Deep Learning

Narayan

Deep Learning

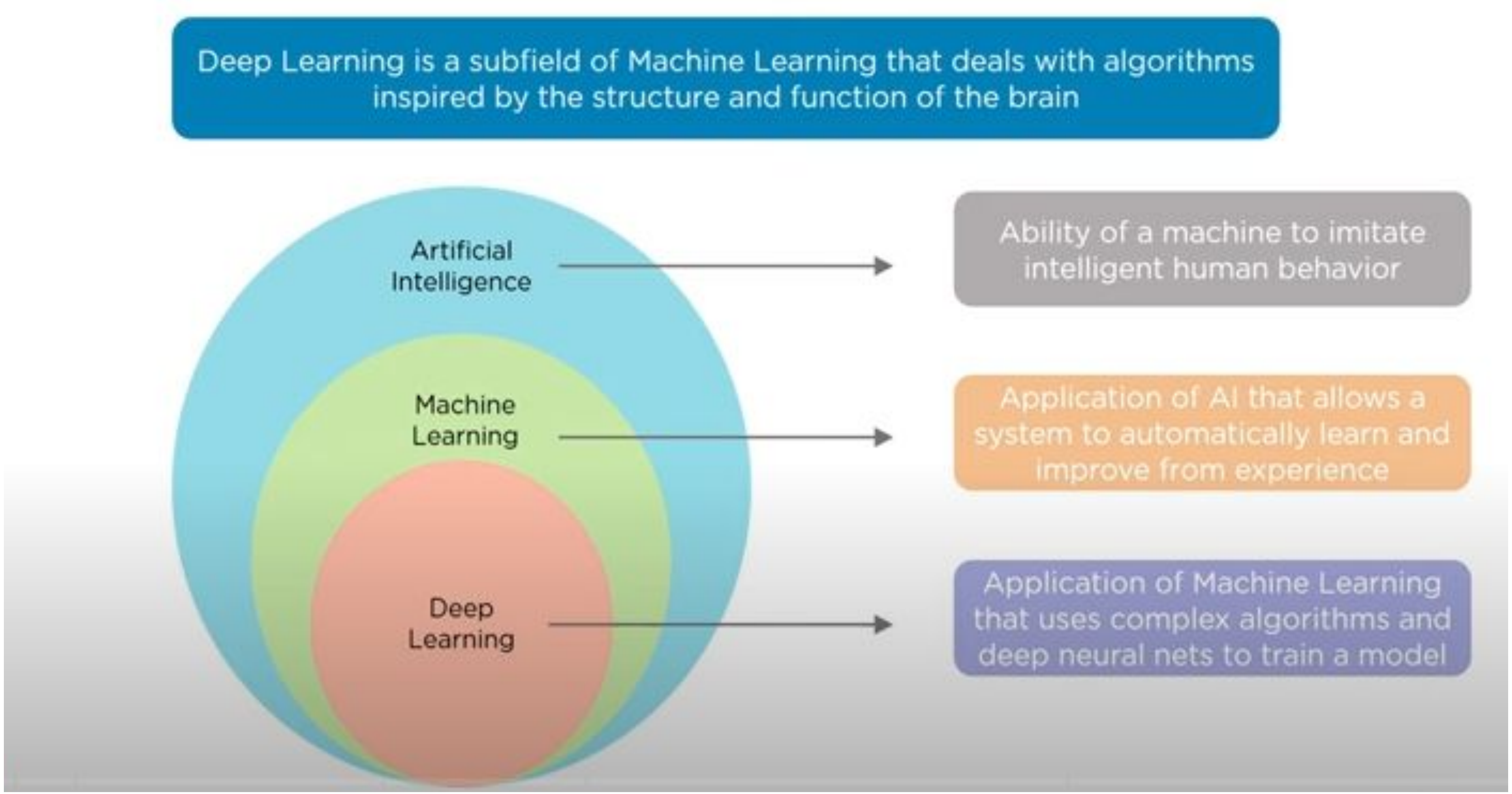
