AI Bootcamp

Neural Networks

Module 18 Day 1

© 2023 edX Boot Camps LLC. Confidential and Proprietary. All Rights Reserved.

Class Objectives

By the end of class, you will be able to:

1. Compare the advantages and disadvantages of using neural network models with other types of machine learning models.
2. Describe the perceptron model and its components.
3. Create, train, and evaluate neural network models using TensorFlow.

Welcome

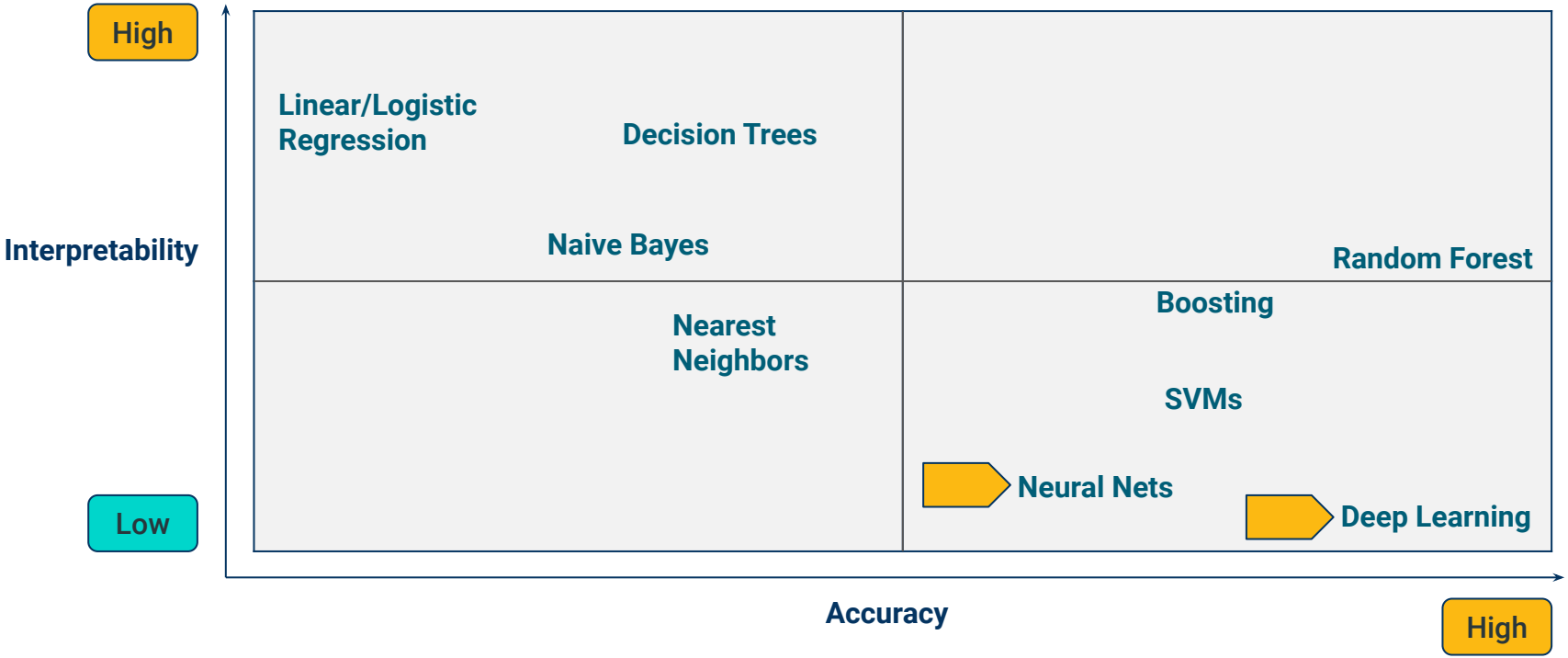
Neural Networks

Here are some examples of neural networks you might have encountered:

- Voice-activated assistants and voice-to-text functionalities such as Siri, Cortana, and Google Assistant are powered by neural networks designed to recognize human speech.
- OpenAI's ChatGPT, capable of generating conversational responses to text prompts, is an innovative application of **natural language processing** (**NLP**), which you will learn about in a later module.
- Self-driving cars and facial recognition both make use of **computer vision** systems that are powered by neural networks and machine learning models to identify features and extract information from images.
- Recommendation engines, such as the ones on Netflix and YouTube, use deep neural networks to understand user behavior and develop personalized content recommendations. Later in this module, we will explore more deep learning networks and recommendation systems.

Neural Net

The following diagram plots the different types of machine learning models:



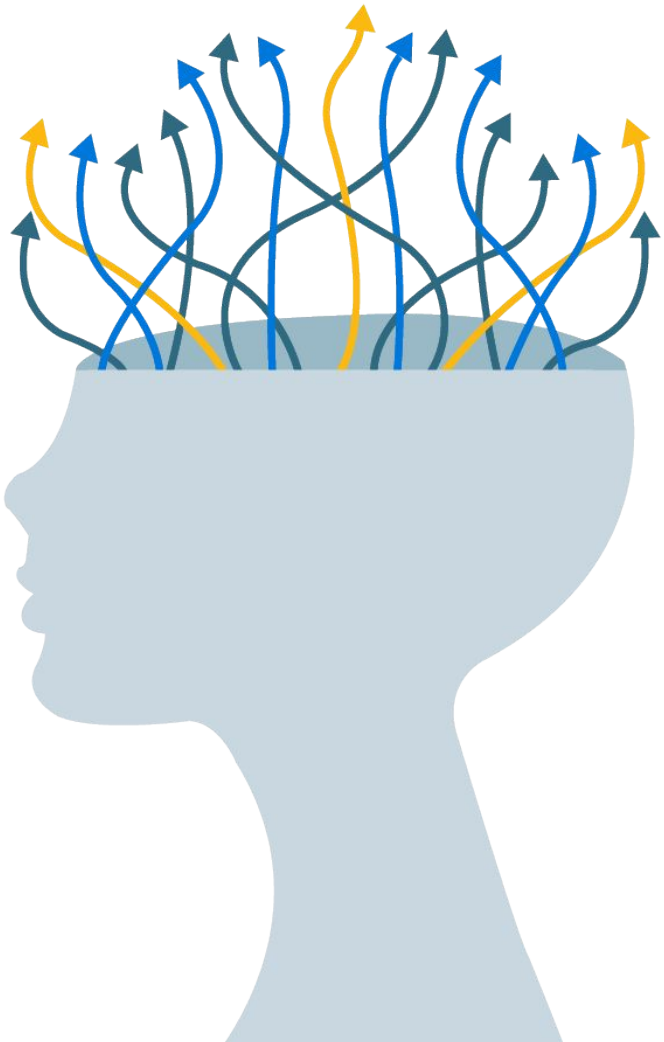
Instructor **Demonstration**

What is a Neural Network?

What is a neural network?

Neural networks are a powerful machine learning technique modeled after neurons in the brain.

Industry leaders such as Google, Facebook, Twitter, and Amazon use neural networks for analyzing complex datasets.



Artificial Neural Network (ANN) models

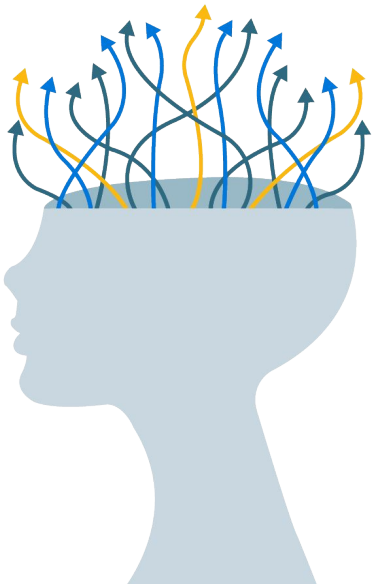
Transformers

Recurrent Neural Networks (RNNs)

Perceptron

Multilayer Perceptron

Large Language Models (LLMs)



Long-Short Term Memory models (LSTMs)

Restricted Boltzmann Machine (RBM)

Convolutional Neural Networks (CNNs)

Neural Network is an advanced form of machine learning that contains multiple layers of nodes which perform individual computations.

The MNIST Dataset

The MNIST (Modified National Institute of Standards and Technology) dataset contains black and white images of handwritten numbers.

A neural network can train on each pixel of each image as a scaled value from zero (completely white) to one (completely black).

With enough data points, a trained neural network model can classify handwritten numbers with a high degree of accuracy.

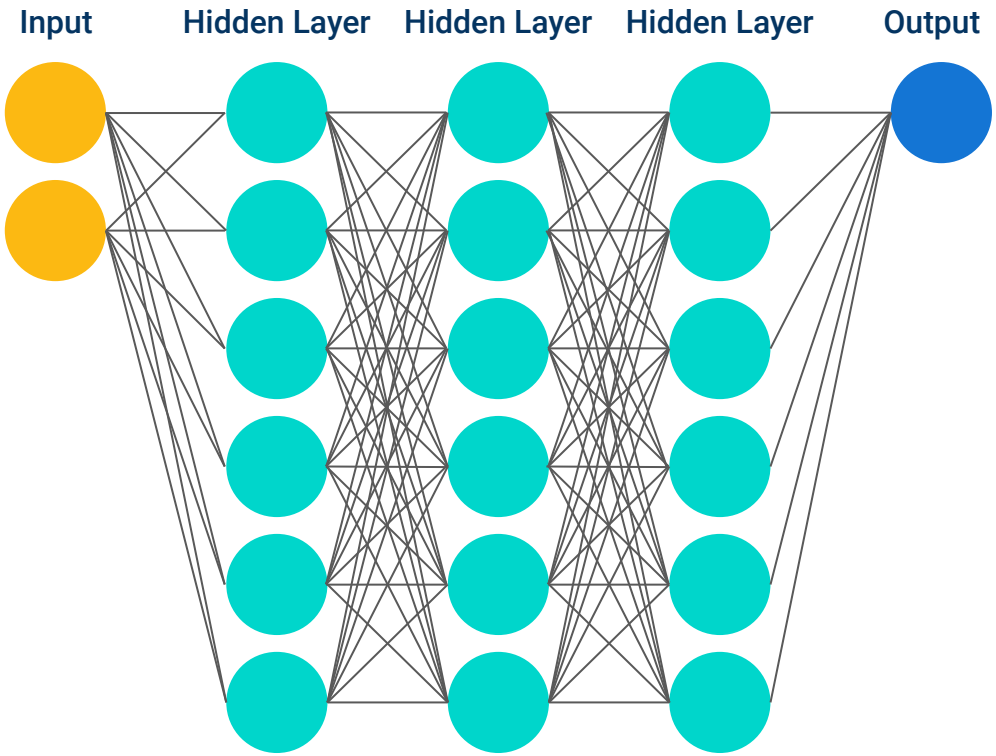
Sample images from MNIST test dataset

Neural Net

Example diagram of a neural network:

