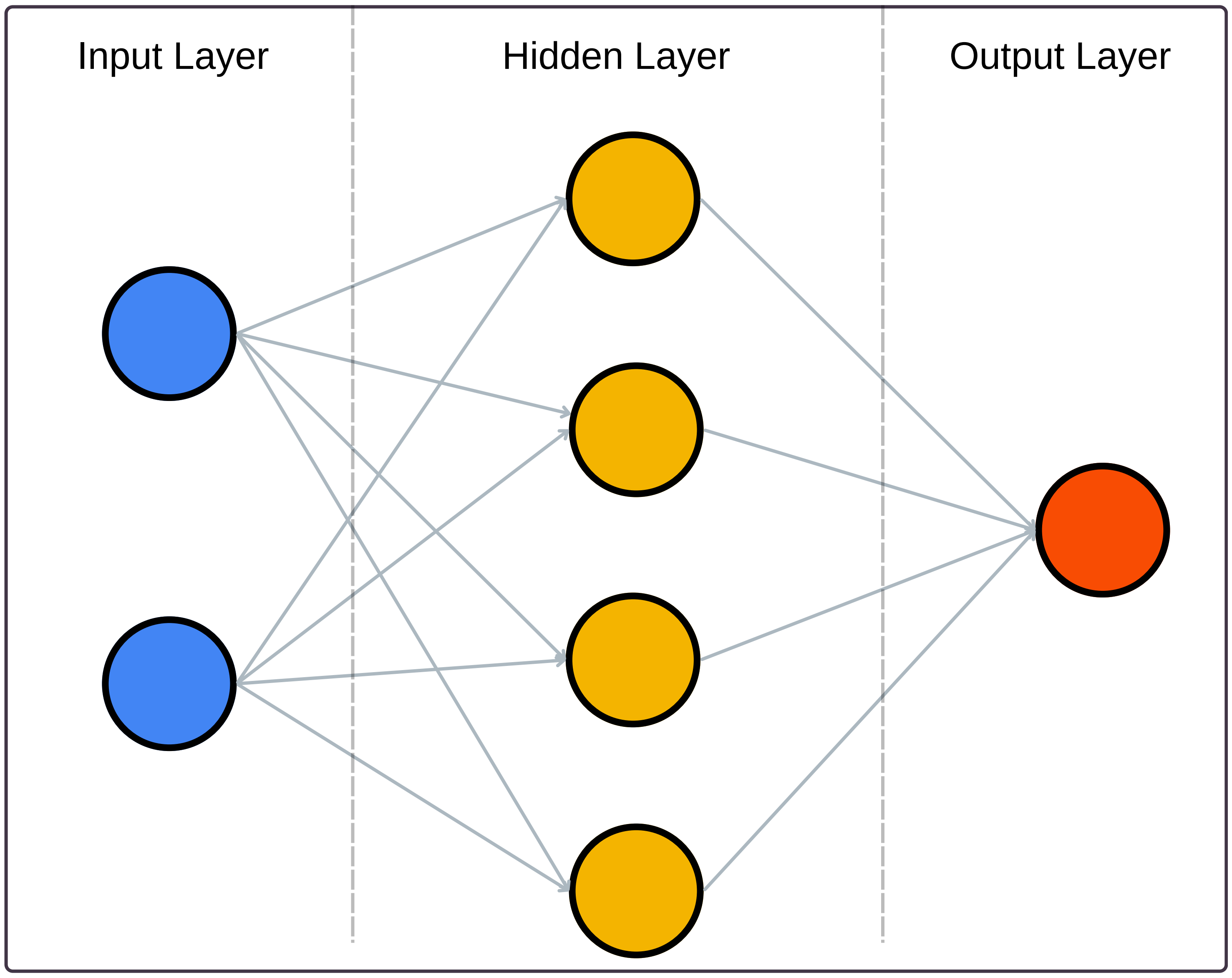
datasciencedojo
data science for everyone

Neural Networks

The Core of Deep Learning



Why Neural Networks?

Traditional Machine Learning struggles with **unstructured data** like images, text, and speech.

