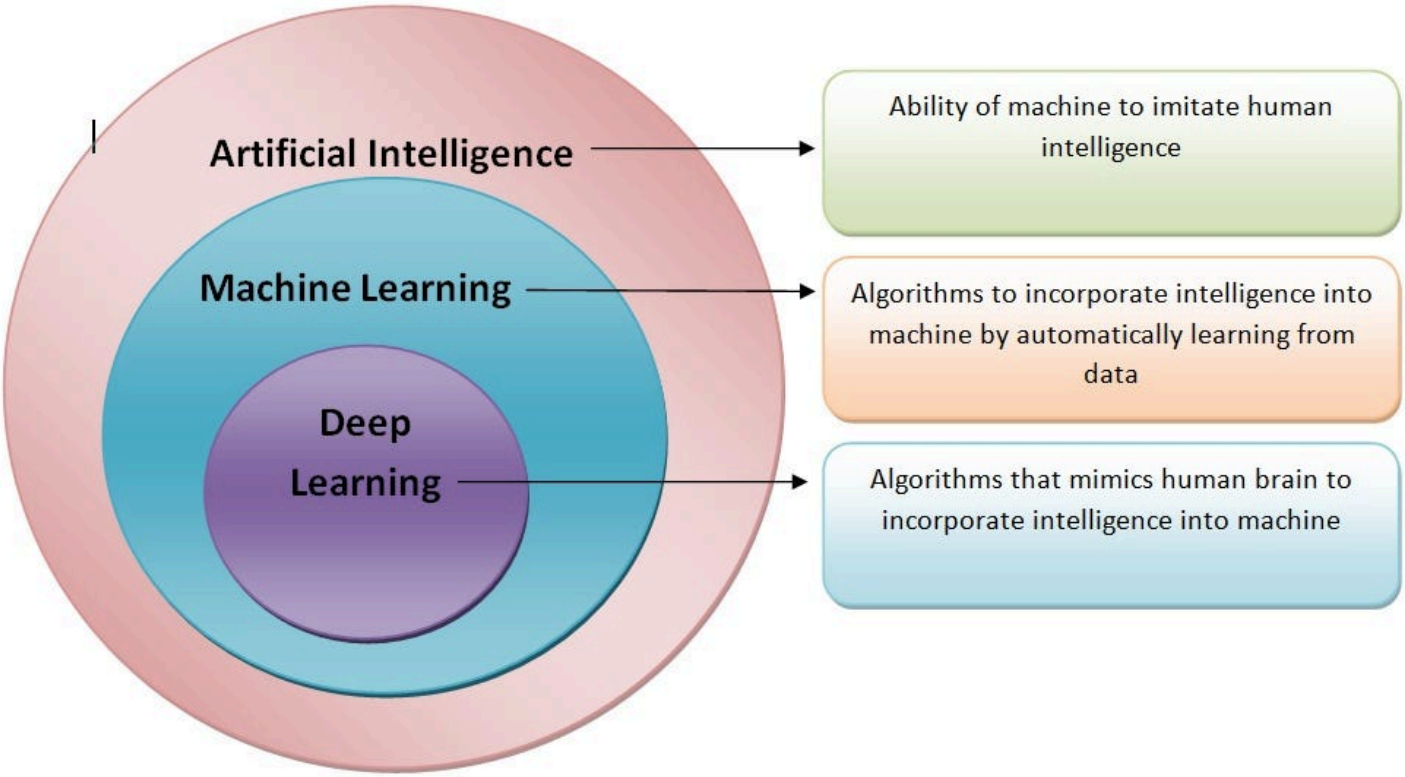
Introduction Deep Learning



Artificial Intelligence (AI):

AI is the overarching concept of creating machines that can think, reason, and learn like humans. It's about enabling computers to perform tasks that typically require human intelligence, such as problem-solving, understanding language, recognizing patterns, and making decisions.

Machine Learning (ML):

Machine Learning is a subset of AI. It focuses on developing algorithms that allow computers to learn from data without being explicitly programmed for every specific task. Instead of writing rules for every possible scenario, ML models learn to find patterns and make predictions based on the data they're trained on.

