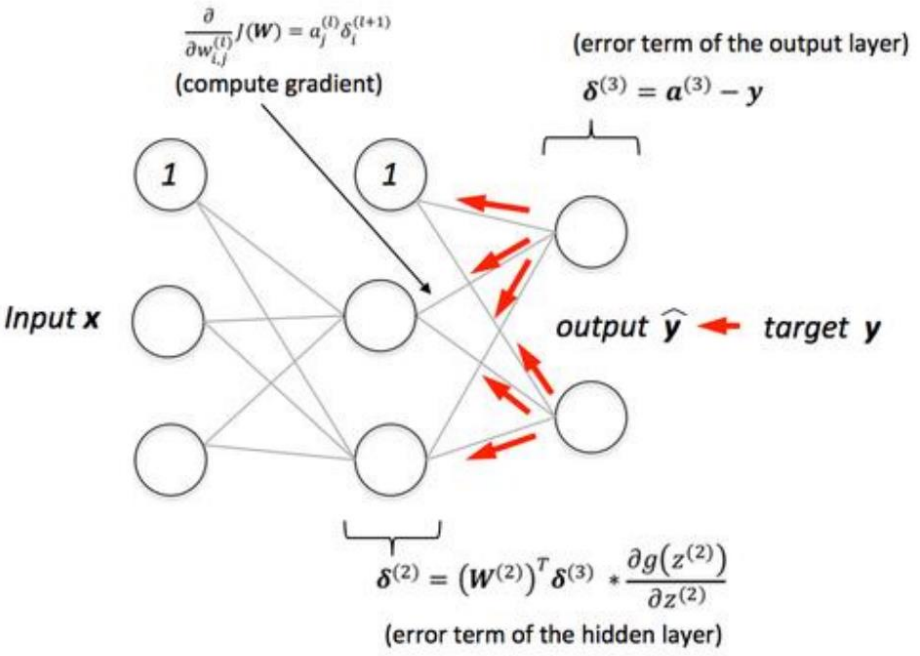
Backpropagation



Actual Value of target : 13
Error: Predicted – Actual = -4

Backpropagation





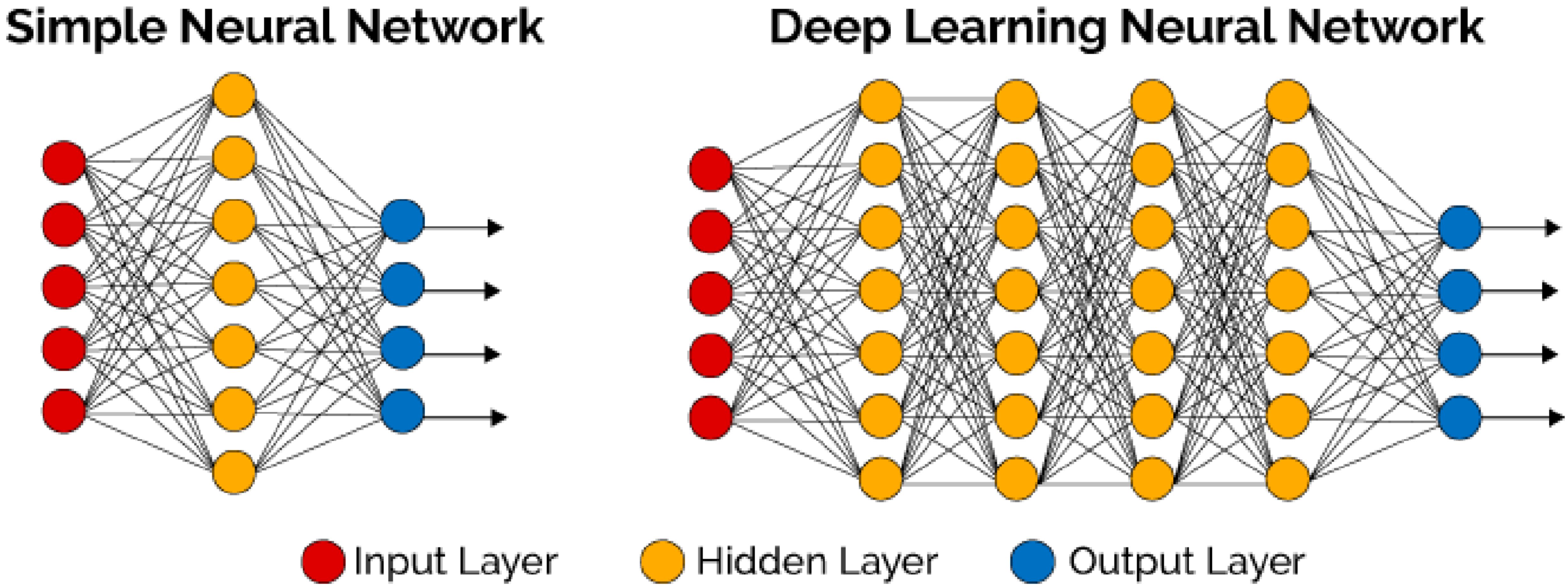
Task: Update the **weights** and **biases** to decrease **loss function**