Neural Networks can learn directly from raw data, spotting patterns without manually programmed rules. That's why they power everything from image recognition to **chatbots like GPT!**

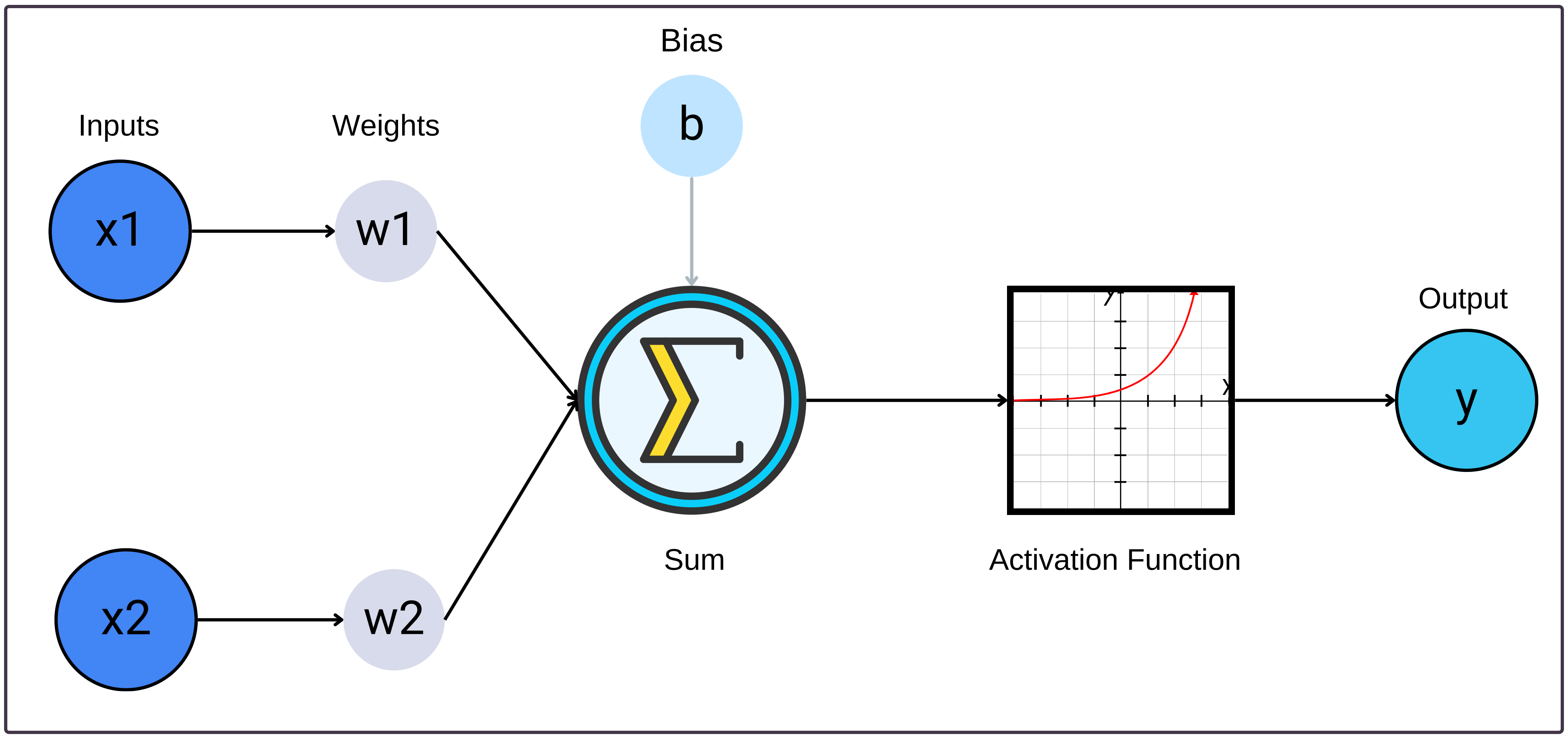
The Basic Structure of a Neural Network

A neural network is like a team of decision-makers, passing information through layers:

1. **Input Layer** – Receives raw data and sends it forward.
2. **Hidden Layers** – Where learning happens! Neurons use weights and biases to extract patterns.
3. **Output Layer** – Gives the final answer—classification, prediction, or probability.

Now, what are weights and bias?

- **Weights** decide how much influence an input has. Higher weight = more impact.
- **Bias** shifts the output up or down, helping the network adjust even when all inputs are zero.