Example: Spam detection, Price Prediction

Deep Learning (DL):

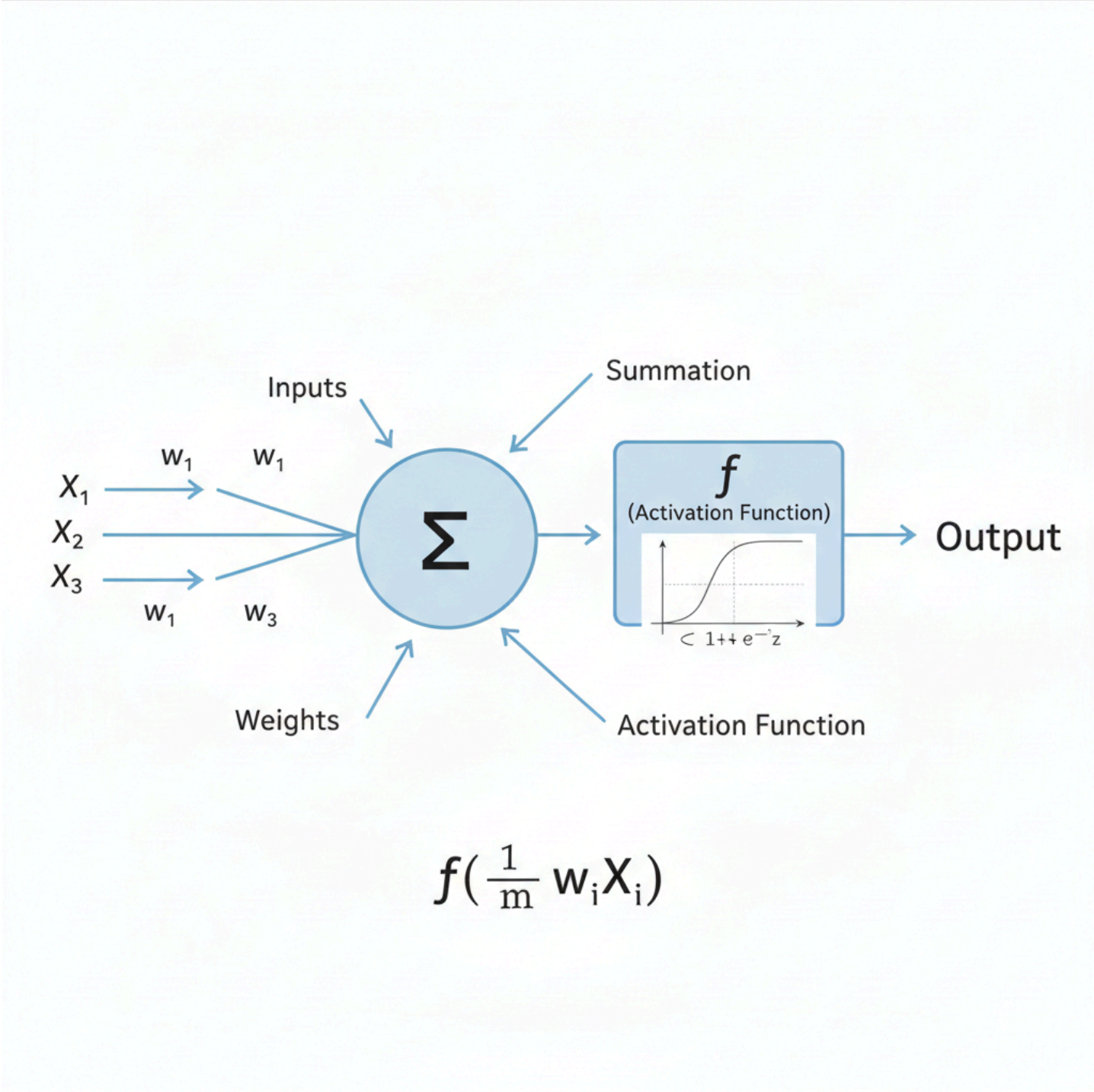
Deep Learning is a specialized subset of Machine Learning. What makes it "deep" is the use of Artificial Neural Networks (ANNs) with many layers, often called "deep neural networks." DL excels at learning from vast amounts of complex, unstructured data, such as images, audio, video, and text.