Subtasks:

1. Forward pass to compute network output and "error"
2. Backward pass to compute gradients
3. A fraction of the weight's gradient is subtracted from the weight.

Learning Rate

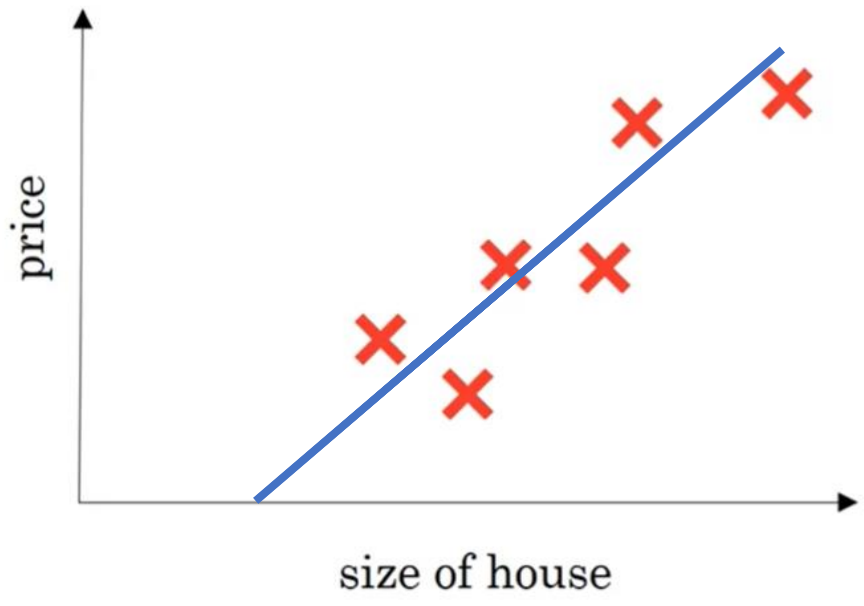
Numerical Method: Automatic Differentiation



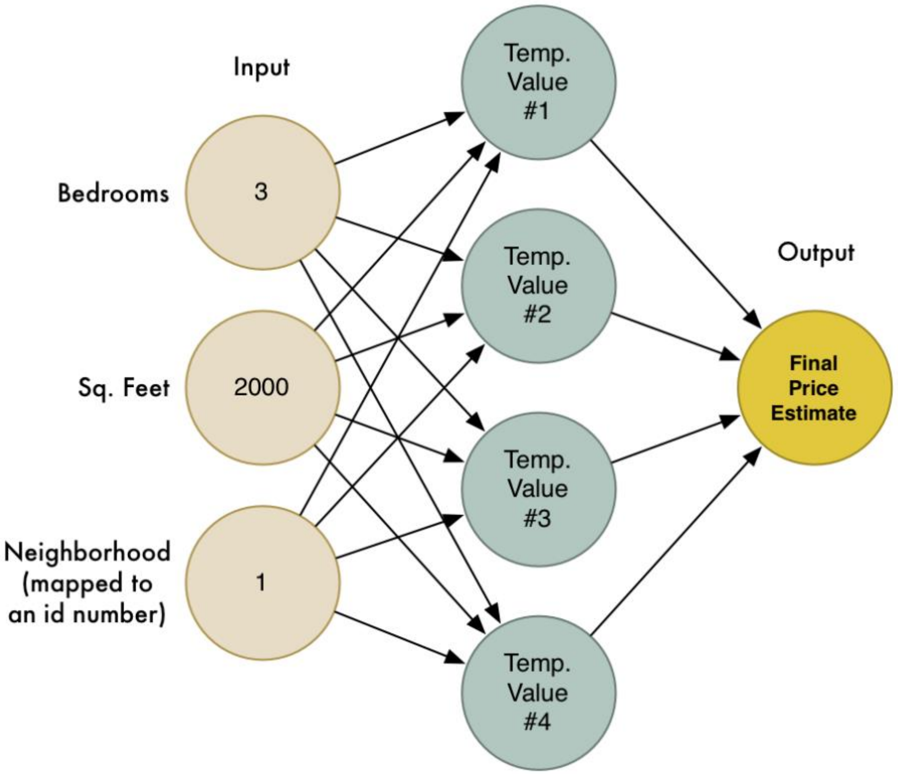
Universality: For any arbitrary function f(x), there exists a neural network that closely approximate it for any input x

