Get IT Right from RIG

Since 2011

Our outcomes are over 5000 trainees.

Artificial Intelligence Engineering
(Level-1)

Learn, Create, and Shine

Digital Space

Level-1

- Module 1: Introduction to AI and Machine Learning
- Module 2: Linear Algebra, Statistics and Probability for AI
- Module 3: Neural Network Architecture
- Module 4: Building Machine Learning Models
- Module 5: Deep Learning Concepts
- Module 6: Python Data Structure
- Module 7: Data Handling with Pandas and NumPy
- Module 8: Python for AI
- Module 9: Classification AI Project
- Module 10: Prediction AI Project

Artificial Intelligence Engineering
(Level-1)

Module 5: Deep Learning Concepts

Content

- Deep Learning Concepts
- Why Deep Learning needs?
- Deep Learning Architecture
- How Convolutional Neural Network Works?

Learning Outcomes

- **Deep Learning Concepts**: Understand the fundamental principles and key characteristics of deep learning.
- **Why Deep Learning is Needed**: Analyze the importance of deep learning in solving complex problems and its advantages over traditional machine learning methods.
- **Deep Learning Architecture**: Identify and explain the components and structure of deep learning models, including layers, activations, and backpropagation.
- **How Convolutional Neural Network Works**: Demonstrate knowledge of CNN operations, including convolution, pooling, and how they are used for image recognition tasks.

Deep Learning Concept

Definition: A subset of Machine Learning focused on neural networks with many layers (deep neural networks).

Real-Life Examples:

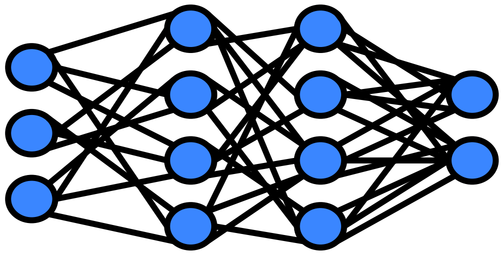
- Image Recognition (Face Detection)
- Natural Language Processing (Chatbots)
- Self-Driving Cars

Why Deep Learning needs?

- ✓ Solves complex problems with high accuracy.
- ✓ Requires less feature engineering.
- ✓ Can process unstructured data (images, audio, text).

Input → **Less feature engineering** → **Output**

Image



Hidden neurons

- **Car**

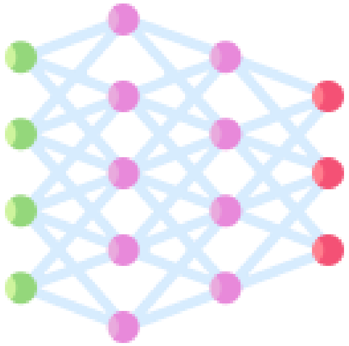
Color: Red
Made: LaFerrari
Model: Ferrari

To Read: https://addepto.com/blog/deep-learning-architecture/

Deep Learning Architecture

Overview

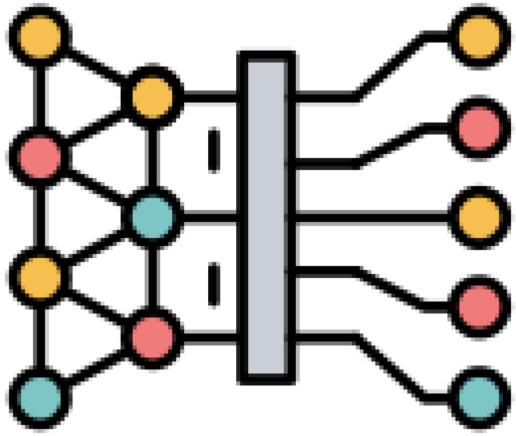
- ✓ **Feedforward Neural Networks**: Basic neural network structure.
- ✓ **Convolutional Neural Networks (CNNs)**: Best for image data.
- ✓ **Recurrent Neural Networks (RNNs)**: Ideal for sequential data (time series, text).
- ✓ **Generative Adversarial Networks (GANs)**: Used for generating data (synthetic images).