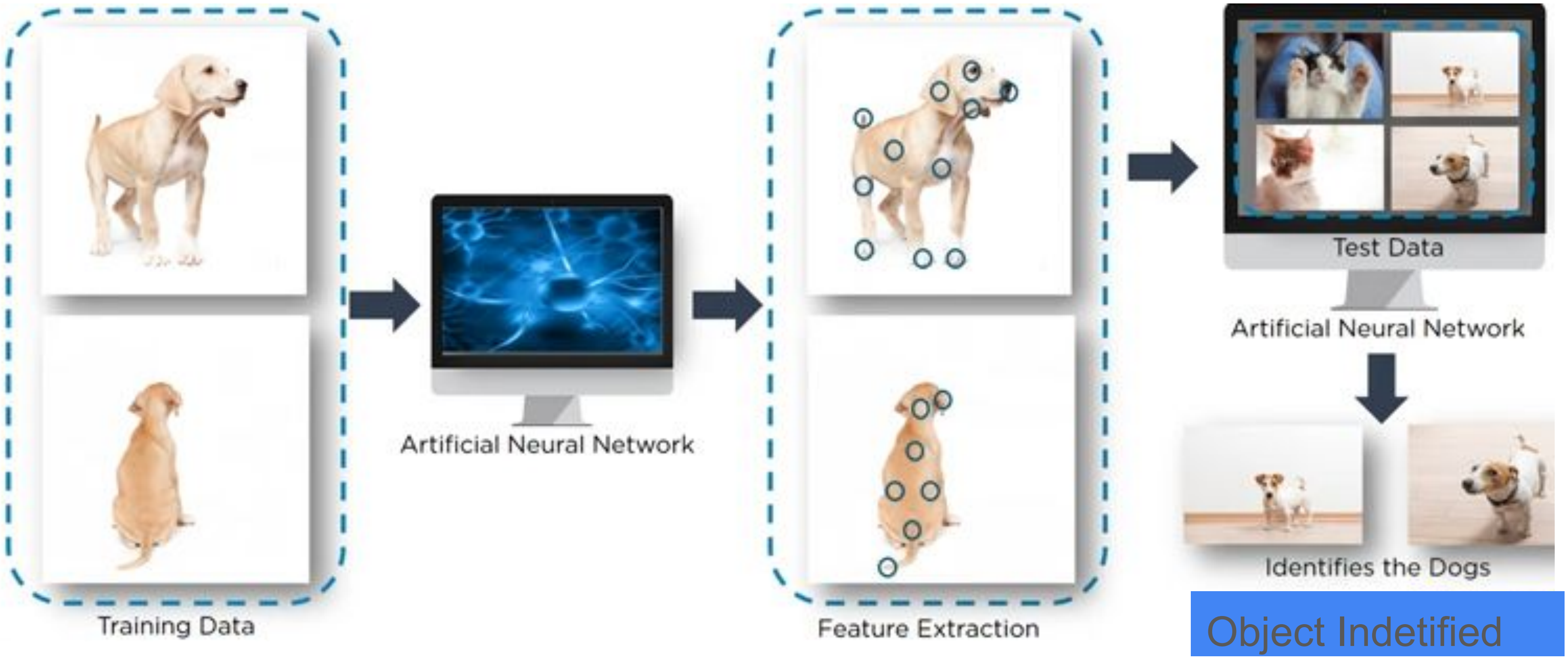