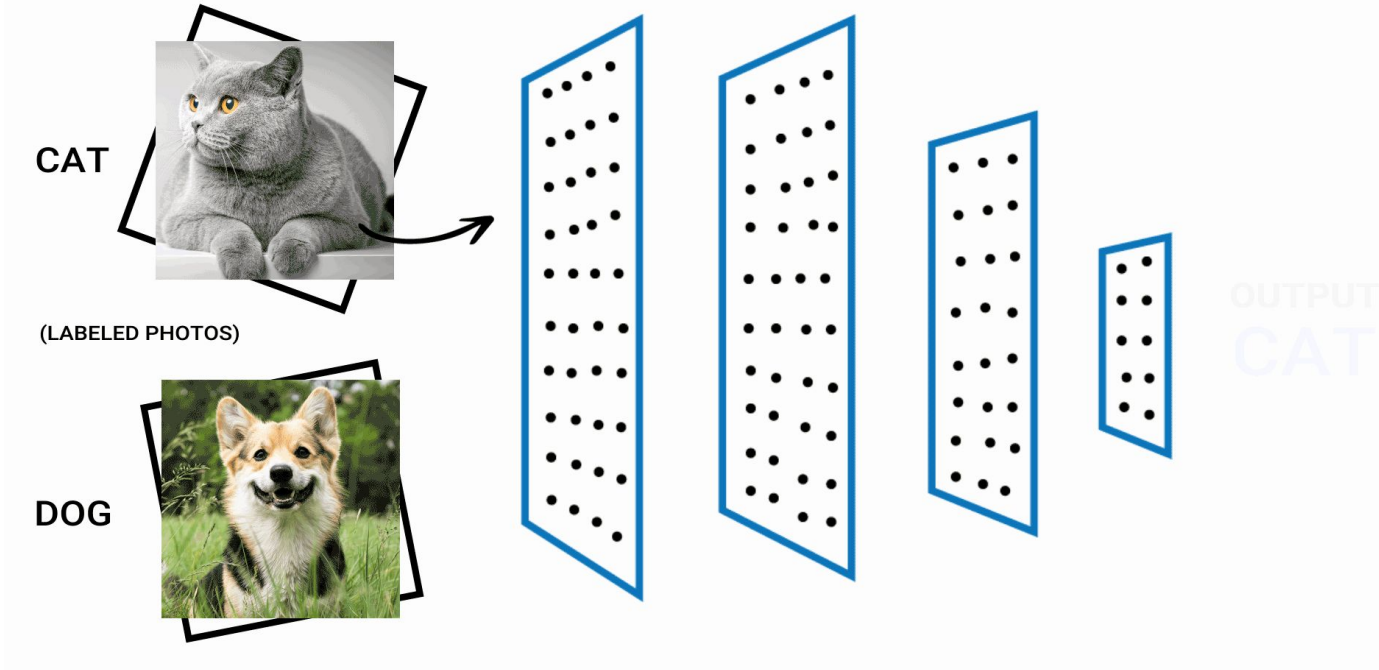
A neural network can train on each pixel of each image as a scaled value from 0 (completely white) to 1 (completely black).

With enough data points, a trained neural network model can classify handwritten numbers with a high degree of accuracy.



What is a neural network?

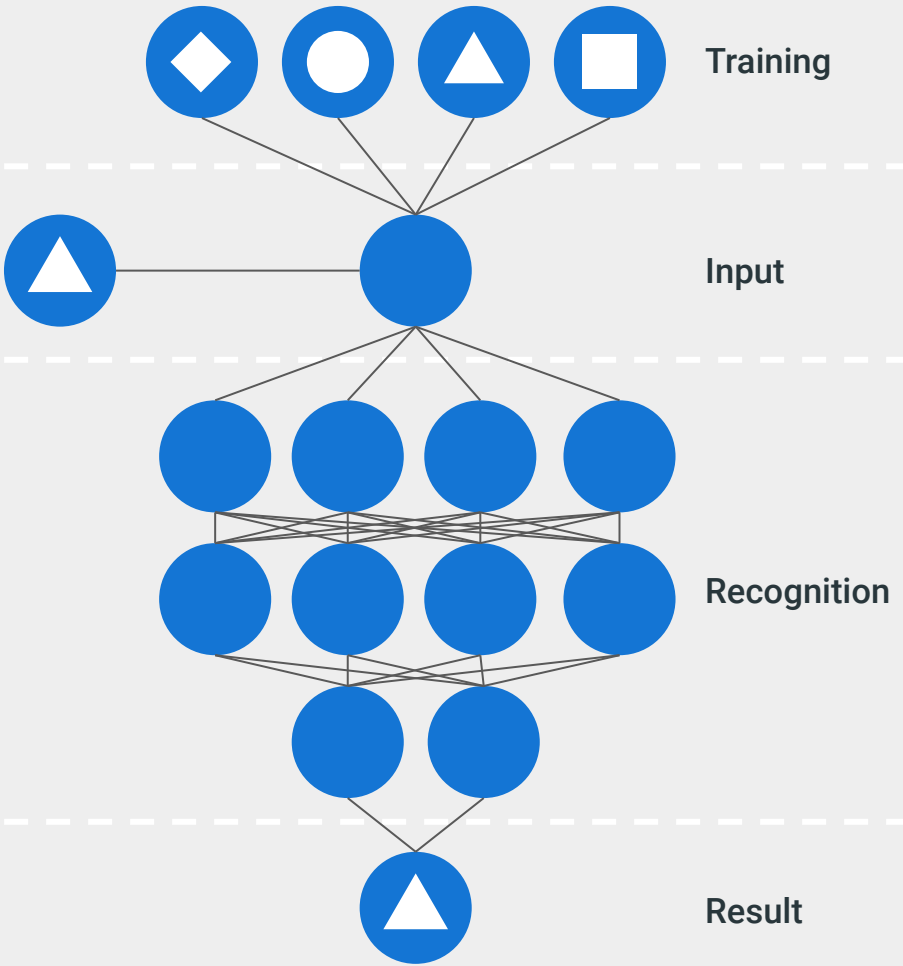
These computations are connected and weighed against one another until the neurons reach the final layer. In the final layer, the neurons return either a numerical result or an encoded categorical result.



What is a neural network?

Neural networks are an advanced form of machine learning that:

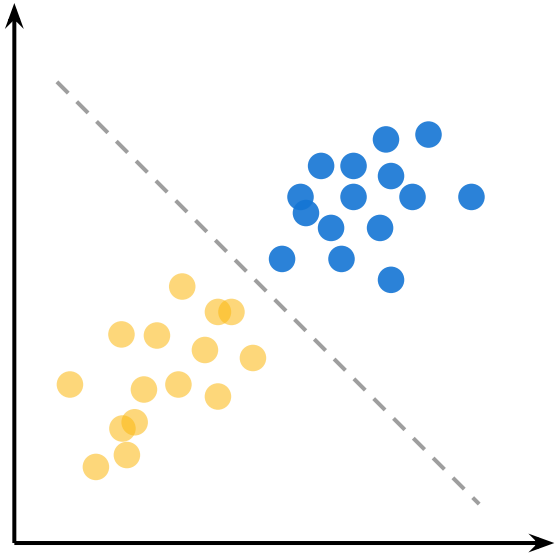
- Recognizes patterns and features of input data
- Provides a clear quantitative output



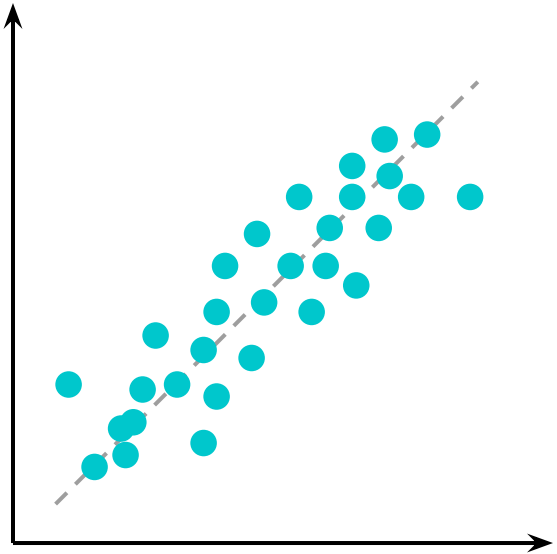
Neural Net

Neural networks are very useful in data science because they serve multiple purposes. Two popular uses include:

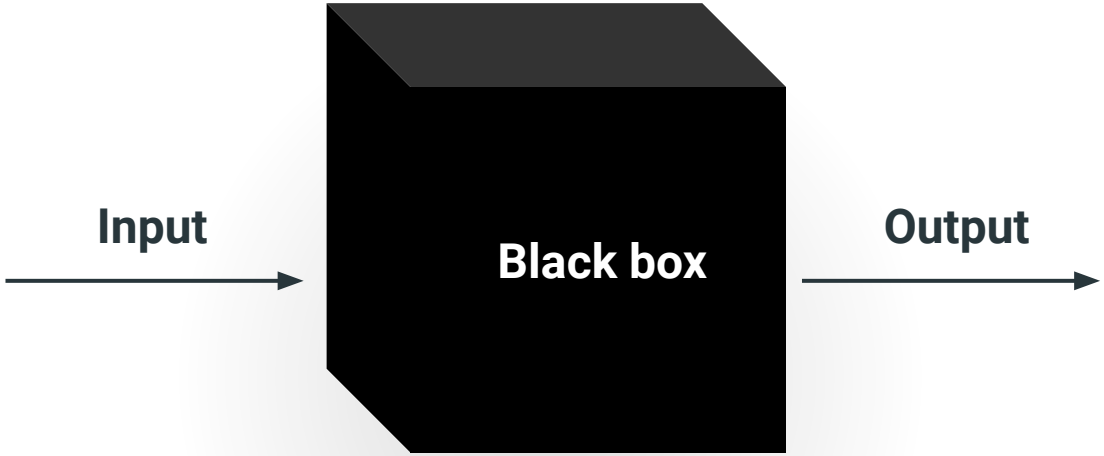
01 **Classification** algorithms that determine how to categorize an input.