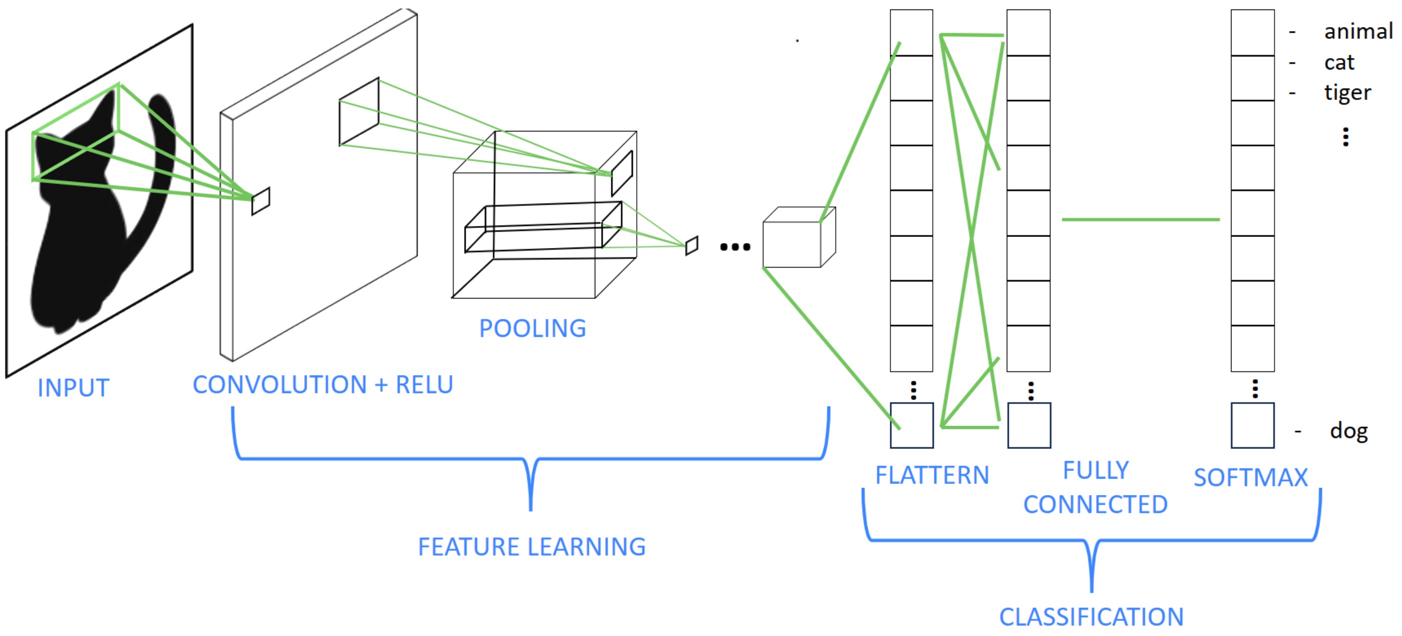
Deep Learning

Understanding CNNs

- ✓ **Convolutional Neural Networks (CNNs)** are a type of **deep learning model** specifically designed for **processing grid-like data**, such as images.
- ✓ CNNs have become the gold standard for many **computer vision tasks** like image classification, object detection, and facial recognition.
- ✓ A Convolutional Neural Network is a specialized type of neural network
  - ✓ Excels at analyzing visual data
  - ✓ Unlike traditional neural networks that treat every input feature as independent, CNNs take advantage of the spatial structure of images,
  - ✓ Detect patterns like edges, shapes, and textures



- **Convolutional Layers:** Extract features from images.
- **Pooling Layers:** Reduce dimensionality.
- **Fully Connected Layers:** Make predictions based on extracted features.



Note: CNNs are great for detecting patterns in images using convolutional layers. Pooling layers reduce computation, making models faster.