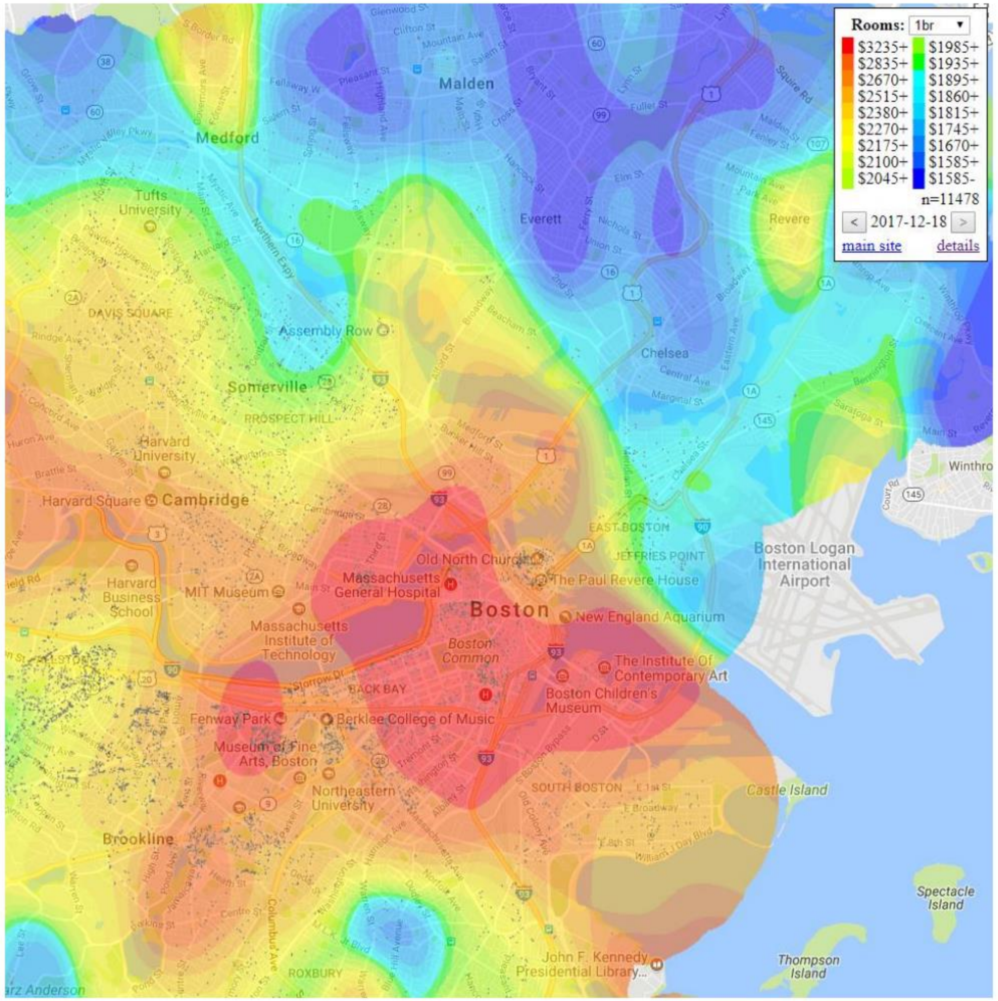
Neural Network

- Housing price prediction

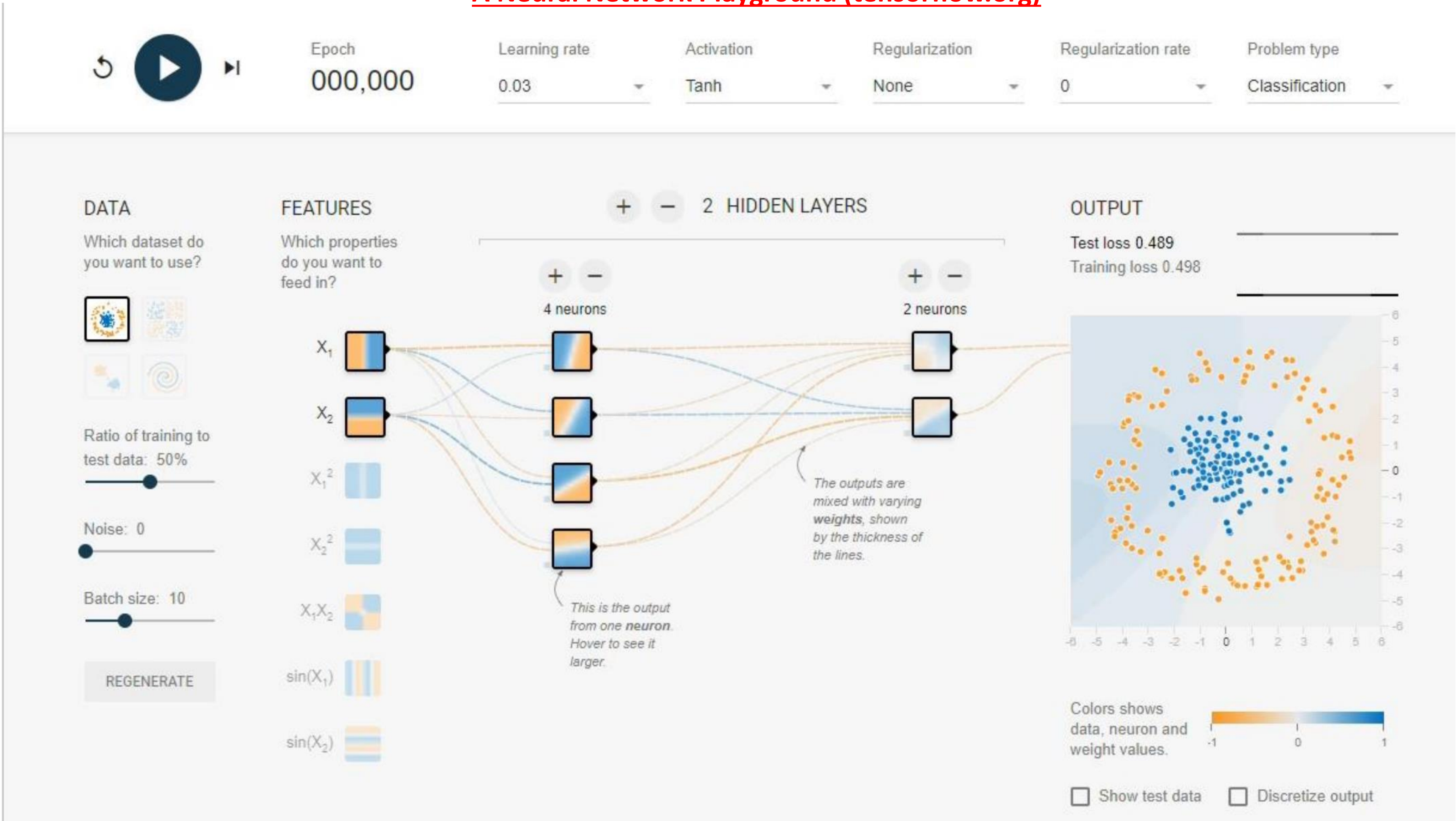


Special Purpose Intelligence:
Estimating Apartment Cost





A Neural Network Playground (tensorflow.org)