- **While traditional ML often requires human experts to design "features" (specific attributes) for the model to learn from, Deep Learning can automatically discover these features directly from the raw data.**
- **Deep Learning models continue to improve significantly as they are exposed to more and more data.**
- Deep Learning models are built upon Artificial Neural Networks (ANNs), which are inspired by the structure and function of the human brain.

Example : Image classification, Object detection

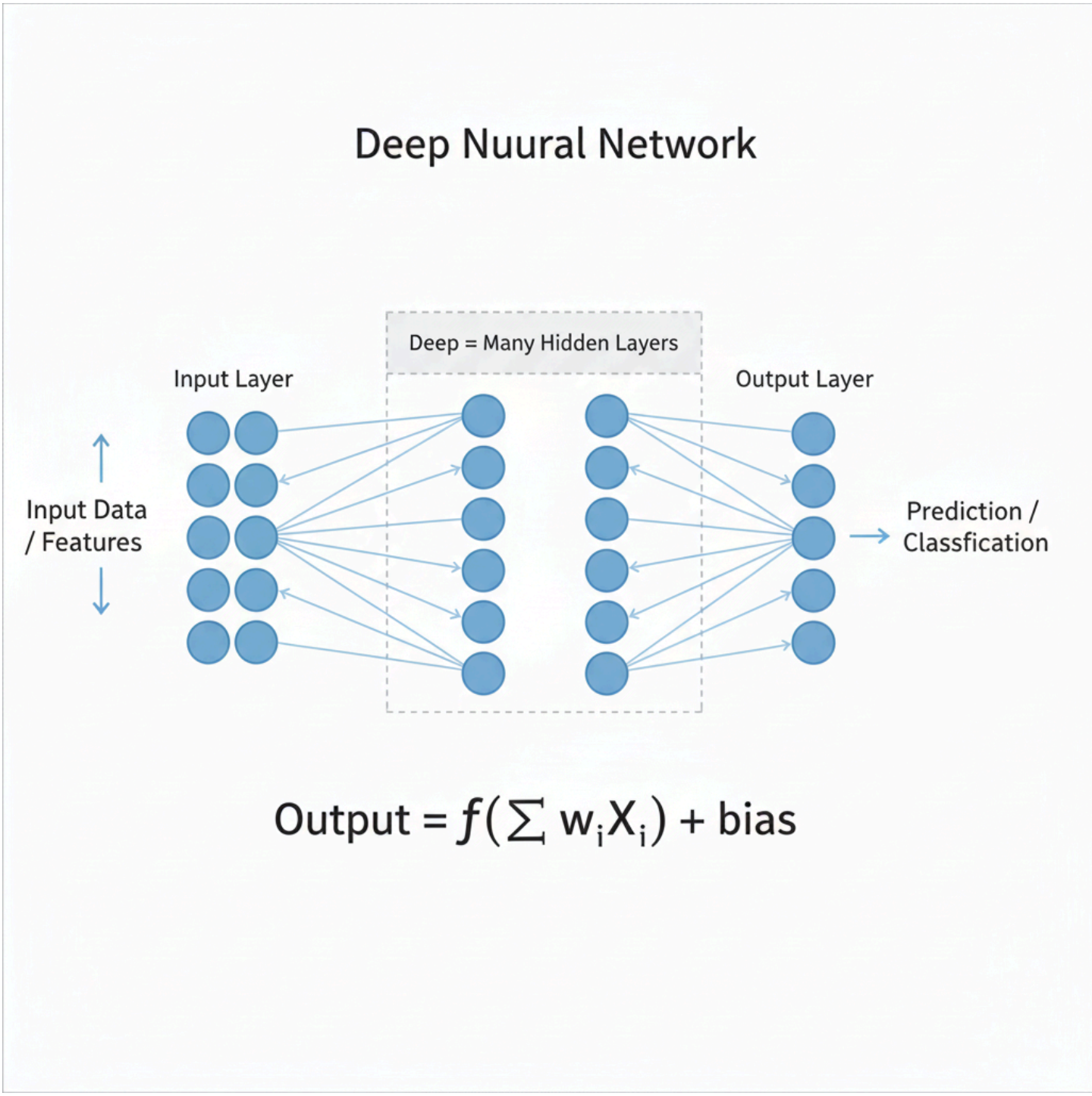
The Artificial Neuron (Perceptron):