02 **Regression** models that predict a dependent output from independent input variables.



The Black Box Problem



Activity:

Working Through the Logistics

In this activity, you will use the logistic regression to build a binary classification model, which is the precursor to neural networks.

Suggested Time:
15 Minutes

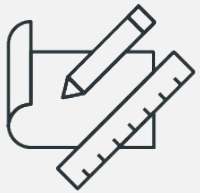
Time’s up!

Let’s review

Questions?

Instructor **Demonstration**

Perceptron



Perceptron,
the Computational Neuron

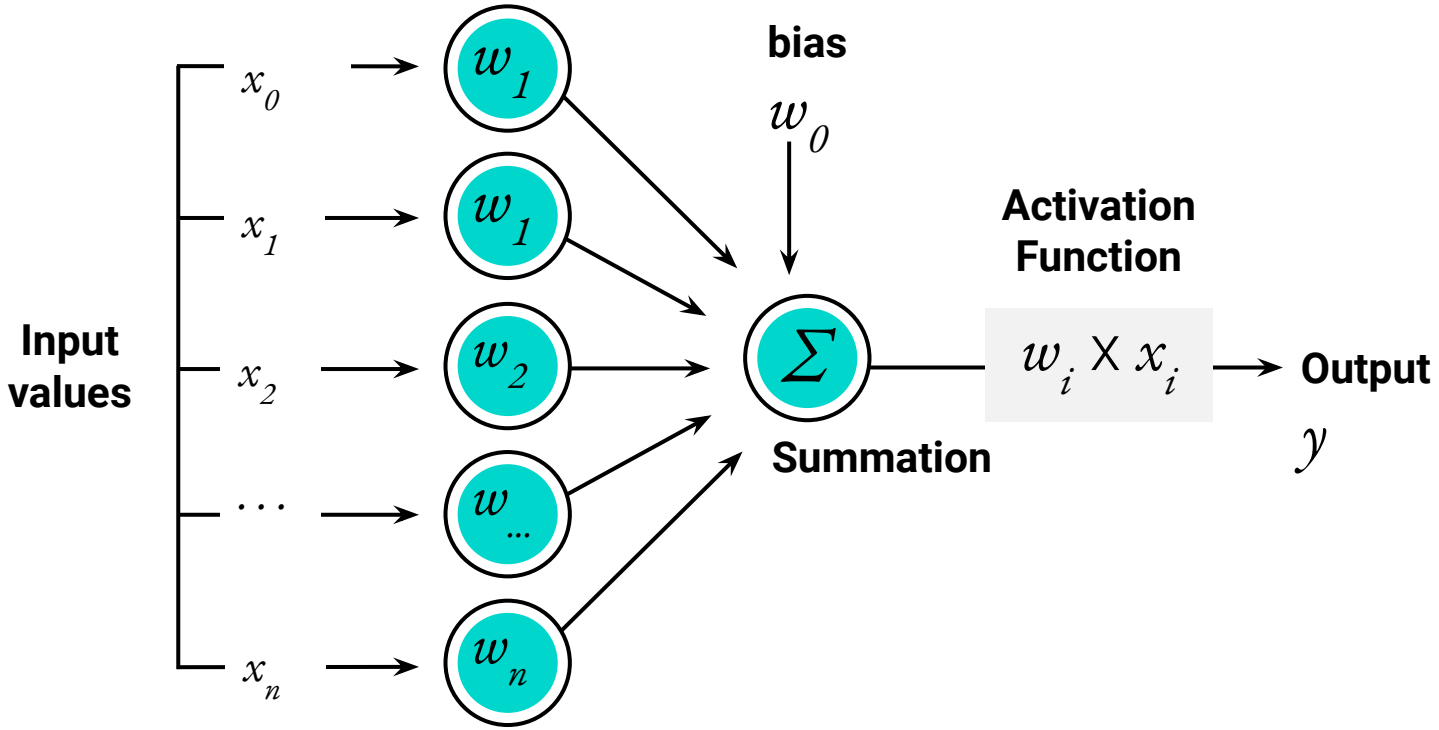
Artificial neural networks have become popular in recent years. But, the original design for a computational neuron, known as a **perceptron**, dates back to the late 1950s.

Perceptron was introduced by Frank Rosenblatt in 1957. He proposed a perceptron learning rule based on the original MCP neuron.

A **perceptron** is an algorithm for supervised learning of binary classifiers. This algorithm enables neurons to learn and processes elements in the training set one at a time.

Meet the Perceptron

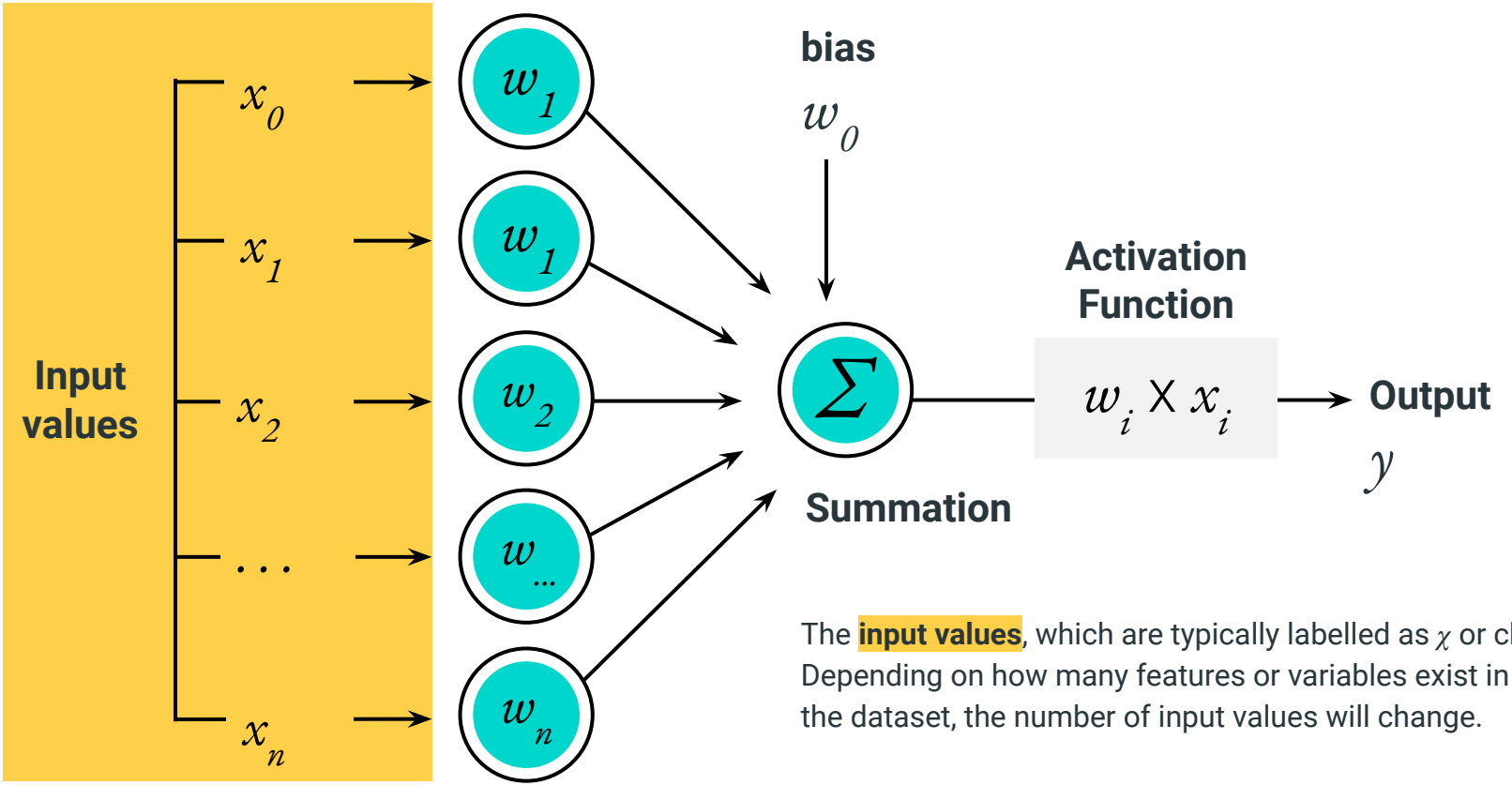
The **perceptron model** mimics a biological neuron by receiving input data, weighting the information, and producing a clear output.



The perceptron model has **four major components**.