Image Recognition



Need of Deep Learning

Process huge amount of data

Machine Learning algorithms work with huge amount of structured data but Deep Learning algorithms can work with enormous amount of structured and unstructured data

Perform complex algorithms

Machine Learning algorithms cannot perform complex operations, to do that we need Deep Learning algorithms

To achieve the best performance with large amount of data

As the amount of data increases, the performance of Machine Learning algorithms decreases, to make sure the performance of a model is good, we need Deep Learning

Feature Extraction

Machine Learning algorithms extract patterns based on labelled sample data, while Deep Learning algorithms take large volumes of data as input, analyze the input to extract features out of an object and identifies similar objects

Applications

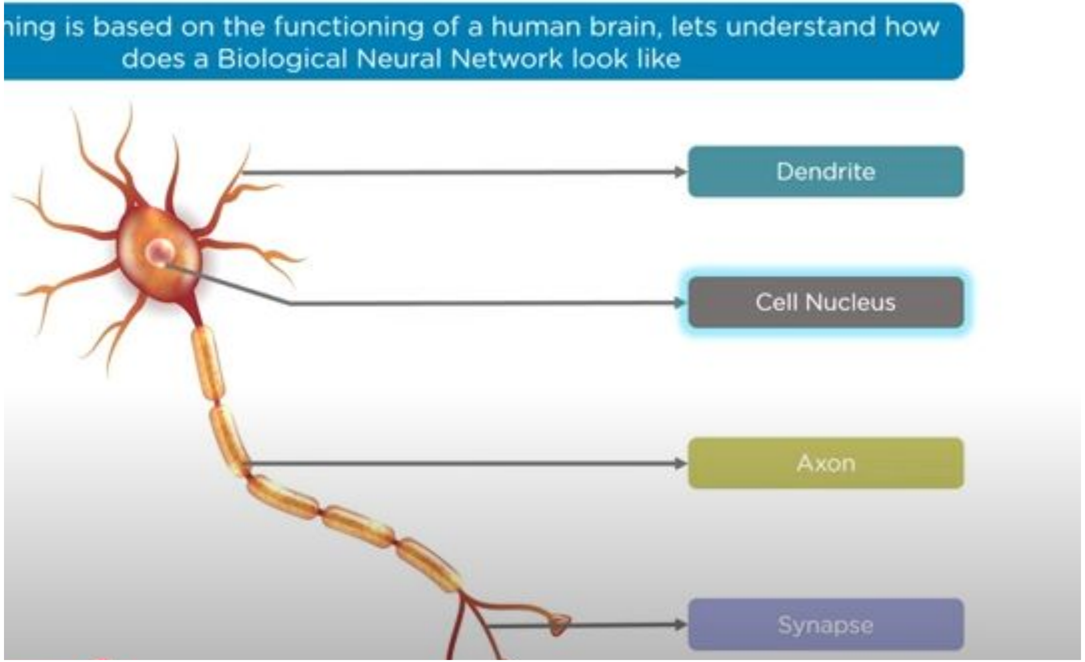
Robot Navigation

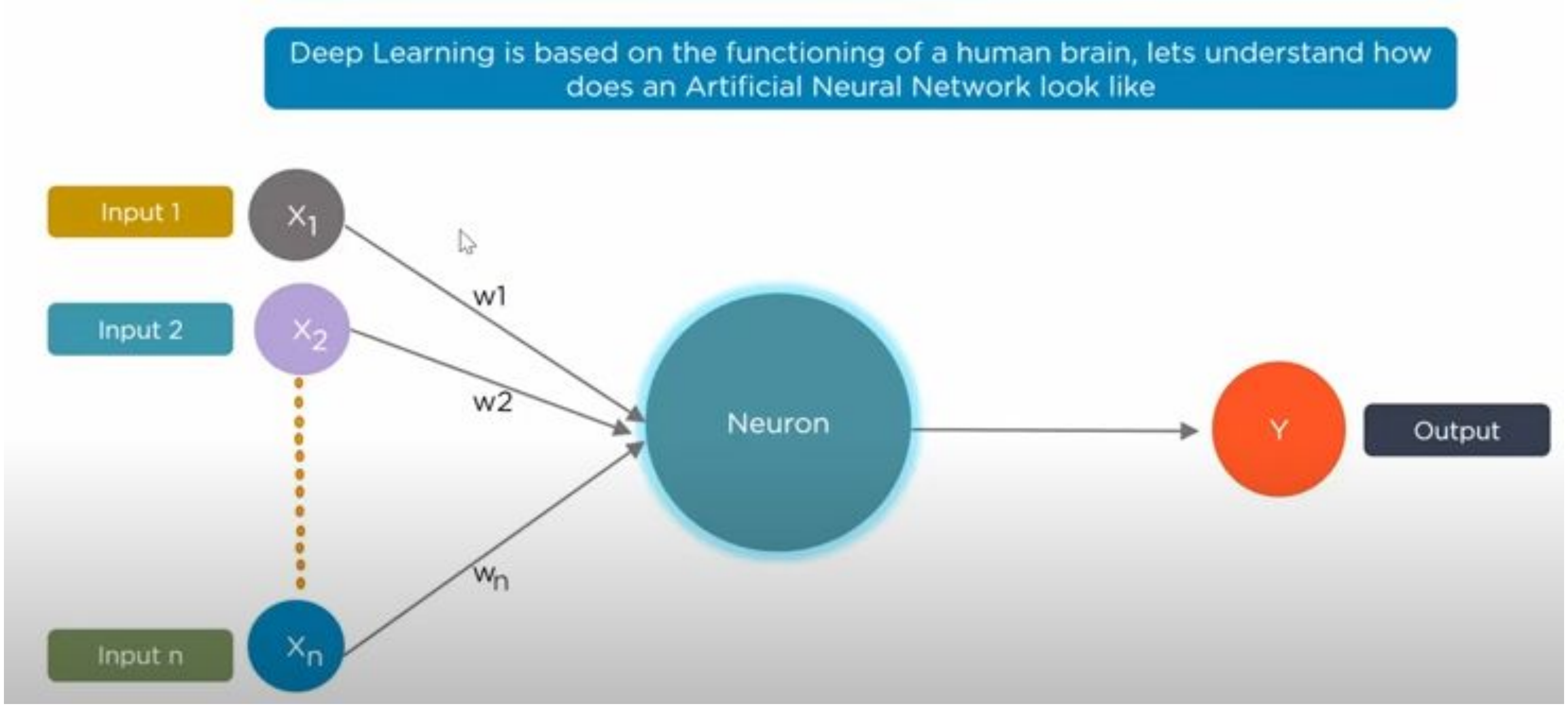

machine
translation
language
communication
internet
vocabulary
technology
software
automation
speech
display
Machine Translation