Imagine a single brain cell (neuron). It receives signals, processes them, and then sends a signal onward. An artificial neuron, or perceptron, mimics this process



1. **Inputs (X i)**
   These are the pieces of information the neuron receives. Think of them as signals coming from other neurons or raw data.
2. **Weights (w i)**
   Each input is assigned a numerical weight. These weights represent the "strength" or "importance" of each input. A higher weight means that input has a greater influence on the neuron's output. During the learning process, the network adjusts these weights.
3. **Summation (∑):**
   The neuron calculates a weighted sum of its inputs. This means each input is multiplied by its corresponding weight, and all these products are added together. Often, a 'bias' term is also added to this sum, which allows the activation function to be shifted.
4. **Activation Function (f):**
   The sum is then passed through an activation function. This is a crucial non-linear transformation that introduces complexity into the network. Without activation functions, a neural network, no matter how deep, would behave like a single-layer model (only capable of linear transformations).
5. **Output:**
   The final value produced by the activation function is the neuron's output, which can then be fed as an input to other neurons or serve as the final prediction of the network.

$$Output = f(\sum_i w_i X_i + \text{bias})$$



Input Layer:

This is the first layer of the network. Each neuron in the input layer receives a single feature from the raw input data. For example, if you're training a network to classify images of handwritten digits, each input neuron might represent a pixel value.

Hidden Layers:

These are layers of neurons positioned between the input and output layers. In a "deep" neural network, there are typically multiple hidden layers. This is where the network performs complex computations, extracting increasingly abstract and useful features from the data. The depth allows the network to learn intricate patterns and representations.

Output Layer:

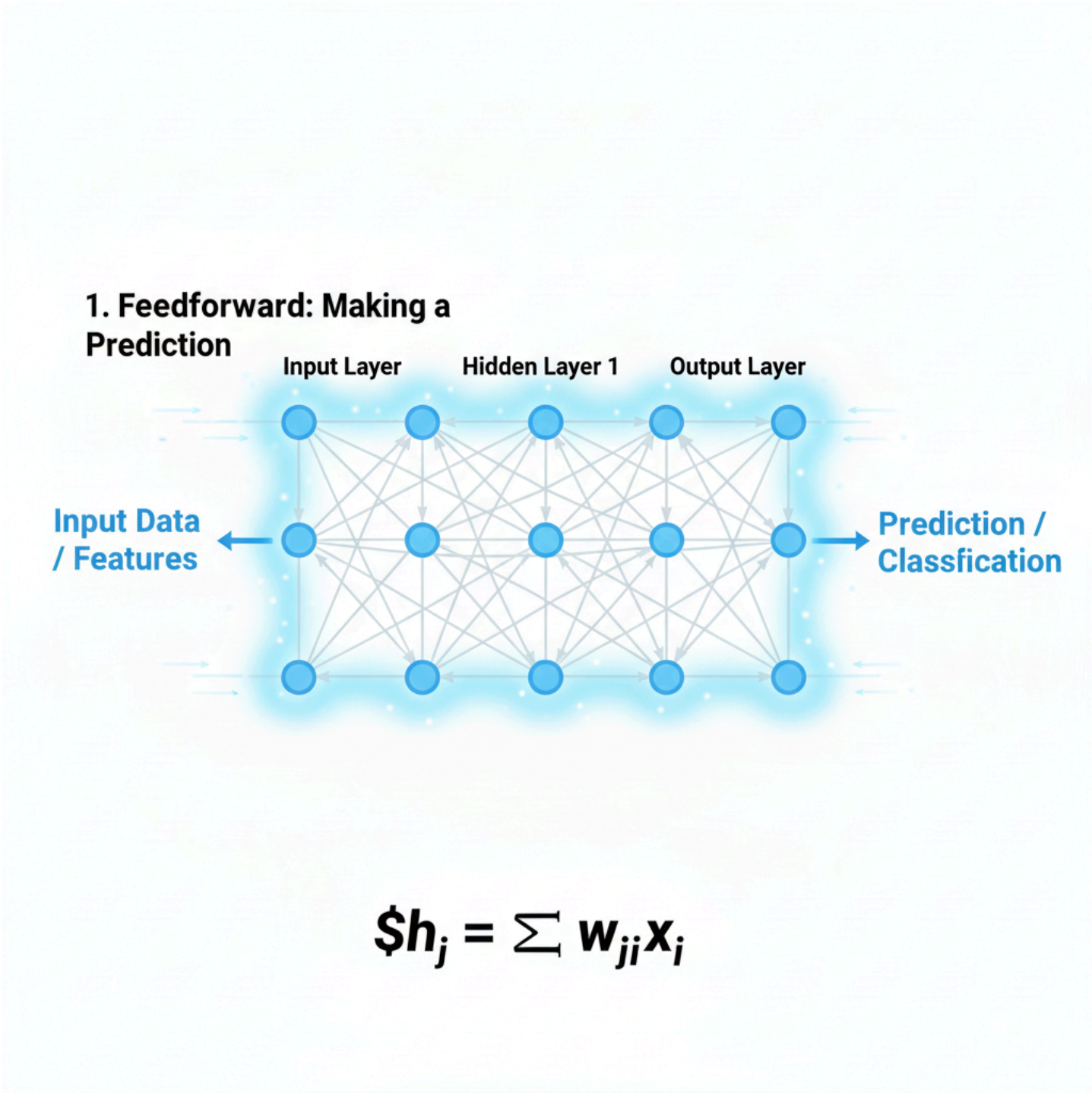
This is the final layer of the network. The number of neurons in the output layer depends on the task. For binary classification (e.g., "cat" or "dog"), you might have one output neuron. For multi-class classification (e.g., "cat," "dog," "bird," "fish"), you would have one neuron for each class. For regression tasks (predicting a continuous value), you might have a single output neuron.