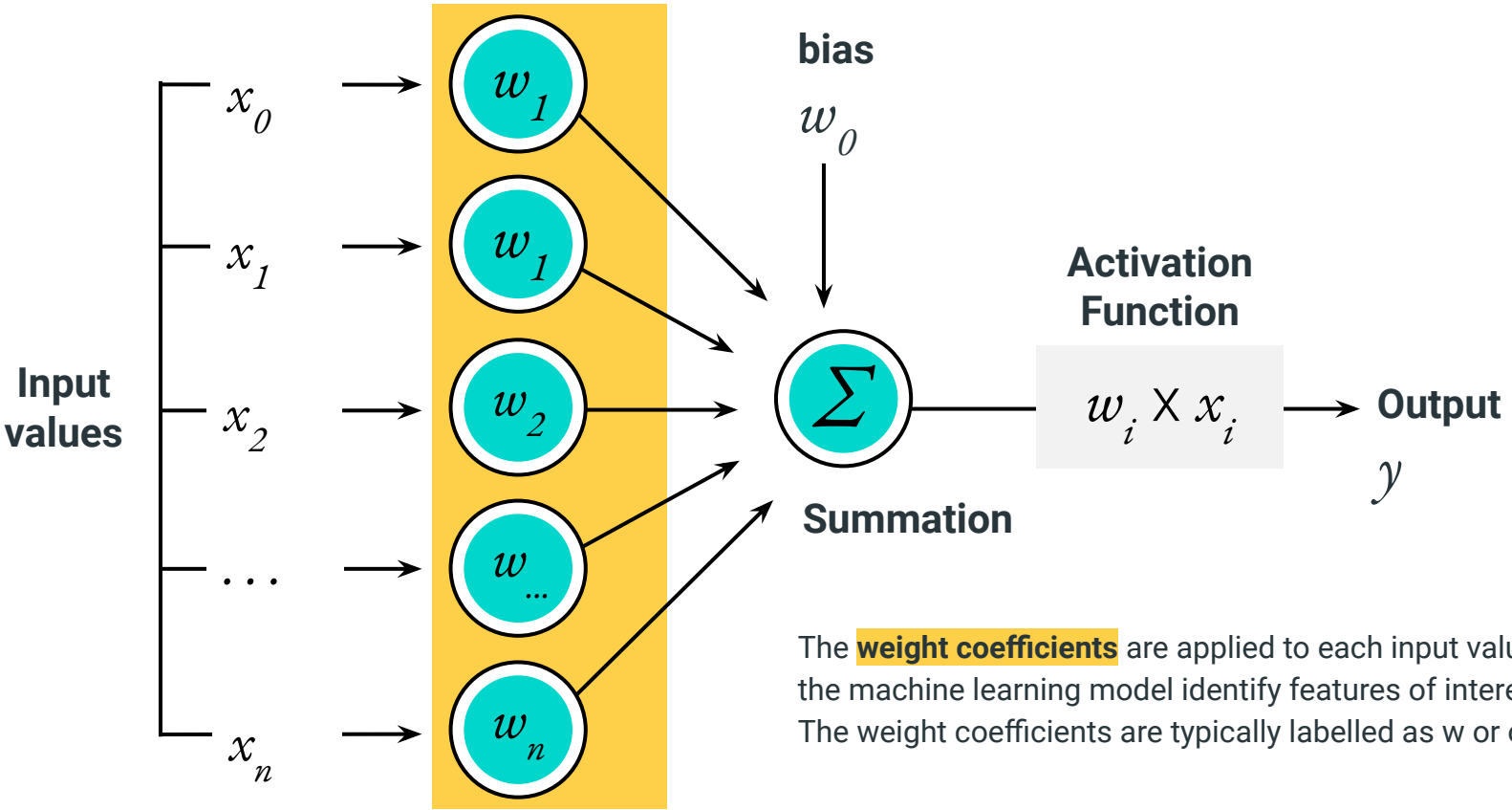
Input values (shown as *x*'s)

The **perceptron model** mimics a biological neuron by receiving input data, weighting the information, and producing a clear output.



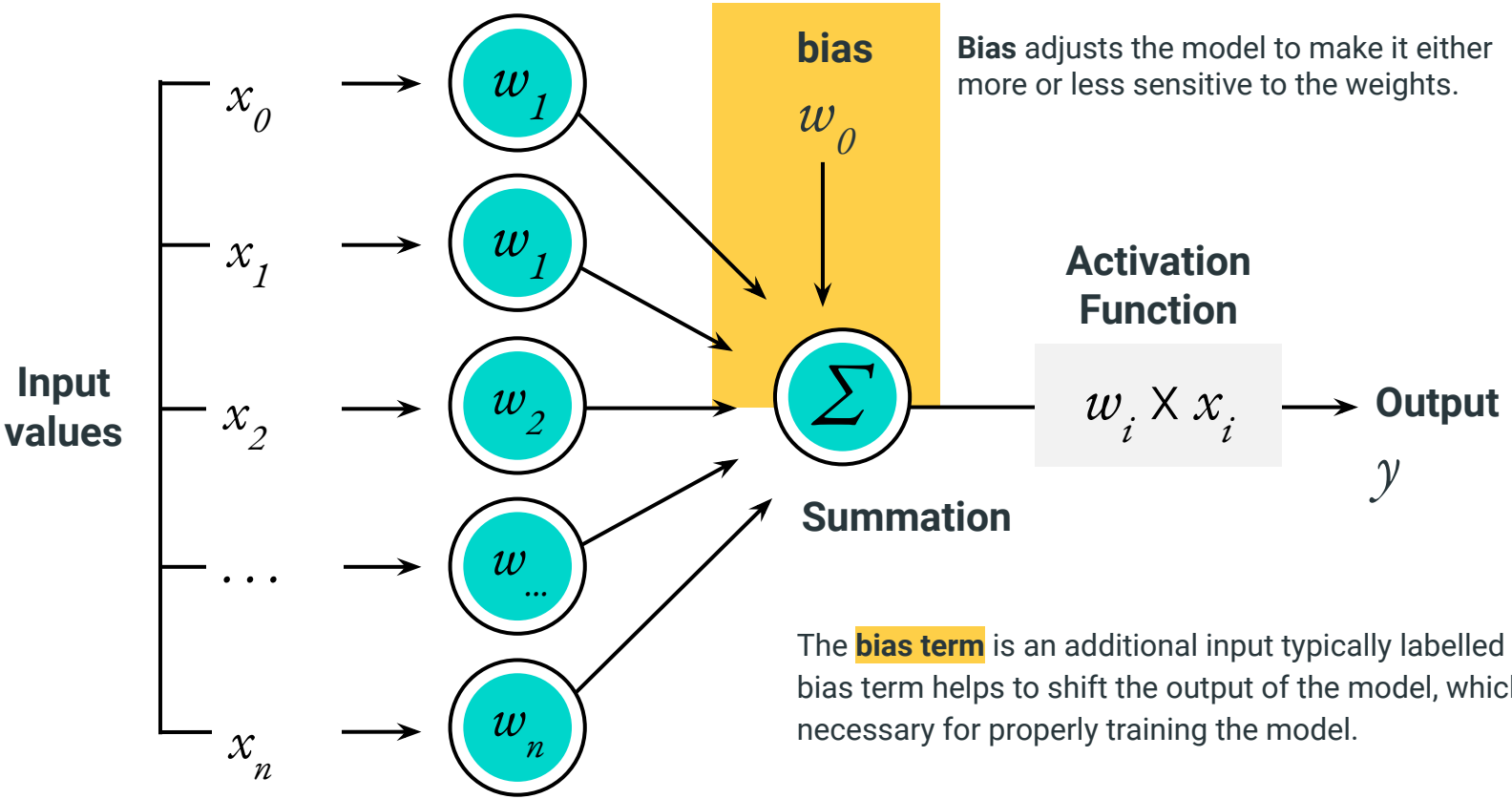
The **input values**, which are typically labelled as χ or chi. Depending on how many features or variables exist in the dataset, the number of input values will change.

Weight coefficients (denoted as *w*'s)



The **weight coefficients** are applied to each input value to help the machine learning model identify features of interest.
The weight coefficients are typically labelled as w or omega.