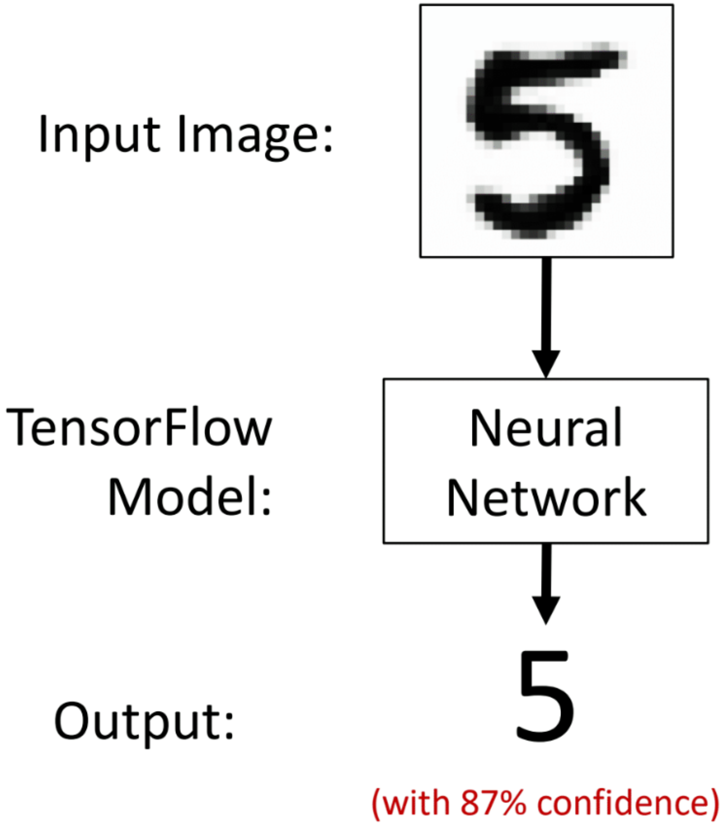


TensorFlow in One Slide

- **What is it:** Deep Learning Library *(and more)*
  - **Facts:** Open Source, Python, Google
- **Community:**
  - 117,000+ GitHub stars
  - TensorFlow.org: Blogs, Documentation, DevSummit, YouTube talks
- **Ecosystem:**
  - Keras: high-level API
  - TensorFlow.js: in the browser
  - TensorFlow Lite: on the phone
  - Colaboratory: in the cloud
  - TPU: optimized hardware
  - TensorBoard: visualization
  - TensorFlow Hub: graph modules
- **Alternatives**: PyTorch, MXNet, CNTK

Extras:

- Swift for TensorFlow
- TensorFlow Serving
- TensorFlow Extended (TFX)
- TensorFlow Probability
- Tensor2Tensor



1

```python
# import tensorflow and keras (tf.keras not "vanilla" Keras)
import tensorflow as tf
from tensorflow import keras
```

2

```python
# get data
(train_images, train_labels), (test_images, test_labels) = \
keras.datasets.mnist.load_data()
```

3

```python
# setup model
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer=tf.train.AdamOptimizer(),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

4