How Neural Networks Learn

The "learning" in a neural network is the process of adjusting its weights and biases so that it can accurately map inputs to outputs. This learning typically involves two main phases: **Feedforward and Backpropagation** (combined with an optimizer like **Gradient Descent**).

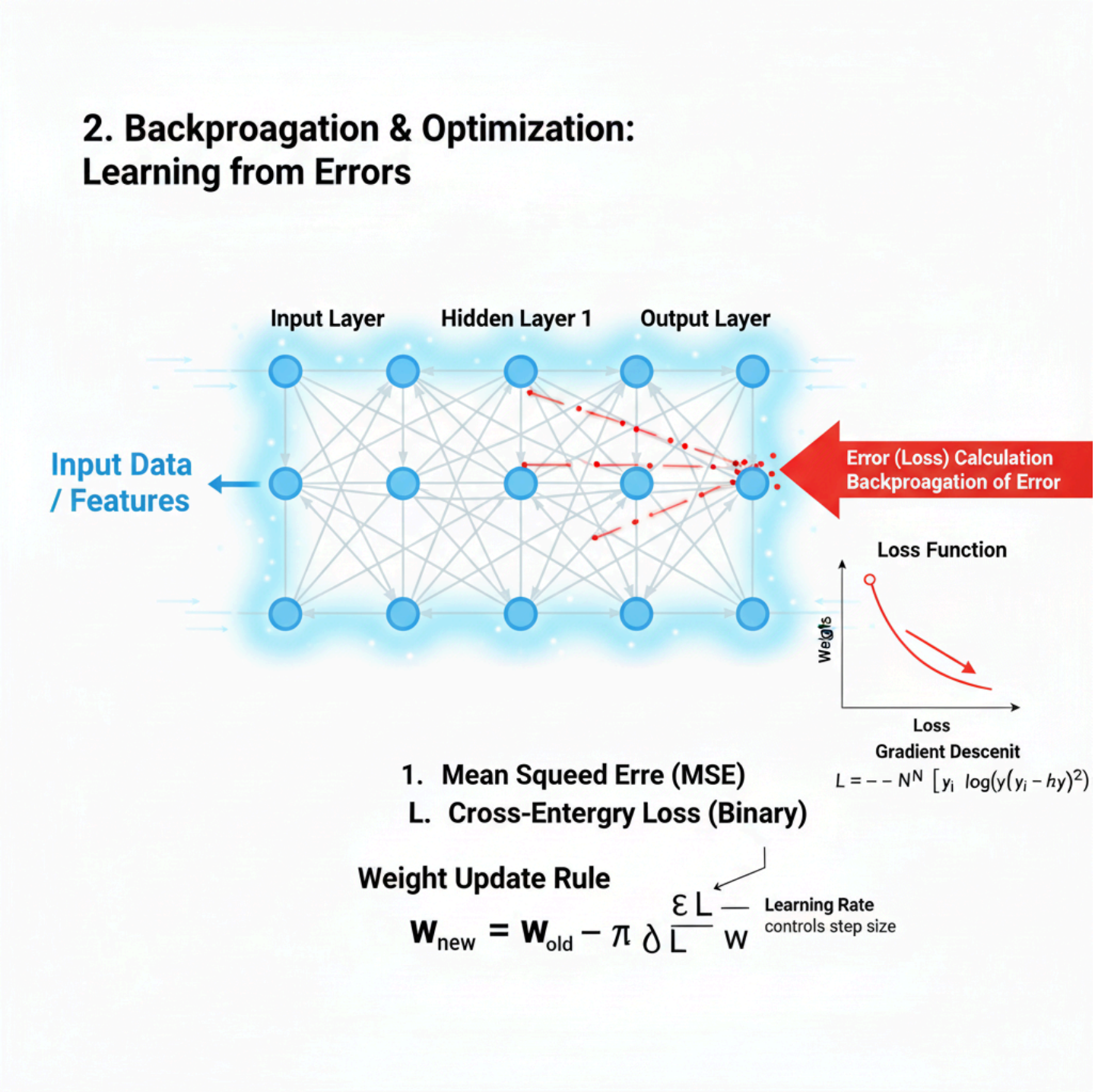
1. **Feedforward: Making a Prediction**

This is the process where data moves from the input layer through the hidden layers to the output layer to generate a prediction