Music Composition

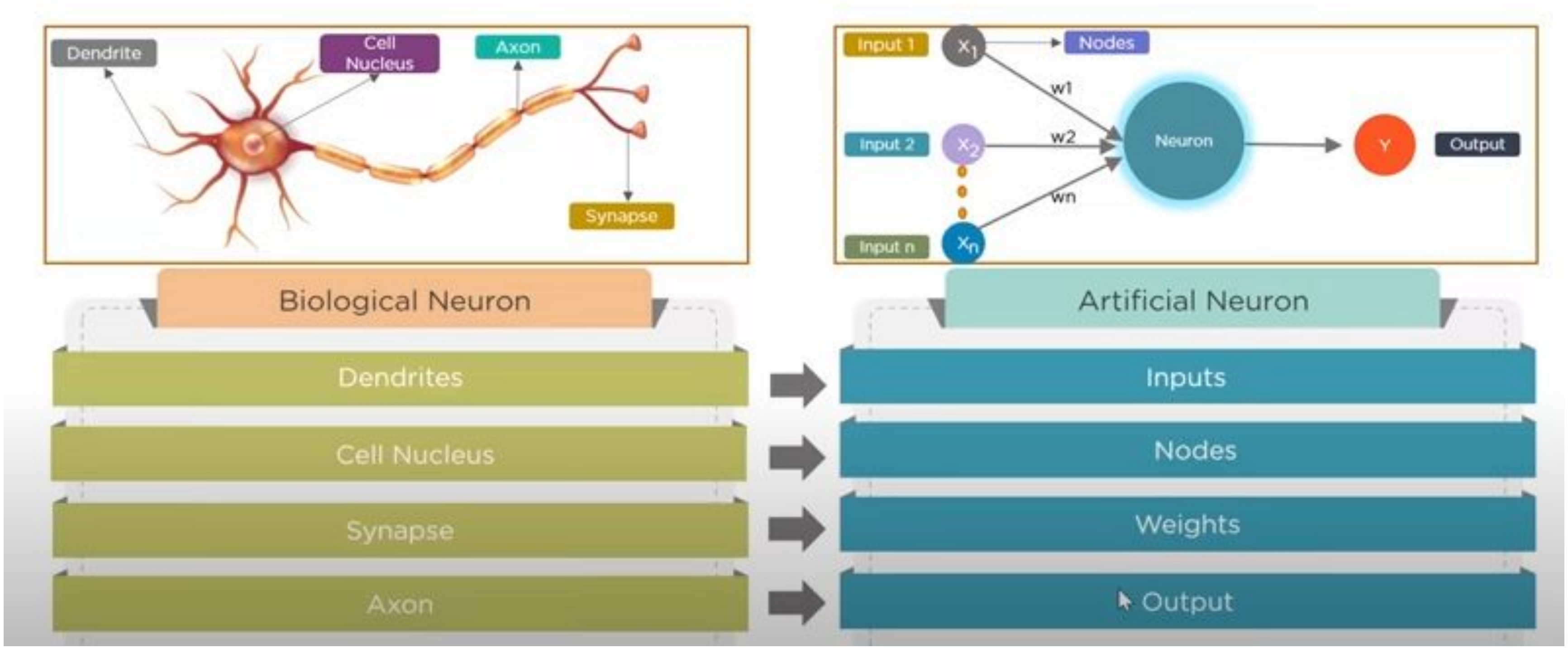
Colorization of images

Neural Network

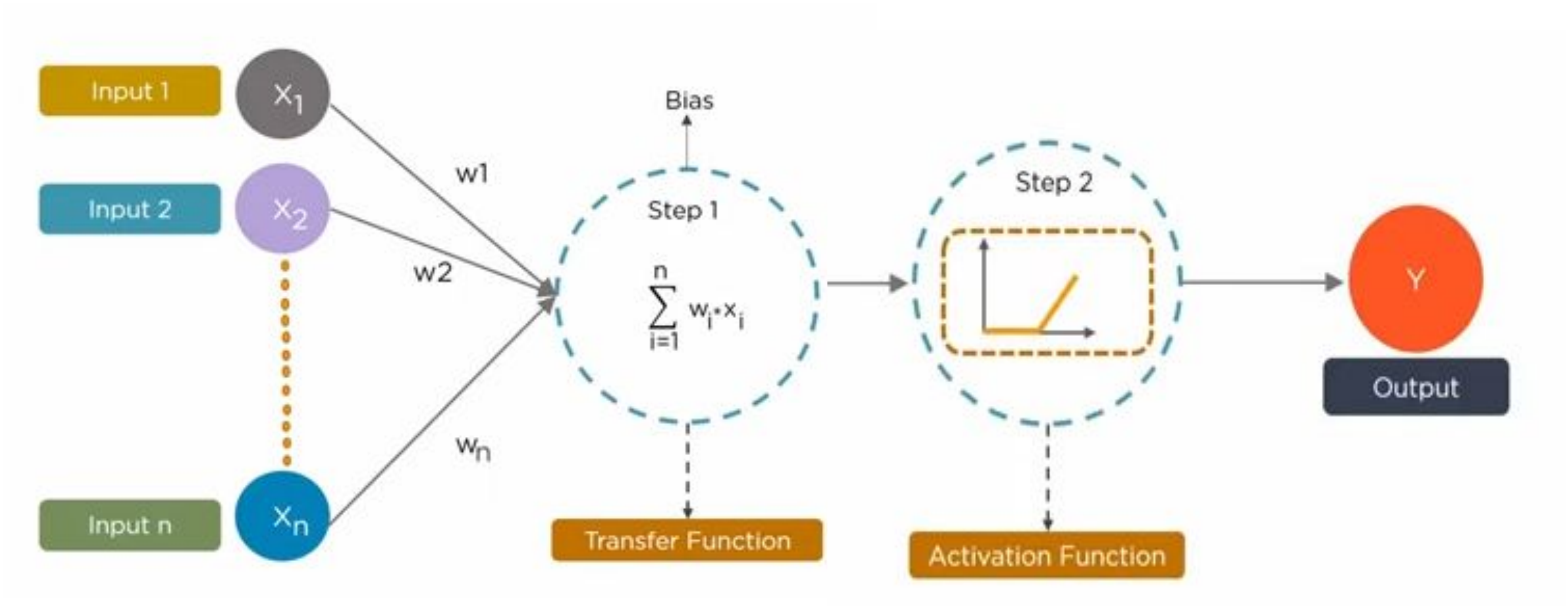



Deep Learning is based on the functioning of a human brain, lets understand how does an Artificial Neural Network look like
Input 1
Input 2
Input n
X_1
X_2
X_n
w1
w2
w_n
Neuron
Y
Output

Biological versus Artificial Neuron



Neural Network



Regularization

Regularization is a technique used to reduce **overfitting** in deep learning models by adding constraints to the learning process.

- In deep networks, large capacity → Ability to memorize training data (overfitting).
- Goal of regularization → Help the model **generalize well to unseen data** rather than just fitting training data perfectly.

Types of Regularization in Deep Learning

L1 and L2 Regularization (Weight Decay)

L2 Regularization (Most Common)

- Adds a penalty proportional to the square of the weights to the loss function.
-

Modified Loss function:

$$\mathcal{L}_{\text{new}} = \mathcal{L}_{\text{original}} + \lambda \sum_i w_i^2$$

where:

- λ: Regularization parameter (controls strength).
- w_i: Weight of the model.

Encourages smaller weights → Smoother models → Reduces overfitting.

L1 Regularization

Adds a penalty proportional to the **absolute value of the weights:**

$$\mathcal{L}_{\text{new}} = \mathcal{L}_{\text{original}} + \lambda \sum_i |w_i|$$

- Encourages sparsity → Many weights become zero → Useful for feature selection.

Dropout Regularization

At each training step:

- Randomly "drop" (turn off) a fraction of neurons in the network.

Example:

- Dropout rate = 0.5 → At each step, 50% of neurons are deactivated.

Benefits:

- Prevents neurons from co-adapting too much.
- Forces the network to learn redundant representations → Better generalization.

Important:

- During inference (testing), all neurons are active.
- The outputs are scaled accordingly to account for dropout.

Early Stopping

During training:

- Monitor validation loss.
- Stop training when validation loss stops improving → Avoid overfitting.

Example:

- Training continues up to 100 epochs, but validation loss stops improving after 20 → Stop at epoch 20.

Data Augmentation

👉 Instead of regularizing the model itself, we regularize by increasing data diversity.

✅ Common Techniques:

- Image augmentation: Rotation, flipping, scaling, adding noise, etc.

🌟 Effect:

- Makes the model more robust to input variations → Better generalization.

Batch Normalization (Kind of Implicit Regularization)

Normalize the inputs of each layer during training:

$$\hat{x} = \frac{x - \mu}{\sigma}$$

Benefits:

- Reduces internal covariate shift.
- Acts like regularization by adding small noise during training → Stabilizes learning.

Summary

- Regularization is crucial to prevent overfitting in deep networks.
- Main methods:
  - L1 / L2 Regularization (weight penalty).
  - Dropout (randomly disabling neurons).
  - Early Stopping (monitor validation).
  - Data Augmentation (diversify data).
  - Batch Normalization (normalizes layer inputs).

👉 These methods help improve model generalization, making it perform well on unseen data.

The goal of machine learning is not to memorize the data, but to generalize from it.