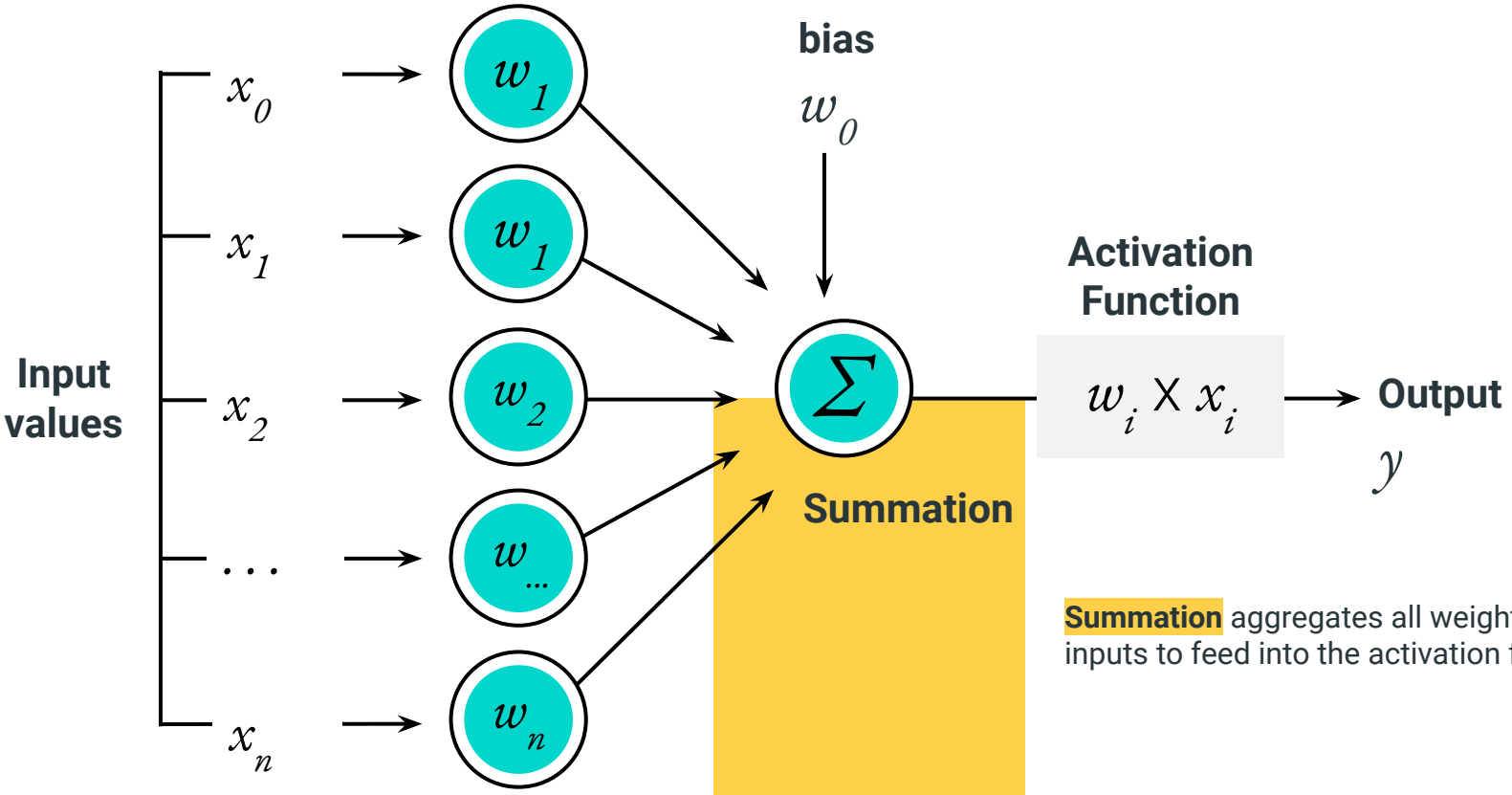
A constant value called bias



Bias adjusts the model to make it either more or less sensitive to the weights.

The **bias term** is an additional input typically labelled as ω. The bias term helps to shift the output of the model, which may be necessary for properly training the model.

Net summary function (the summation)



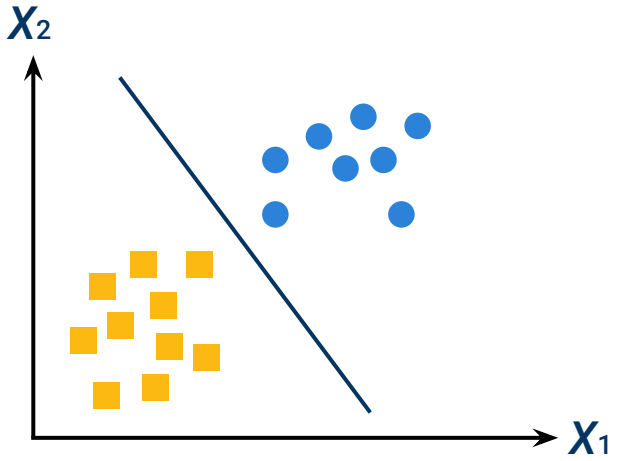
Summation aggregates all weighted inputs to feed into the activation function.

A perceptron model—also known as a **linear binary classifier**—contains a linear equation that separates data into two groups. This is the most common use.

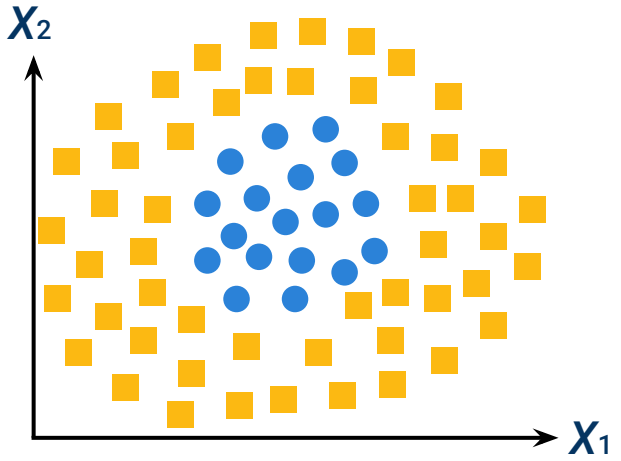
Linearly Separable

Consider linearly separable: the perceptron algorithms separate and classify the data into two groups using linear equation.

Linearly Separable



Not Linearly Separable



Since we provide the model our input features and parameters, the perceptron model is a form of **supervised machine learning**.

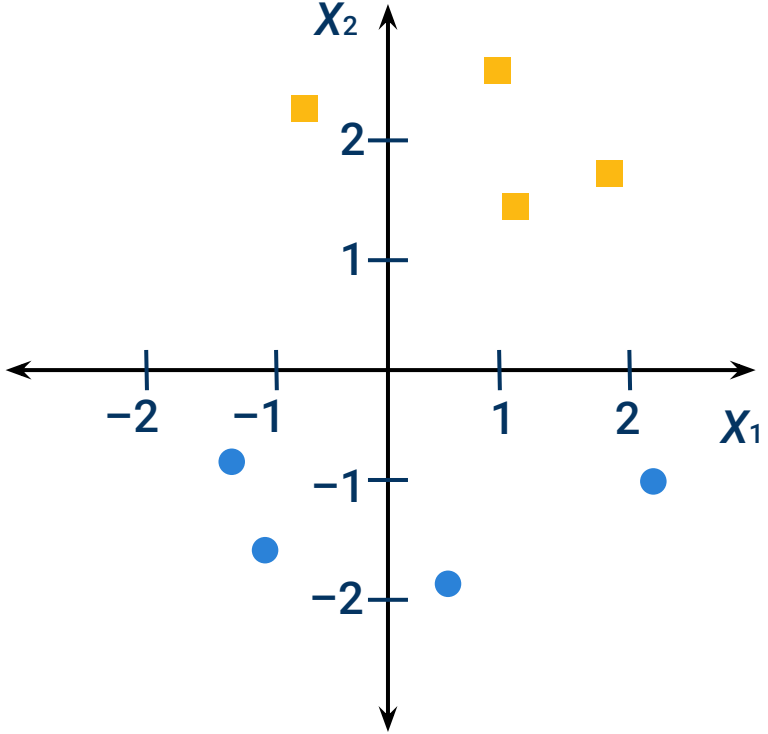
Perceptron Example

Our model will try to classify values in a two-dimensional space; our perceptron model will use three inputs:

x_1	the *x* value
x_2	the *y* value
w_0	the bias constant

The end result of our two-dimensional perceptron model is the net sum function:

$$w_0 + x_1 w_1 + x_2 w_2.$$

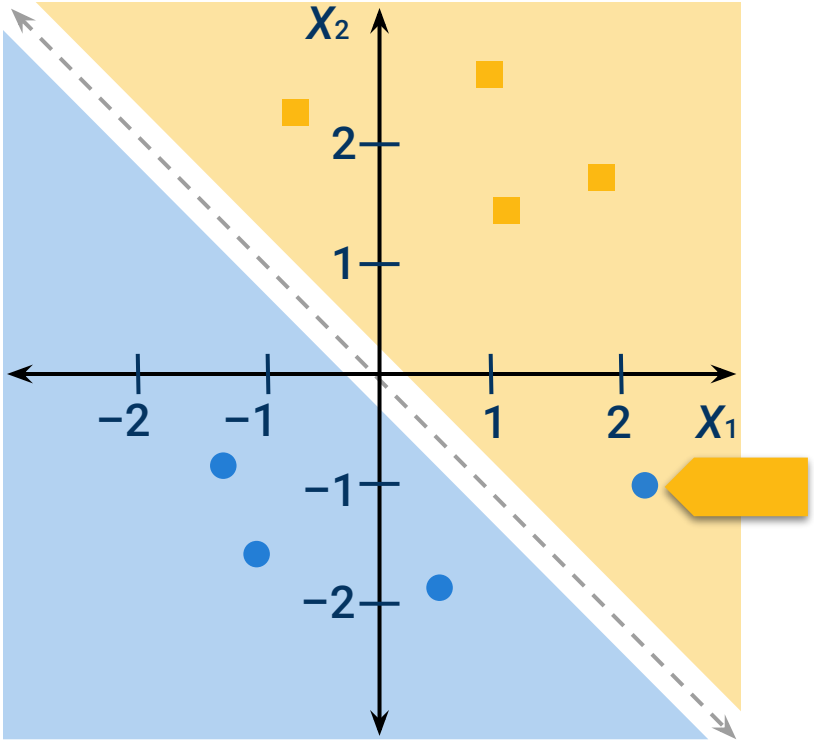


If a data point is misclassified, the weights will move the model closer to the **missed** data point.

Untrained Perceptron Model on Our Dataset

An untrained model almost classified the two groups perfectly.

It misclassified one blue circle.

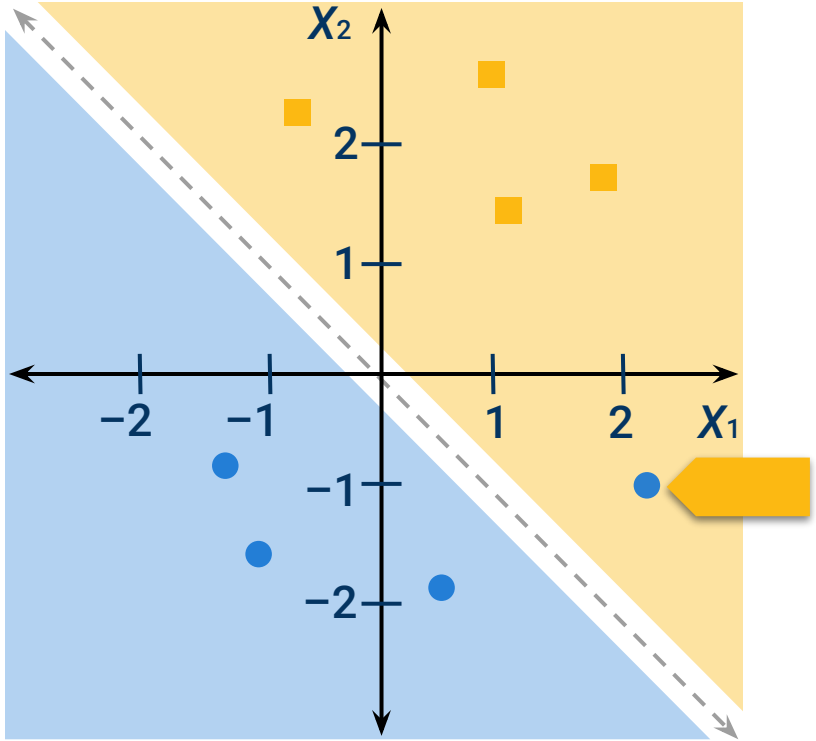


Untrained Perceptron Model on Our Dataset

The perceptron model will evaluate each data point and determine if the input weights should change.