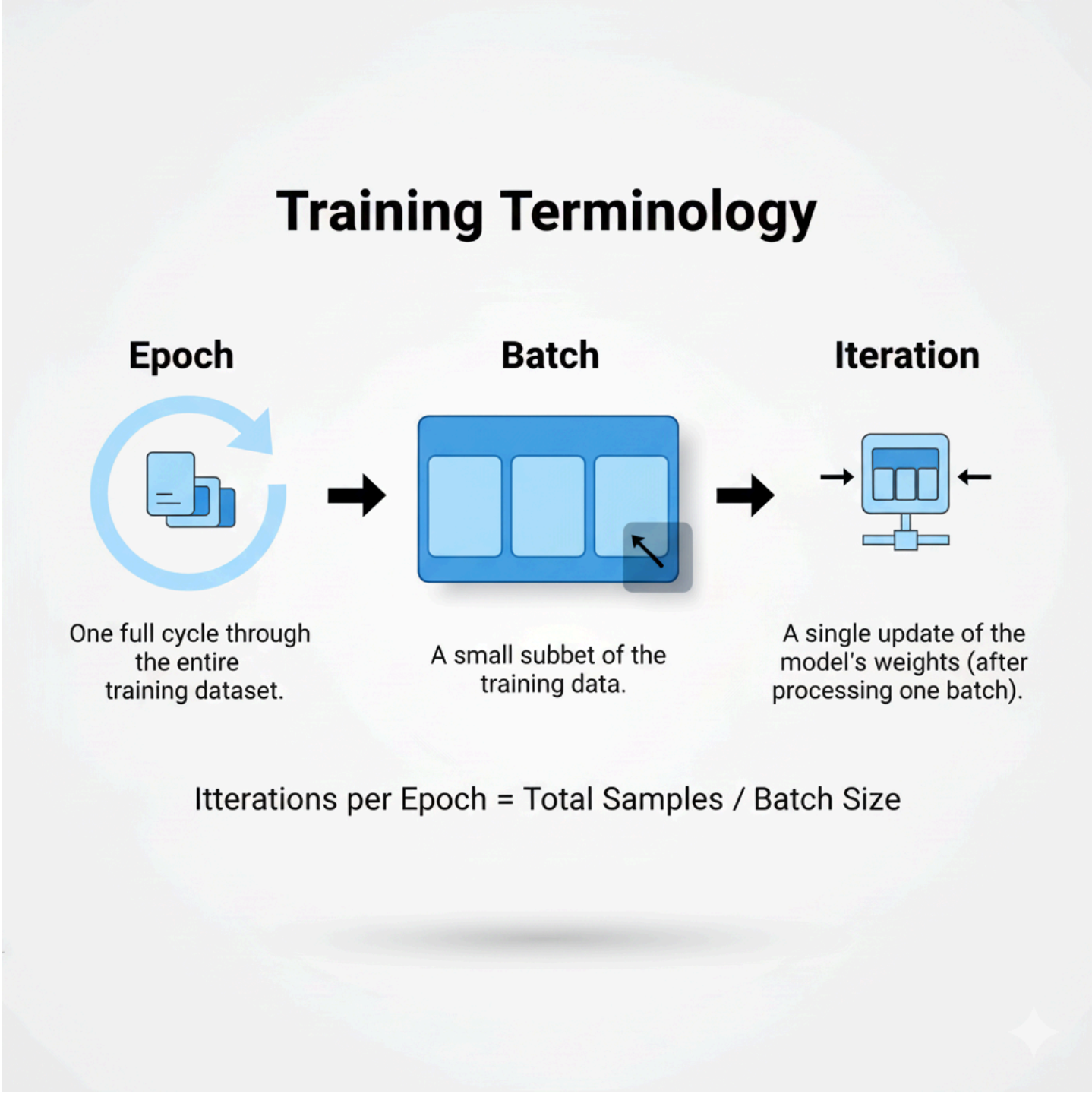


2. Backpropagation Learning from Errors

Once the loss is calculated, it's propagated backward through the network, from the output layer to the input layer. This process determines how much each weight in the network contributed to the overall error. It involves calculating the gradient of the loss function with respect to each weight.



Key Training Terminology:

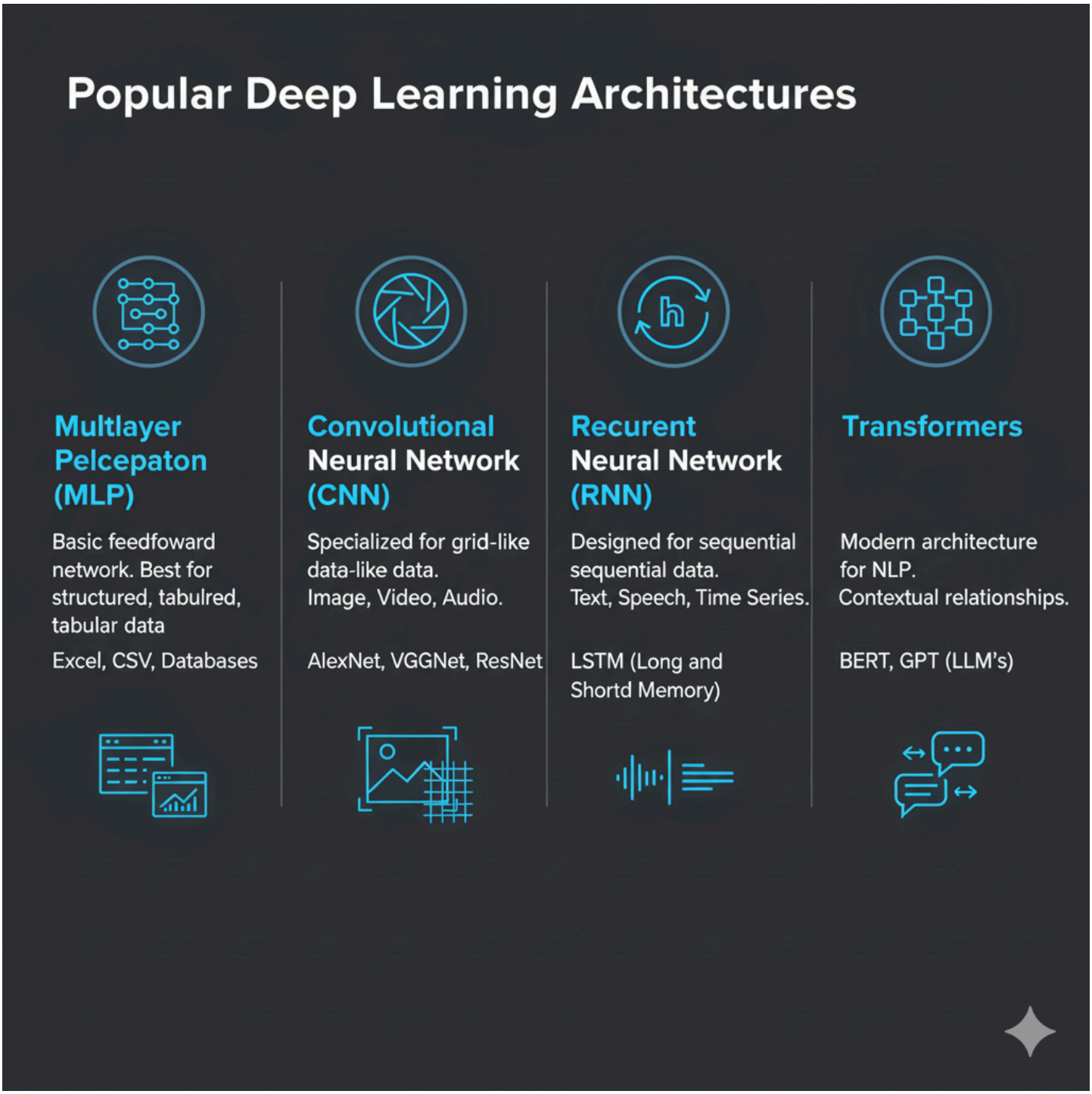


- **Epoch:**
  One complete pass through the entire training dataset. During one epoch, every single training example is seen by the model once.
- **Batch Size:**
  Instead of processing all training data at once (which can be computationally expensive for large datasets), the training data is usually divided into smaller subsets called batches. The model updates its weights after processing each batch.
- **Iteration:**
  An iteration refers to the processing of one batch of data. After each iteration, the model's weights are updated using backpropagation and gradient descent.
    - **Relationship:** The number of iterations per epoch is calculated as:

$$\text{Iterations per Epoch} = \frac{\text{Total Number of Training Samples}}{\text{Batch Size}}$$

- **Example: If you have 1000 training images and a batch size of 100, then one epoch will consist of 10 iterations (1000 / 100 = 10).**

Popular Deep Learning Architectures



1. Multilayer Perceptron (MLP):

- **Description:** This is the most basic type of deep feedforward neural network. It consists of an input layer, one or more hidden layers, and an output layer, with connections moving only in one direction (forward).
- **Best for:** Structured or tabular data (like data in Excel spreadsheets, CSV files, or databases). It's great for tasks like predicting house prices based on features like size, number of

bedrooms, and location.
- **Example:** Predicting customer churn based on historical transaction data and demographics.

2. Convolutional Neural Network (CNN):