```python
# train model
model.fit(train_images, train_labels, epochs=5)
```

5

```python
# evaluate
test_loss, test_acc = model.evaluate(test_images, test_labels)
print('test accuracy:', test_acc)
```

6

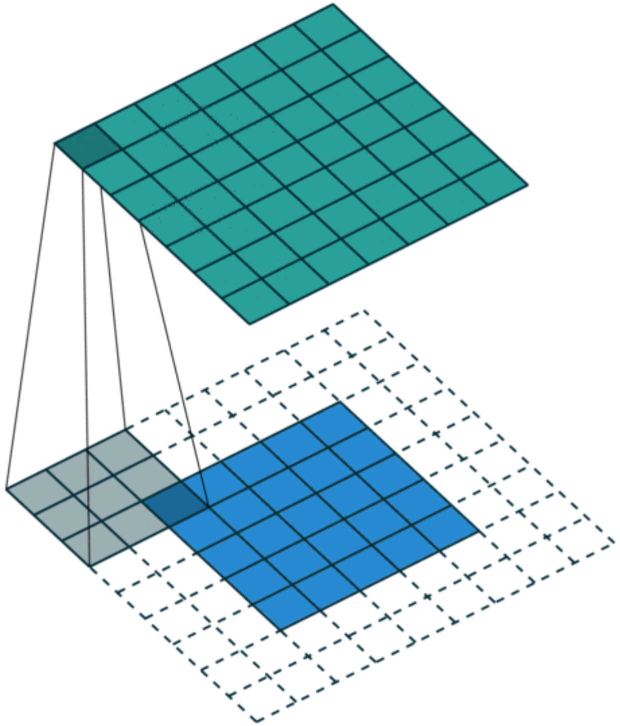
```python
# make predictions
predictions = model.predict(test_images)
```

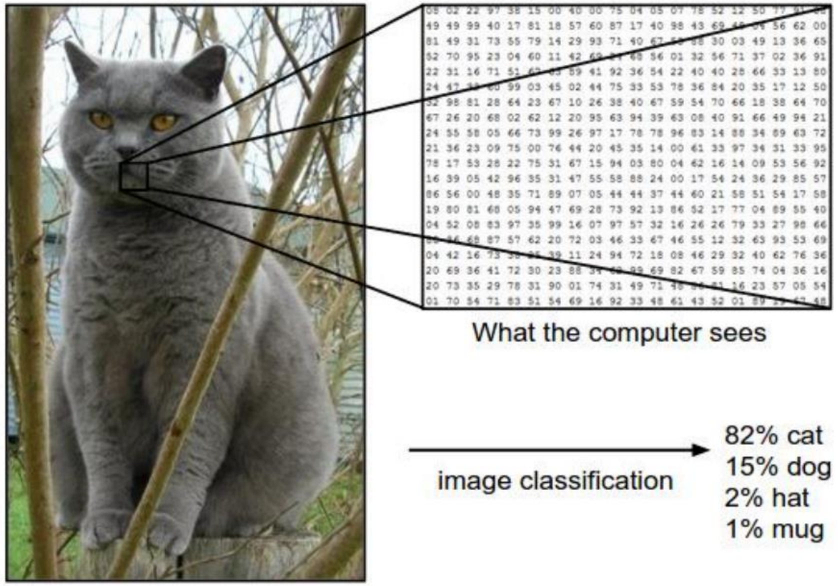
Deep learning models:

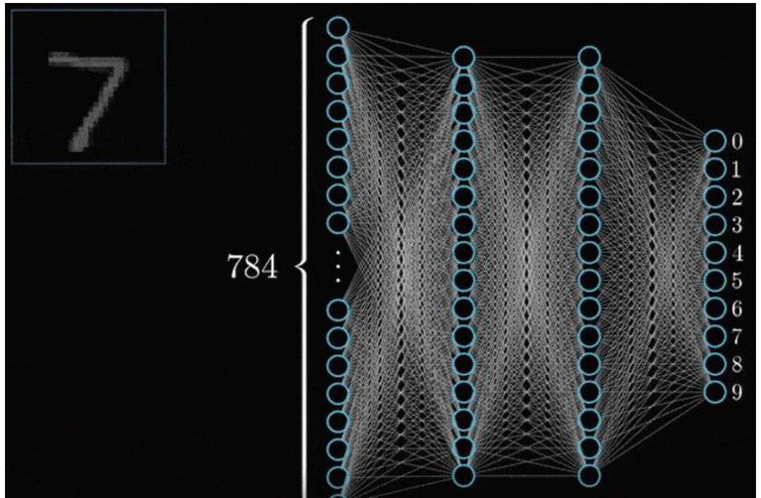
Some of the top Deep Learning Algorithms and Models include the following:

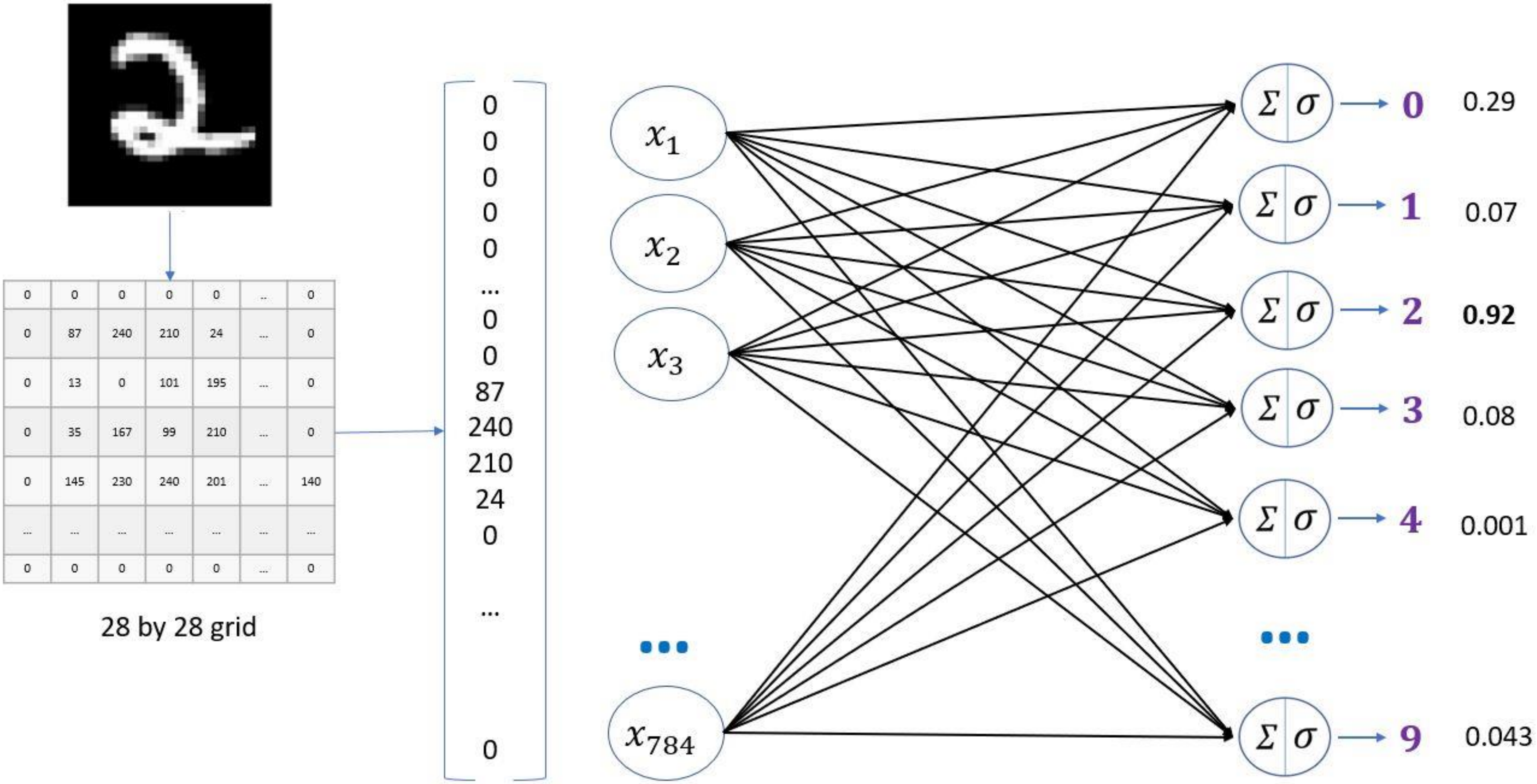
Convolutional Neural Network (CNN)
Radial Basis Function Networks (RBFNs)
Multilayer Perceptrons (MLPs)