How Neurons Process Information

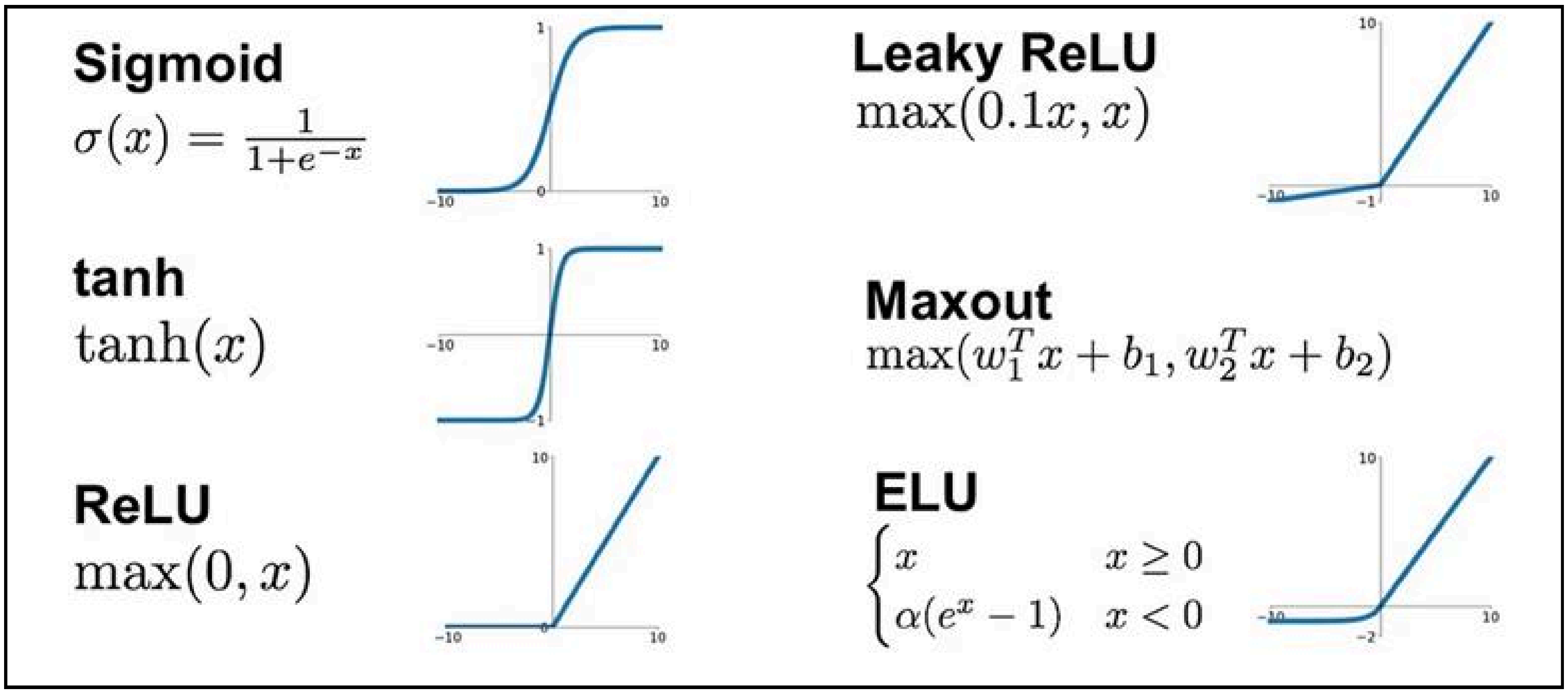
Each **neuron** takes inputs and performs a simple calculation:

1. Multiply inputs by **weights** (adjusting importance).
2. Add a **bias** (fine-tuning flexibility).
3. Apply an **activation function** (introducing non-linearity).

Mathematically:

$$y = f(WX + b)$$

This lets neurons recognize complex patterns instead of just linear relationships.



Activation Functions

Without activation functions, neural networks are just **stacked linear equations.** They add non-linearity, enabling networks to learn complex patterns like curves, edges, and language structures.

Think of it like this:
• Without activation functions, a neural network is like a **light switch**—either on or off (linear).
• With activation functions, it becomes a **dimmer**—capable of adjusting smoothly based on input.

Forward Propagation – How Predictions are Made

Here's how a neural network makes a prediction:
1. Input values pass through the network.
2. Each neuron computes a **weighted sum** and applies an activation function.
3. The output layer produces a result—a label, a number, or a probability.

Think of it as **data flowing forward,** transforming at each layer until a final decision is reached.