- **Description:** CNNs are specifically designed to process grid-like data, making them incredibly effective for tasks involving spatial relationships. They use specialized layers called "convolutional layers" to automatically learn features like edges, textures, and patterns.
- **Best for:** Images, video, and sometimes audio.
- **Popular Examples:** AlexNet, VGGNet, ResNet (these are specific, highly successful CNN architectures).
- **Example:** Image classification (identifying objects in photos), facial recognition, medical image analysis (detecting tumors).

3. Recurrent Neural Network (RNN):

- **Description:** RNNs are designed to handle sequential data, where the order of information matters. Unlike MLPs or CNNs, RNNs have connections that loop back, allowing them to maintain an internal "memory" of previous inputs.
- **Best for:** Sequential data like text, speech, time series (e.g., stock prices).
- **Example:** Predicting the next word in a sentence, speech recognition, machine translation.

4. Long Short-Term Memory (LSTM):

- **Description:** LSTMs are a special kind of RNN that were developed to overcome a major limitation of standard RNNs: the inability to learn long-term dependencies (the "vanishing gradient problem"). LSTMs have a more complex internal structure with "gates" that control the flow of information, allowing them to selectively remember or forget past information over long sequences.
- **Best for:** Tasks requiring understanding context over long sequences, such as language modeling, sentiment analysis, and complex time series forecasting.
- **Example:** Understanding the full meaning of a long sentence, where important context might have appeared much earlier in the text.