How do CNNs work?

How do CNNs work?

CNNs are composed of several layers, each with its specific purpose.

1. Input Layer: Takes in the raw image data.

2. Convolutional Layer: Extracts features from the input image using filters.

3. Activation Layer (ReLU): Introduces non-linearity into the model.

4. Pooling Layer: Reduces the spatial size of the feature maps, making the model more efficient.

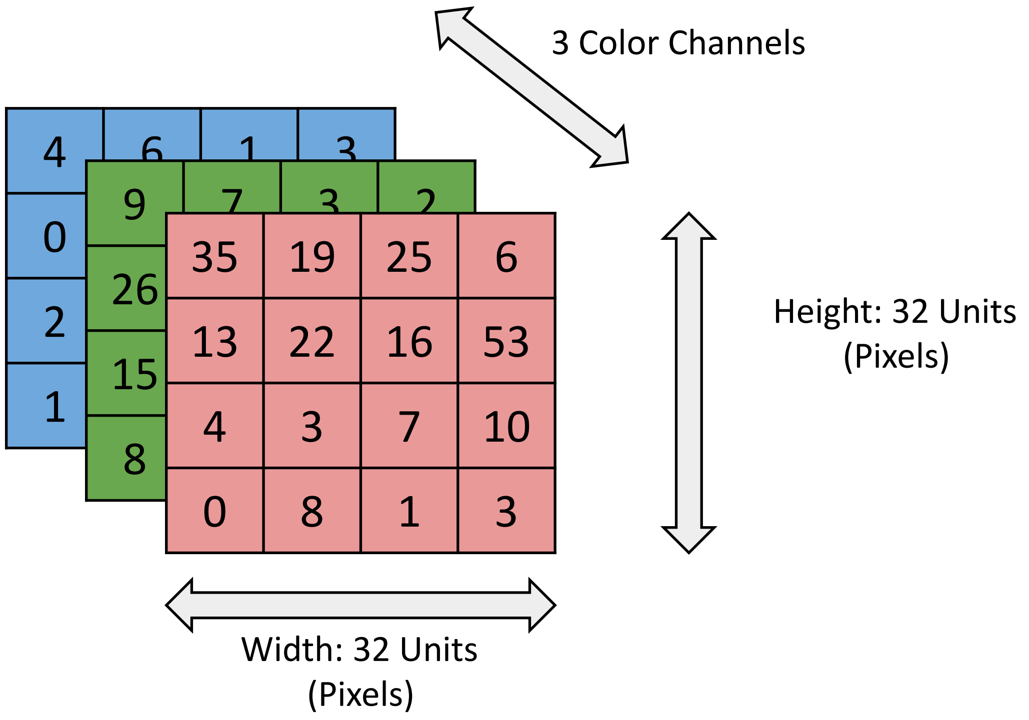
5.Fully Connected Layer: Combines all the features to make the final prediction.

6.Output Layer: Produces the final classification result (e.g., cat, dog, car).

Step-by-Step

Step 1: Input Layer

- ✓ The **input** to a CNN is an image represented as a matrix of pixel values.
- ✓ A **color image** typically has three channels (Red, Green, Blue) and can be represented as a 3D matrix: height x width x channels.
- ✓ For example, a 32x32 color image has a shape of (32, 32, 3).



Step 2: Convolutional Layer

- The first layer in a CNN is usually a **convolutional layer**.
- This layer applies small filters (kernels) over the input image to extract **features** like edges, textures, or patterns.
- A filter (3x3 matrix) slides over the input image, performing a **dot product** with the pixels to produce a **feature map** (also known as an activation map).

- ✓ The filter slides over the image and extracts patterns like vertical or horizontal edges.
- ✓ This process is repeated with multiple filters to extract different features.