Self Organizing Maps (SOMs)
Deep Belief Networks (DBNs)
Restricted Boltzmann Machines (RBMs)
Autoencoders
Convolutional Neural Networks (CNNs)
Long Short-Term Memory Networks (LSTMs)
Recurrent Neural Networks (RNNs) tx
Generative Adversarial Networks (GANs)

Convolutional Neural Networks: Image Classification





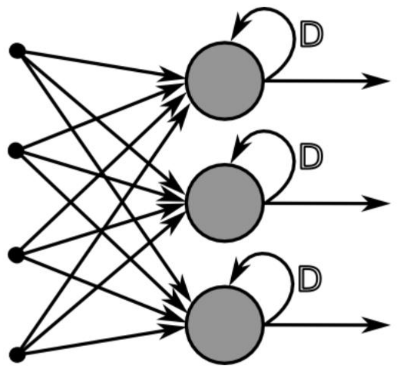


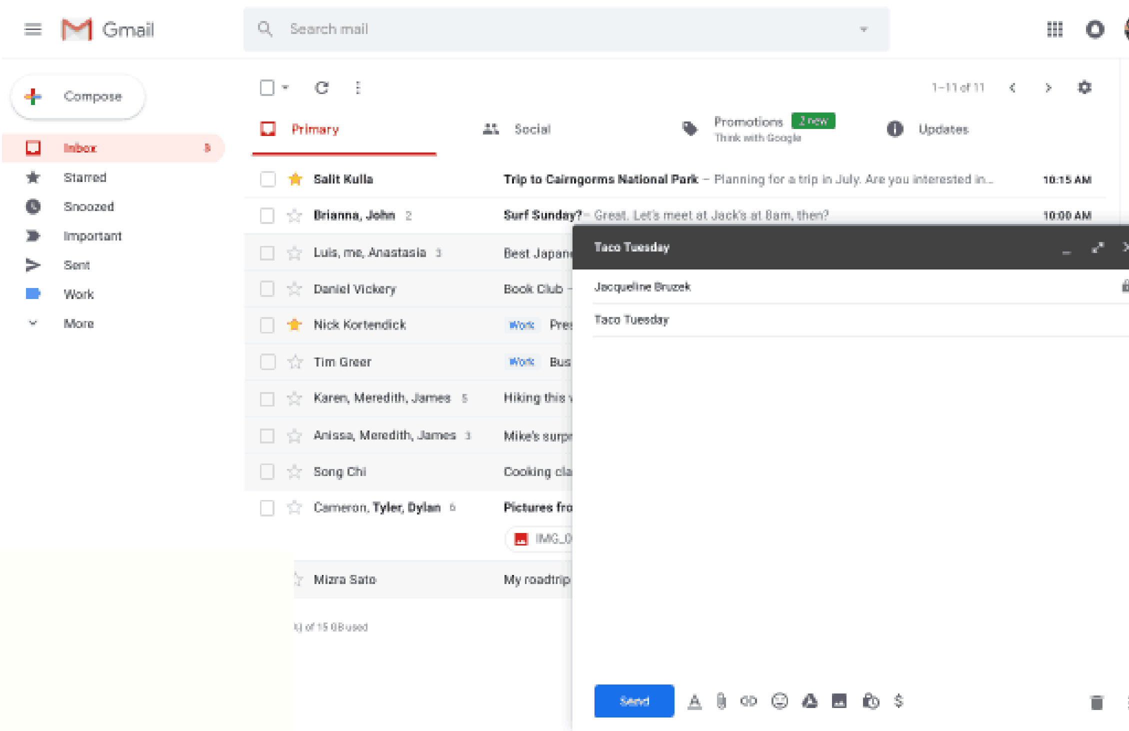


0 0 0 0 0 ... 0
0 87 240 210 24 ... 0
0 13 0 101 195 ... 0
0 35 167 99 210 ... 0
0 145 230 240 201 ... 140
... ...
0 0 0 0 0 ... 0
28 by 28 grid
0 0 0 0 0 ... 0 0 87 240 210 24 0 ... 0
x_1
x_2
x_3
x_{784}
Σ σ
0 0.29
1 0.07
2 0.92
3 0.08
4 0.001
9 0.043