datasciencedojo
data science for everyone

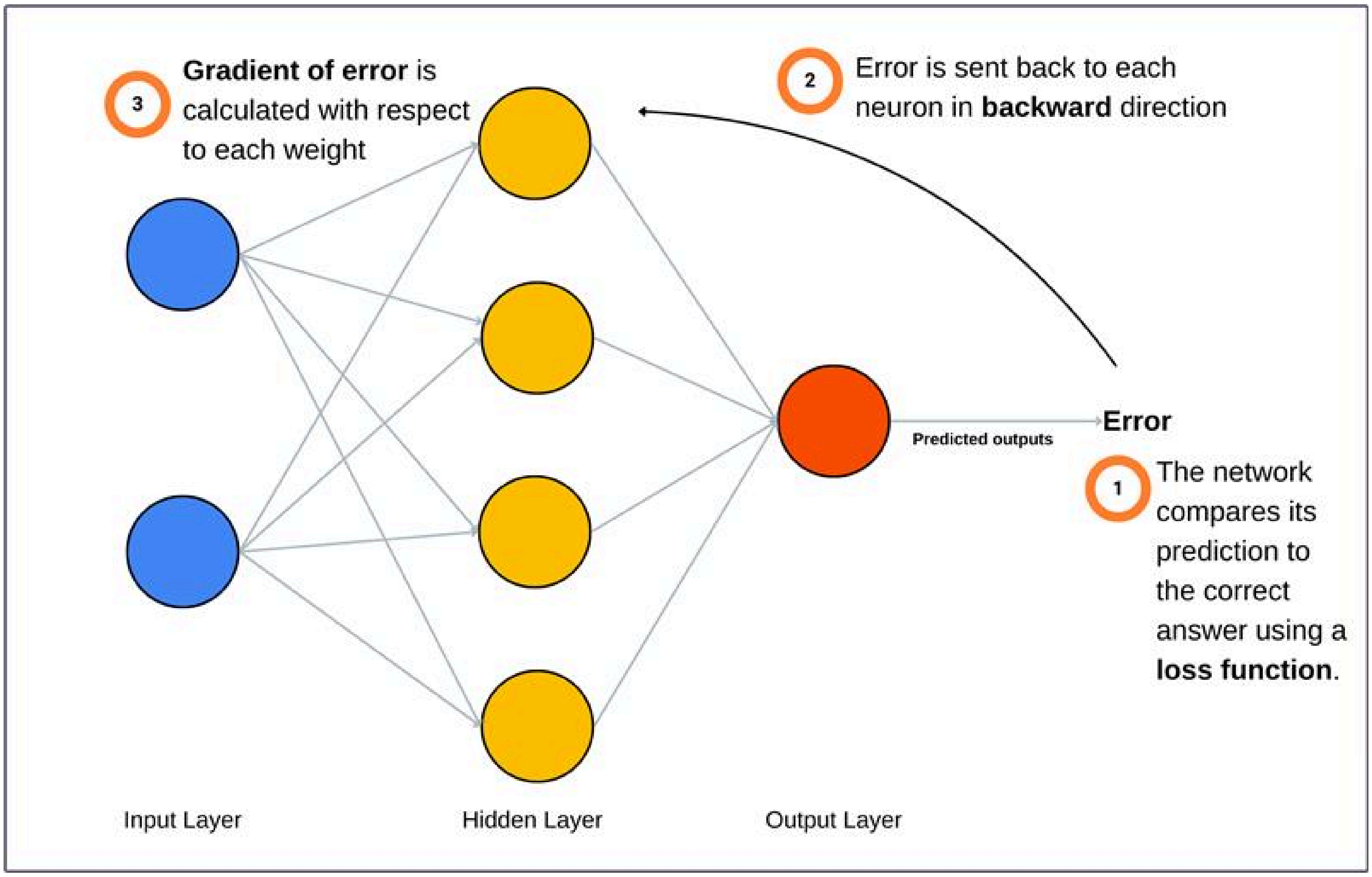
Loss Function – Measuring Errors

Once the network makes a prediction, we need to **check how wrong it is**. That's where a loss function comes in.

Common types:

- **MSE (Mean Squared Error):** For regression (predicting numbers).
- **Cross-Entropy Loss:** For classification (assigning labels).