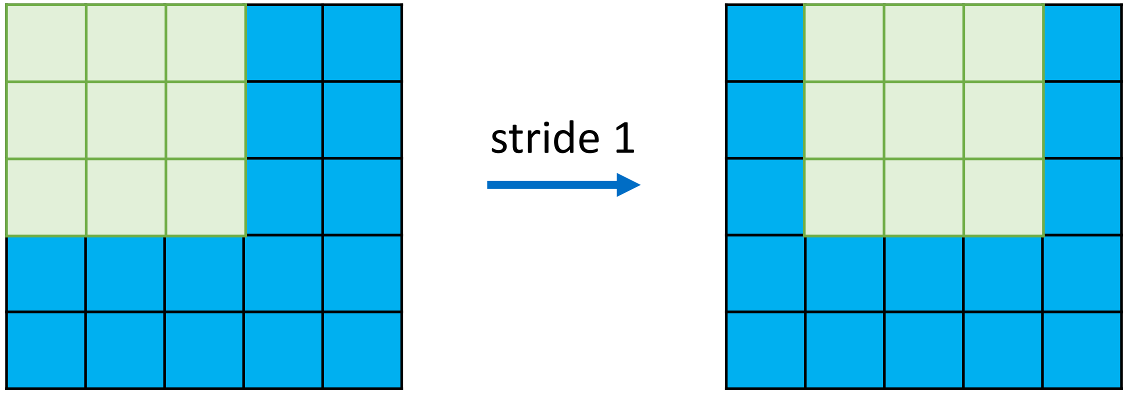
2	4	7	6	5
9	7	1	2	6
8	3	4	5	7
4	3	3	8	4
5	2	1	1	2

5 x 5
Input Image

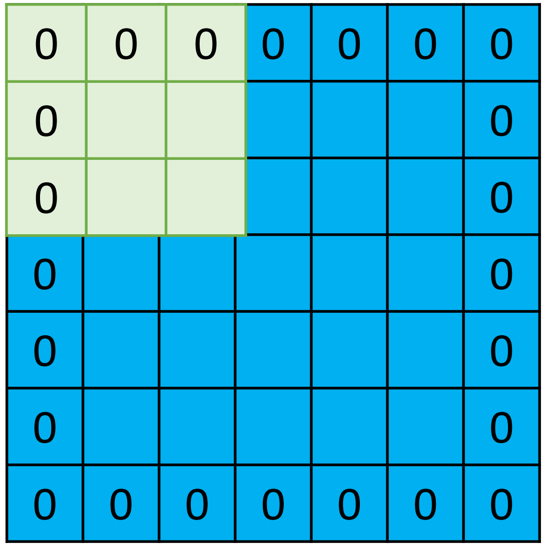
1	0	-1
1	0	-1
1	0	-1

3 x 3
filter

- ✓ The stride determines how much the filter moves at each step (e.g., stride of 1 moves one pixel at a time).
- ✓ Padding adds a border of zeros around the input to keep the output size the same as the input.



(Image with Stride)



(With Padding)

Step 3: Activation Function (ReLU)

- ✓ After the convolution operation, an **activation function** is applied to introduce non-linearity.
- ✓ The most common activation function is **ReLU (Rectified Linear Unit)**, which replaces all negative values with zero.
- ✓ ReLU helps the model learn complex patterns and speeds up training.

$$\boldsymbol{ReLU(x) = \max(0, x)}$$

This means that the output of the ReLU function is:

- x, if $x > 0$
- 0, if $x \le 0$

Step 4: Pooling Layer (Max Pooling)

- ✓ The **Pooling Layer** reduces the **spatial dimensions** (height and width) of the feature maps, keeping only the most important information.
- ✓ **Max Pooling** is the most common type, which selects the **maximum value** from each region covered by the filter.

- If you use a 2x2 filter with a stride of 2, the pooling layer takes the maximum value in each 2x2 block and reduces the size of the feature map.
- This reduces computational complexity and helps prevent overfitting.