If a data point is classified correctly, the weights will not change.

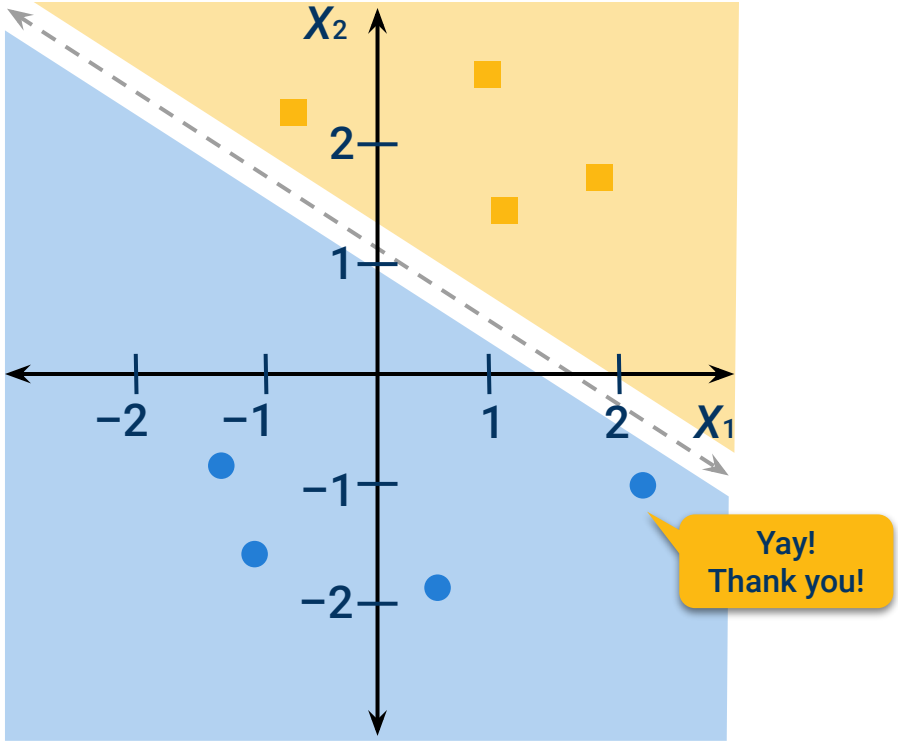
If a data point is misclassified, the weights will move the model closer to the missed data point.



Trained Perceptron Model on Our Dataset

Each data point is evaluated to determine if the input weights should change.

Since a data point is misclassified, the weights will move the model closer to the missed data point.

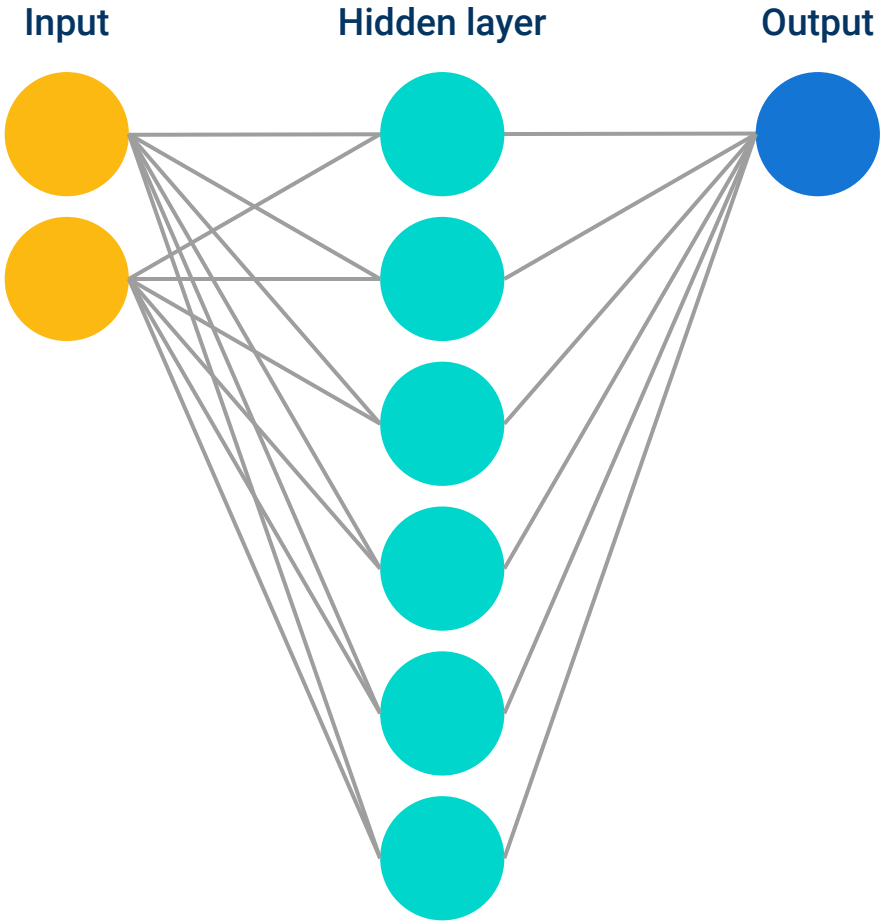


Instructor **Demonstration**

Make the Connections

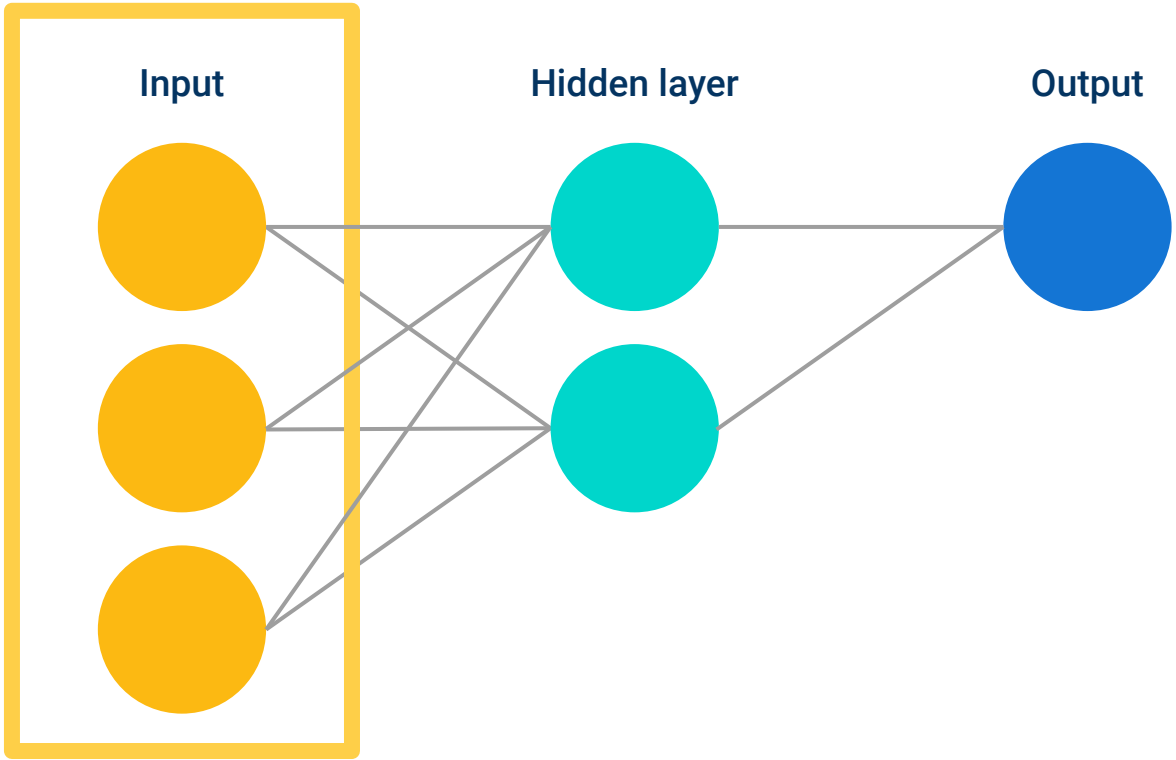
The Neural Network

A modern neural network model is a structure composed of several connected perceptrons that learn from input data to produce an output.