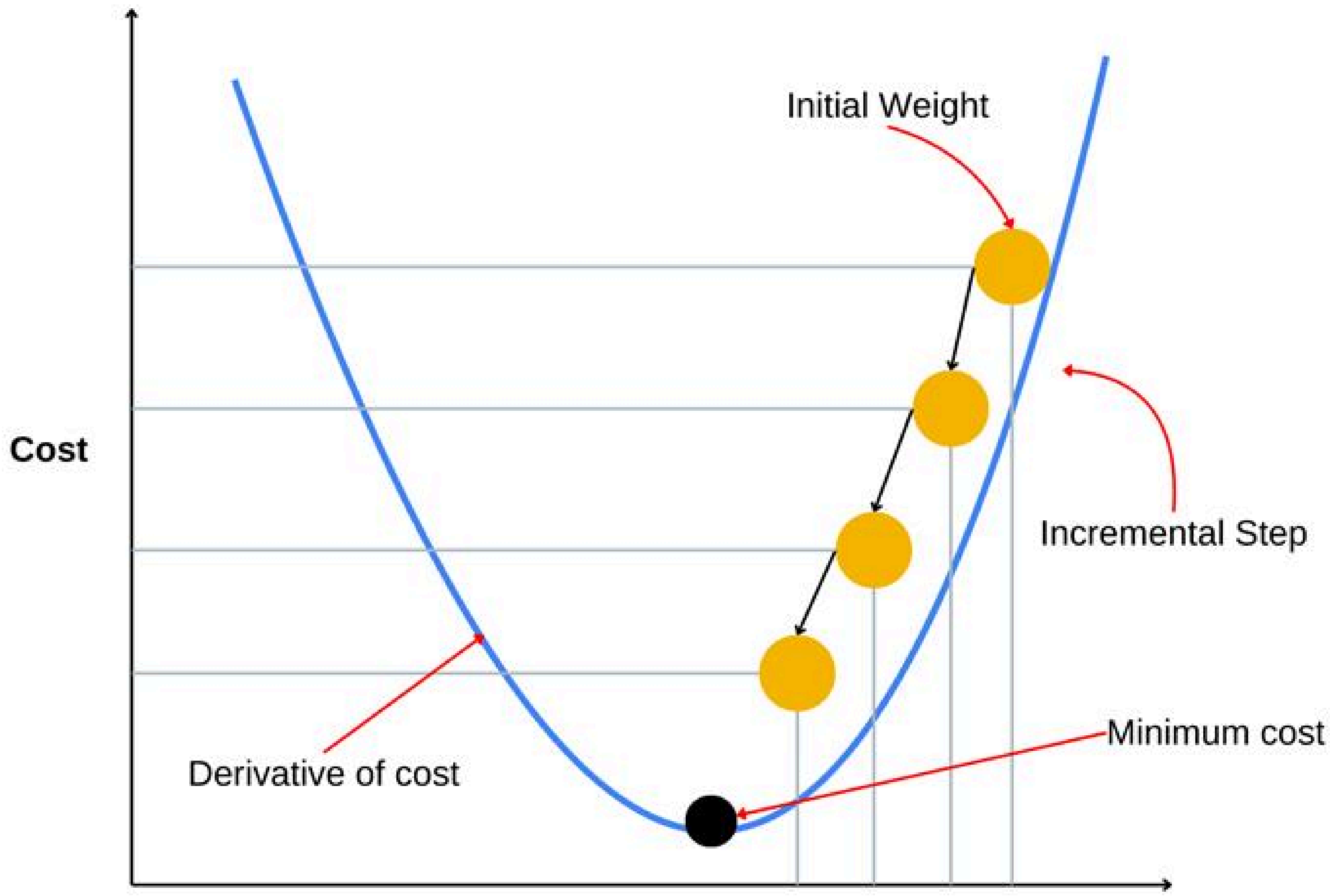
A lower loss means better predictions. The goal? Minimize the loss as much as possible!



Backpropagation – Learning from Mistakes

Neural networks don't get things right on the first try—they learn by making mistakes and correcting them. This learning process is called **backpropagation**.

This process repeats thousands or even millions of times, gradually improving the network's predictions.



What is Gradient Descent?

Imagine you're hiking down a mountain in thick fog. You can't see the bottom, so you take small steps downward, adjusting as you go. That's exactly how **gradient descent** works—**it gradually tweaks the network's weights to minimize errors.**

Process of Gradient Descent

Here's the process:

1. Calculate the **slope** (gradient) of the loss function.
2. Take a small step in the opposite direction to reduce error.
3. Repeat until the network reaches the **lowest possible error.**

This ensures the model learns efficiently

Why Deep Networks?

Shallow networks learn basic patterns, but deep networks capture complex relationships.

Example:

- Early layers in an image model detect **edges**.
- Middle layers recognize **shapes**.
- Deep layers identify **objects** (like faces or cars).

The deeper the network, the more abstract its understanding becomes.

datasciencedojo
data science for everyone

Overfitting & Regularization

Neural networks can memorize training data instead of learning patterns—this is overfitting.

To prevent it:

- **Dropout**: Randomly disables neurons during training.
- **L1/L2 Regularization**: Adds penalties to keep weights from growing too large.
- **More Data**: A bigger dataset helps the model generalize better.

A well-trained model performs well on new data, not just what it has seen before.

Final **Thoughts**

- Neural Networks process data **layer by layer** to make predictions.
- **Weights and biases** are adjusted to improve accuracy.
- Deep networks allow AI to understand **complex patterns.**

Understanding these fundamentals is the first step toward LLMs and Transformers!