Recurrent Neural Network (RNN)

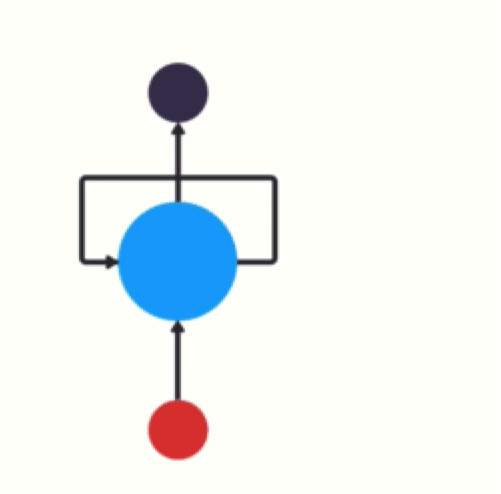
A recurrent neural network (RNN) is **a special type of an artificial neural network adapted to work for time series data or data that involves sequences**

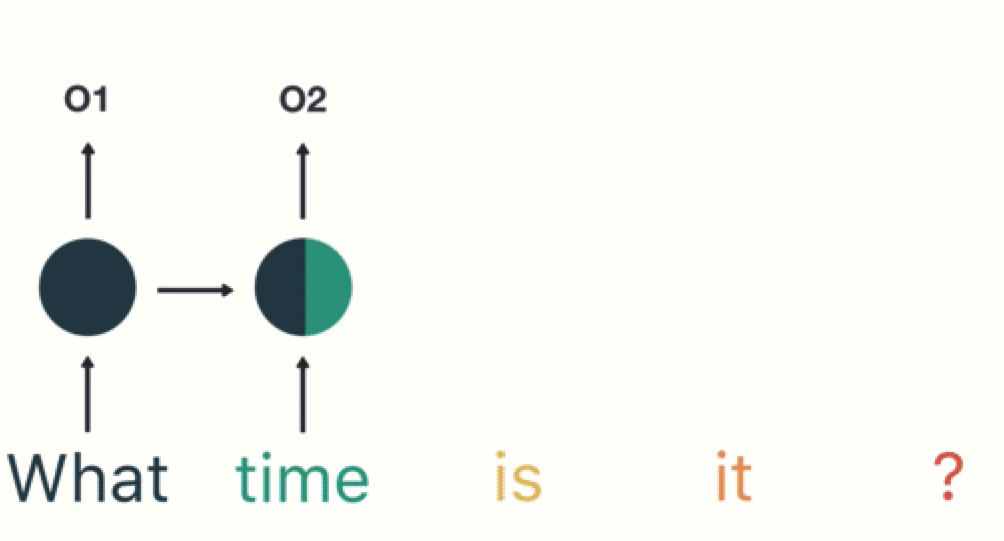




- Applications
  - Sequence Data
  - Text
  - Speech
  - Audio
  - Video
  - Generation

How do we get a feed-forward neural network to be able to use previous information to effect later ones?
What if we add a loop in the neural network that can pass prior information forward?





Refences and materials to read

- 3Blue1Brown – YouTube
- codebasics – YouTube
- Lex Fridman - YouTube