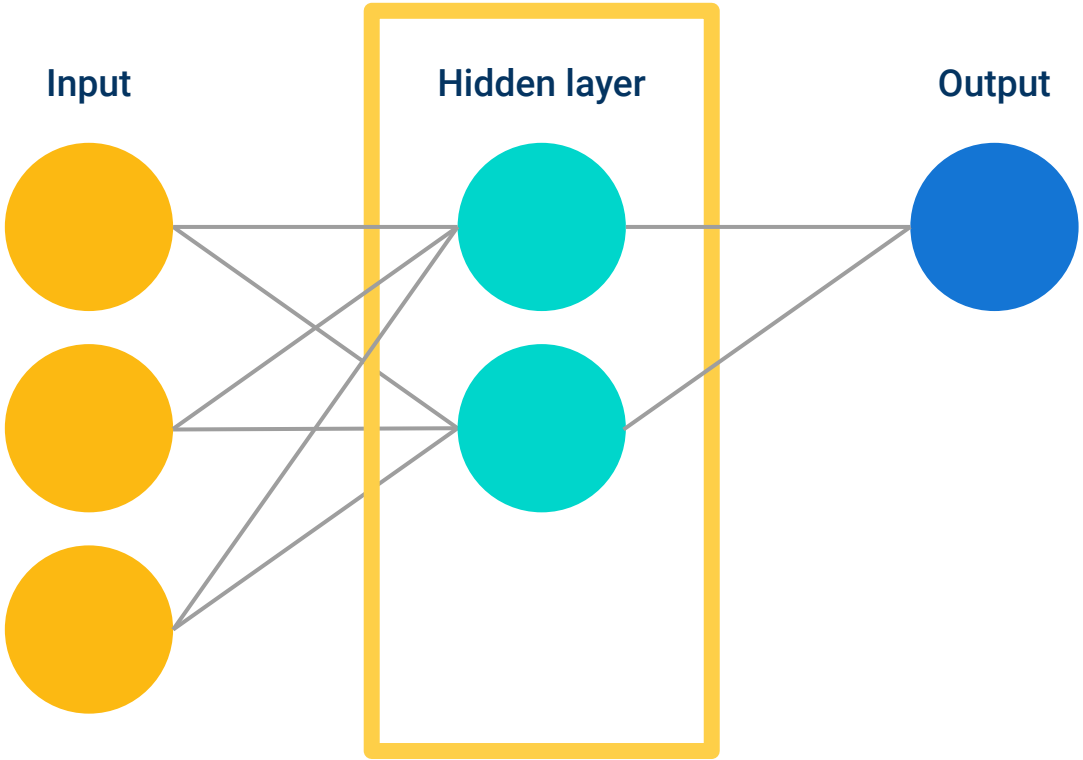
The Structure of a Neural Network

An **input layer** of input values transformed by weight coefficients:



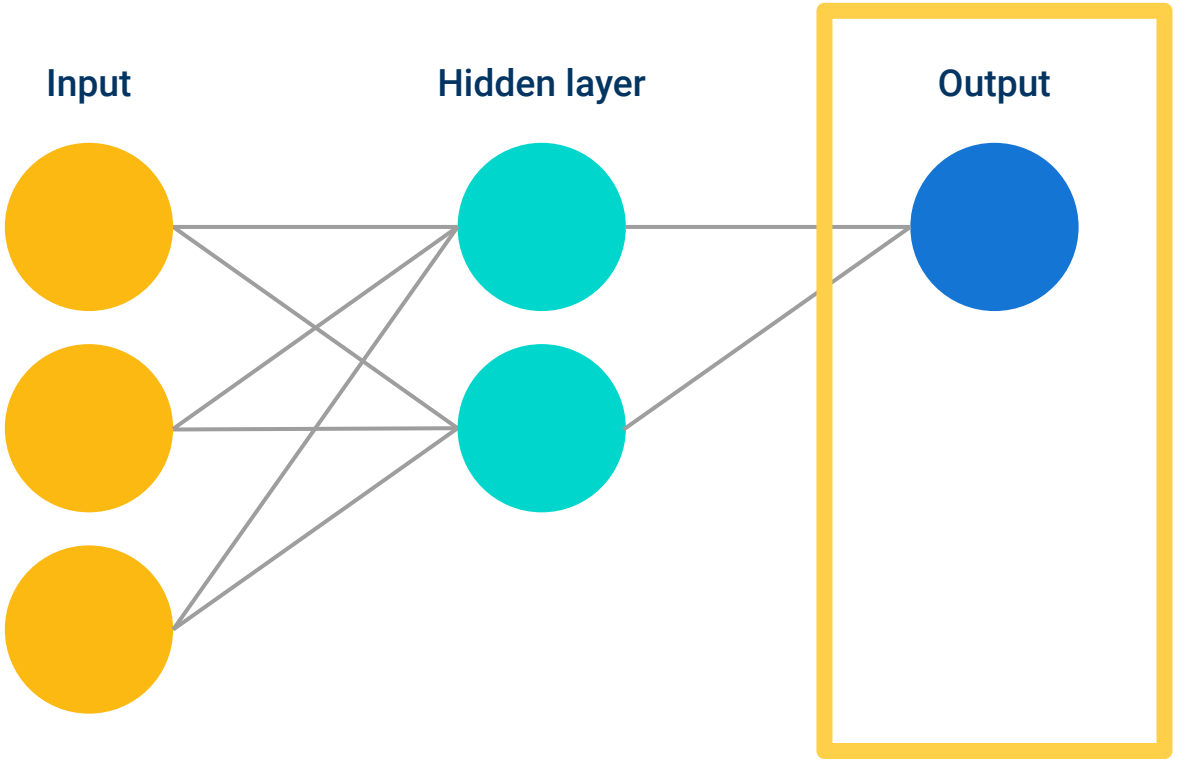
The Structure of a Neural Network

A single **hidden layer** that can contain a single neuron or multiple neurons:

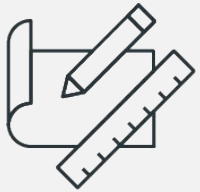


The Structure of a Neural Network

An **output layer** that reports the outcome of the value:



If each neuron has its own output, how does the neural network combine each neuron's output into the model's classification or regression output?



Activation **Functions**

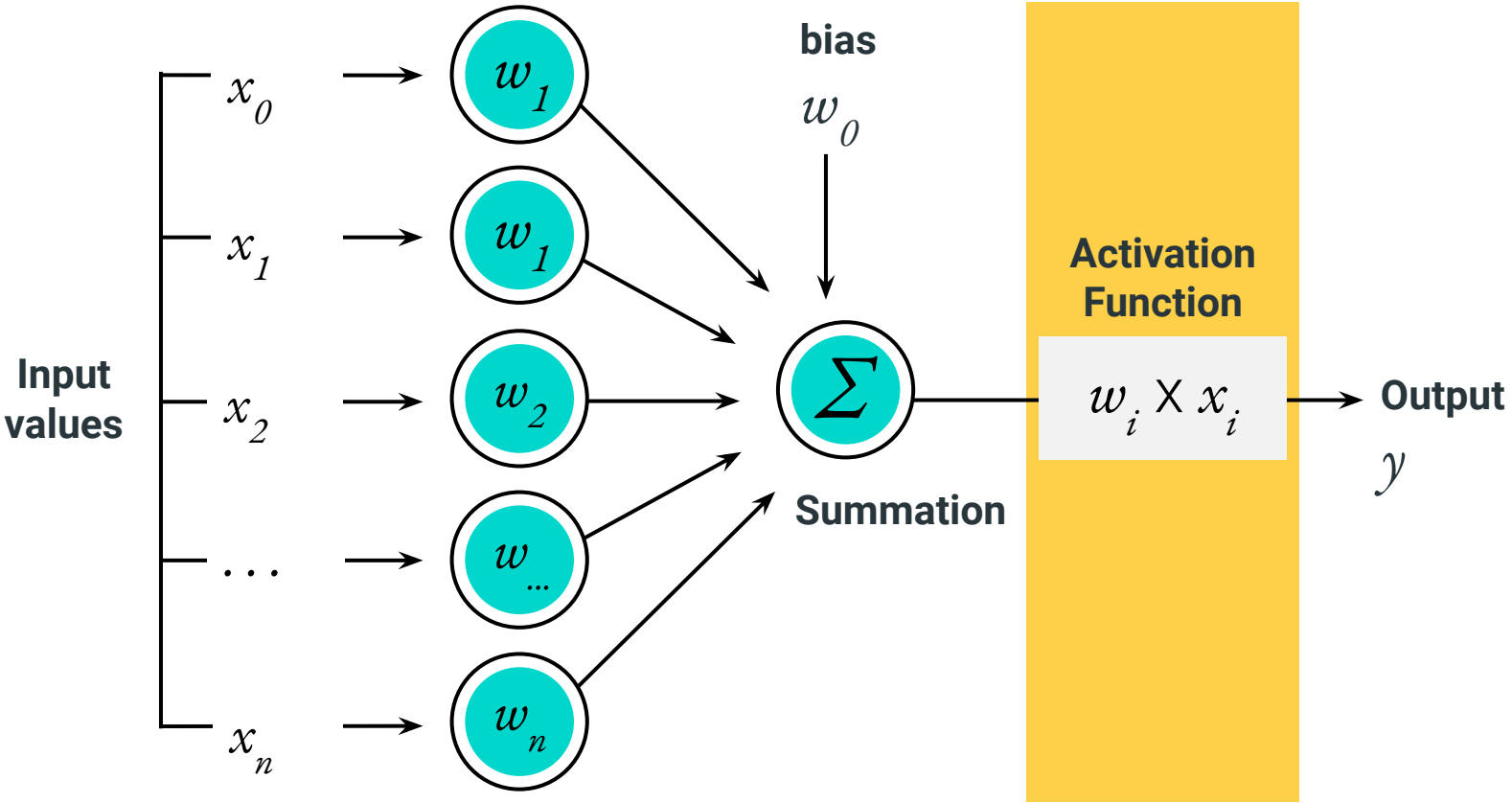
Activation Functions

Neural networks link neurons that process input together to produce a clear, quantitative output.

An **activation function** combines all these outputs into a single classifier or regression model.

- When building a model, we apply the activation function to the end of each neuron (each individual perceptron model).
- This mathematical function transforms each neuron's output into a quantitative value.
- The quantitative output value becomes the input value for the next layer in the neural network model.