1	3	2	5
8	7	4	6
3	1	9	2
4	2	1	0

$$\mathrm{Max}\left(\begin{matrix} 1 & 3 \\ 8 & 7 \end{matrix}\right) = 8$$

$$\mathrm{Max}\left(\begin{matrix} 2 & 5 \\ 4 & 6 \end{matrix}\right) = 6$$

$$\mathrm{Max}\left(\begin{matrix} 3 & 1 \\ 4 & 2 \end{matrix}\right) = 4$$

$$\mathrm{Max}\left(\begin{matrix} 9 & 2 \\ 1 & 0 \end{matrix}\right) = 9$$

8	6
4	9

Step 5: Fully Connected Layer (Dense Layer)

- ✓ After several convolutional and pooling layers, the feature maps are **flattened** into a 1D vector.
- ✓ This vector is fed into a **Fully Connected Layer** where each neuron is connected to every neuron in the previous layer.
- ✓ The fully connected layer learns to combine the extracted features to make predictions.

Fully Connected Layer

Step 6: Output Layer

✓ The **Output Layer** produces the final prediction using an activation function like **Softmax** (for multi-class classification) or **Sigmoid** (for binary classification).

✓ The result could be a label (e.g., "cat", "dog") or a probability distribution over classes.

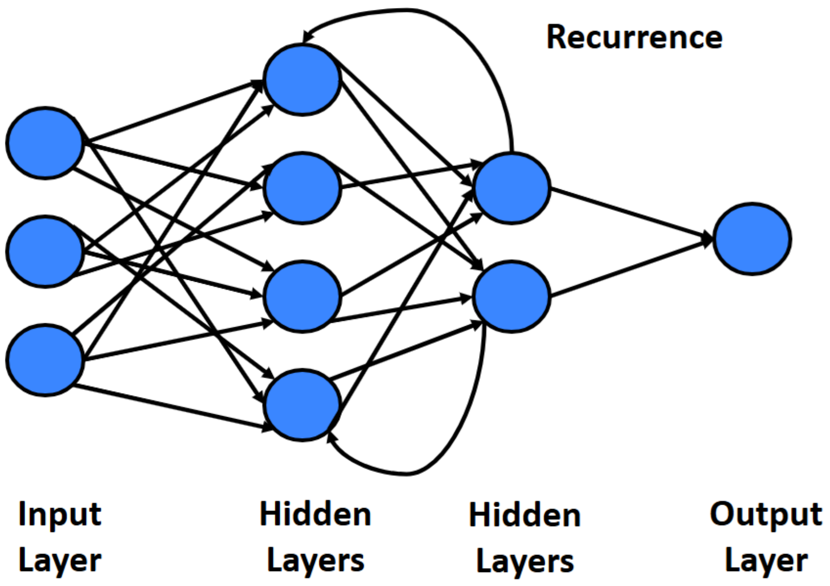
Softmax	Sigmoid
○ Decide who ("cat", "dog") gets how many percentage.	○ Decide between two things.
○ Shares everything equally (100%).	○ Only One, Just yes or no likelihood
○ Works for 2+ options (Cat, Dog, Rabbit, etc.,)	○ Works for 1 option (yes/no decision)

Practical Applications of CNNs

1. **Image Classification**: Recognizing objects in images (cats, dogs, cars).
2. **Object Detection**: Detecting and localizing multiple objects within an image (YOLO, SSD).
3. **Face Recognition**: Identifying people from photos (Face ID).
4. **Medical Imaging**: Detecting diseases from X-rays, MRIs, or CT scans.
5. **Self-Driving Cars**: Understanding the environment through camera feeds to make driving decisions.
6. **Video Analysis**: Analyzing video frames to detect actions or events.

Understanding RNNs

- ✓ **How RNNs Work:** Information flows in a loop, making them suitable for sequential data.
- ✓ **Use Cases:** Language translation, sentiment analysis, and stock price prediction.



Realistic Infotech Group
IT Training & Services
No.79/A, First Floor
Corner of Insein Road and
Damaryon Street
Quarter (9), Hlaing Township
Near Thukha Bus Station
09256675642, 09953933826
http://www.rig-info.com