Activation Function (Transformation After Summation)

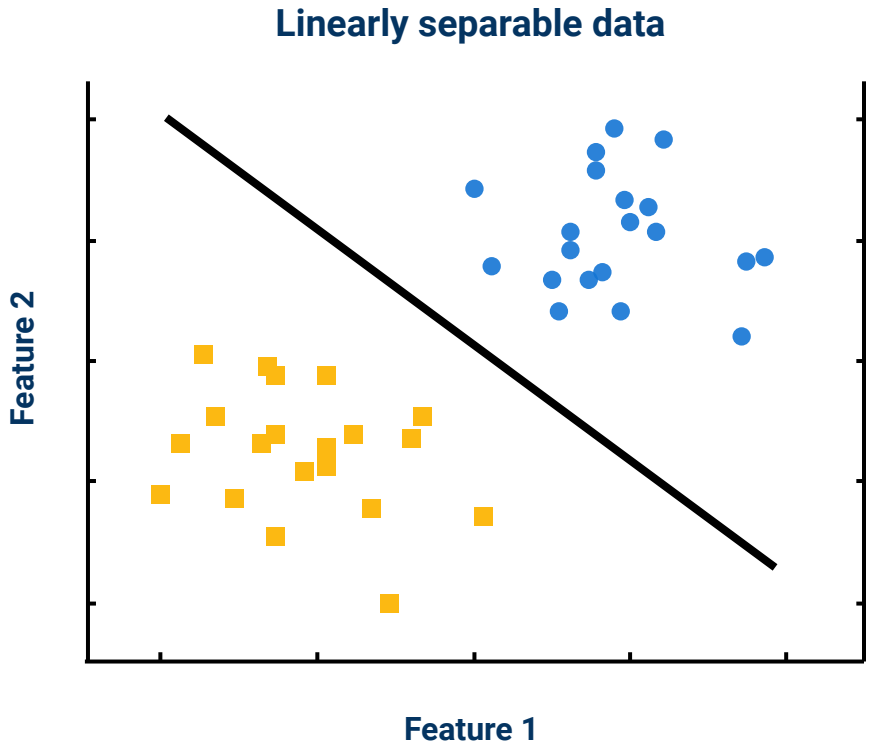


A wide variety of activation functions exist, and each has a specific purpose.

However, most neural networks will use one of the following activation functions.

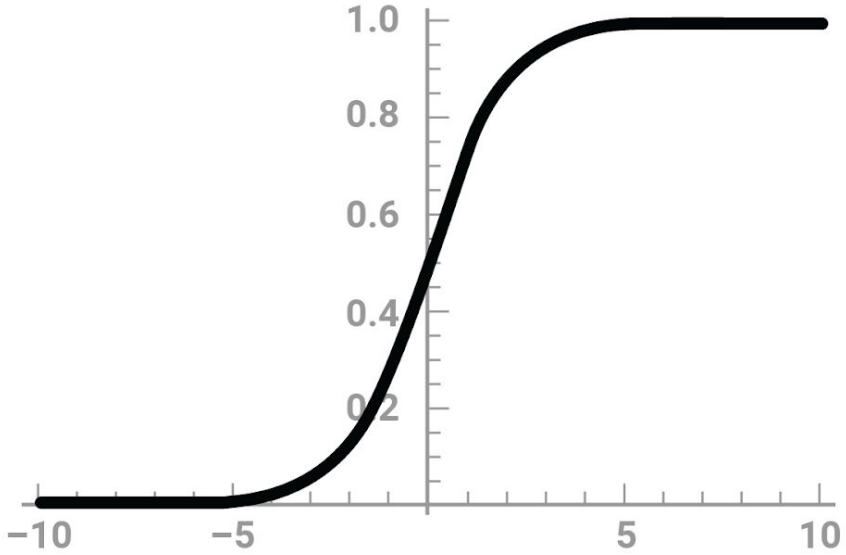
Linear Function

The **linear function** transforms the output into the coefficients of a linear model (the equation of a line).



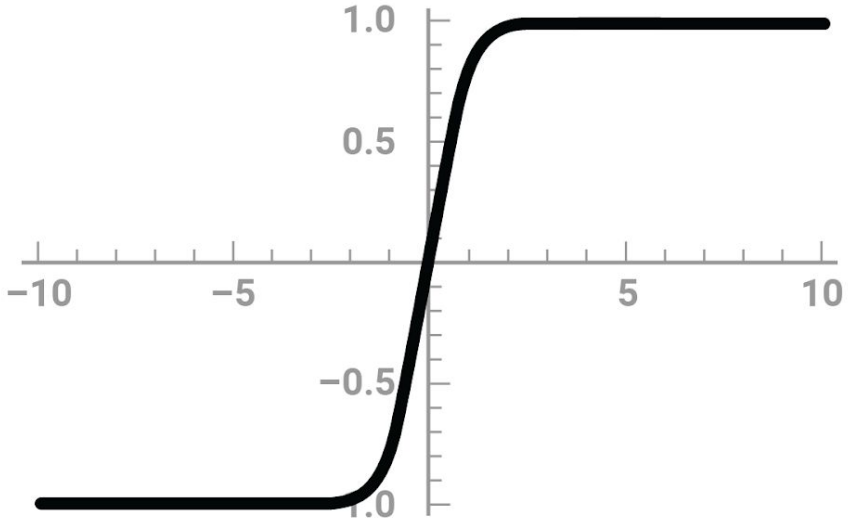
Sigmoid Function

The **sigmoid function** transforms the neuron's output to a range between 0 and 1, which is especially useful for predicting probabilities. A neural network that uses the sigmoid function will output a model with a characteristic S-curve.



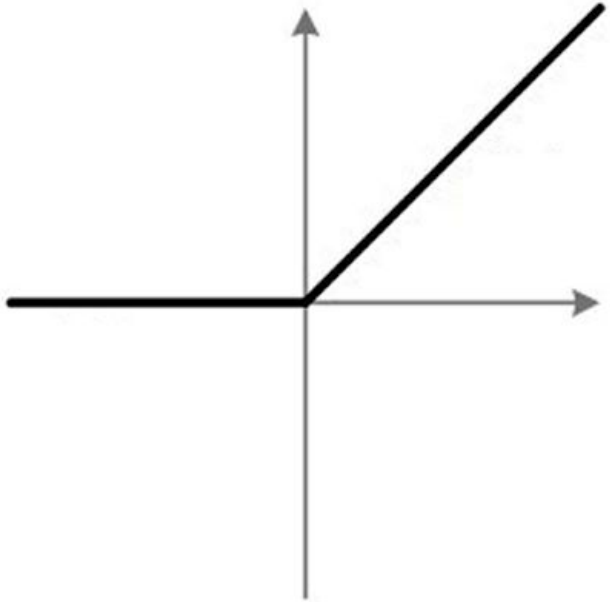
Tanh Function

The **tanh function** transforms the output to a range between −1 and 1. The output for a model using a tanh function also forms a characteristic S-curve. It's primary use is classifying data into one of two classes.



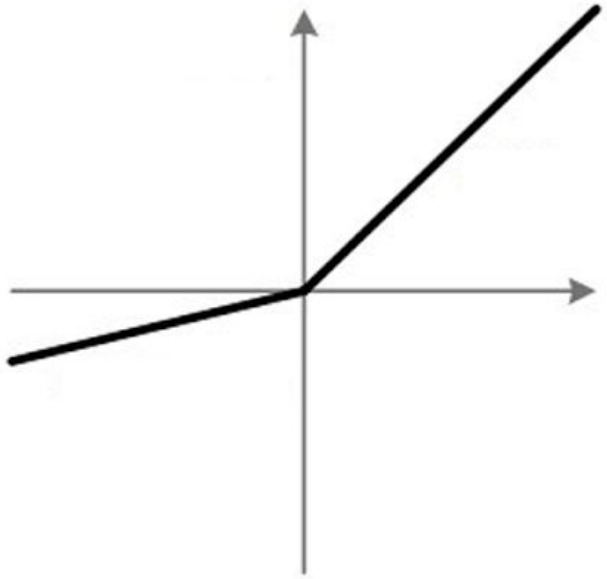
Rectified Linear Unit (ReLU)

The **rectified linear unit (ReLU)** function returns a value from 0 to infinity. This activation function transforms any negative input to 0. It is the most commonly used activation function in neural networks due to its faster learning and simplified output. However, it is not always appropriate for simpler models.



Leaky ReLU Function

The **leaky ReLU function** is an alternative to the ReLU function, which can sometimes work better. Instead of transforming negative input values to 0, it transforms them into much smaller negative values.



TensorFlow **Playground**

TensorFlow is a neural network and machine learning library for Python that has become an industry standard for developing robust neural network models.

Creating a Neural Network

Why TensorFlow?

1. Machine learning library
2. Used to build neural networks
3. Runs on multiple platforms
4. Supports distributed computing
5. Allows detailed fine-tuning

Modeling Workflow

Regardless of what machine learning model or technology we use, we follow the same general modeling workflow across all of machine learning:

01 Decide on a model and create a model instance.

02 Split the dataset into training and testing sets and preprocess the data.

03 Train/fit the training data to the model.

04 Evaluate the model for predictions and transformations.

Activity:

Playing in the TensorFlow Playground

In this activity, you will use TensorFlow Playground to explore all the different components of a neural network and their interactions.

Suggested Time:
15 Minutes

Time’s up!

Let’s review

Questions?

Instructor **Demonstration**

Understanding the TensorFlow Neural Network Structure

Break

15 mins

Activity:

Work Through a Neural Network Workflow

In this activity, you will use TensorFlow to explore all the different components of a neural network and their interactions.

Suggested Time:
20 Minutes

Time’s up!

Let’s review

Questions?

Activity:

BYONNM—Build Your Own Neural Network Model

In this activity, you will implement your own basic classification neural network model using the TensorFlow Keras module. In addition, you will create your own dummy data, split the data into training and test sets, and normalize the data using scikit-learn.

Suggested Time:
20 Minutes

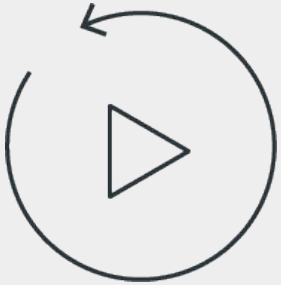
Time’s up!

Let’s review

Questions?

Instructor **Demonstration**

Neural Networks Aren’t Plug ‘n Play



Let’s **recap**

Recap

After today's lesson you are able to:

1. Compare the advantages and disadvantages of using neural network models with other types of machine learning models.
2. Describe the perceptron model and its components.
3. Create, train, and evaluate neural network models using TensorFlow.

Next

In the next lesson, you'll learn ...

The End