# 📓 Day 15 – Types of Neural Networks and Activation Functions

### 🔑 Overview

On Day 15 of the AI-ML training at **A2IT InternEdge, Mohali**, we delved deep into one of the most essential foundations of **Deep Learning**:

- Understanding the **different types of Neural Networks** (NNs)
- The role and **types of Activation Functions**
- Visualization using Python and Matplotlib

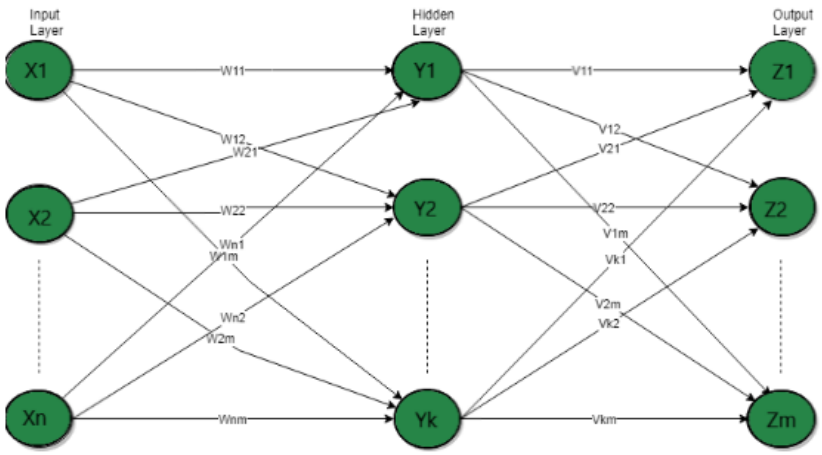
---

### ◆ 1. Types of Neural Networks

Neural Networks are at the heart of modern AI systems, inspired by the functioning of the human brain. There are various types of NNs used for different kinds of data and problems.

### ❖ 1.1 Feedforward Neural Network (FNN)

- **Structure**: Basic form where data flows in one direction — from input to output.
- **No memory**: It doesn't remember past inputs.
- **Use cases**: Spam detection, regression tasks, image classification.
- **Layers**:
  - Input Layer → Hidden Layers (with activations) → Output Layer

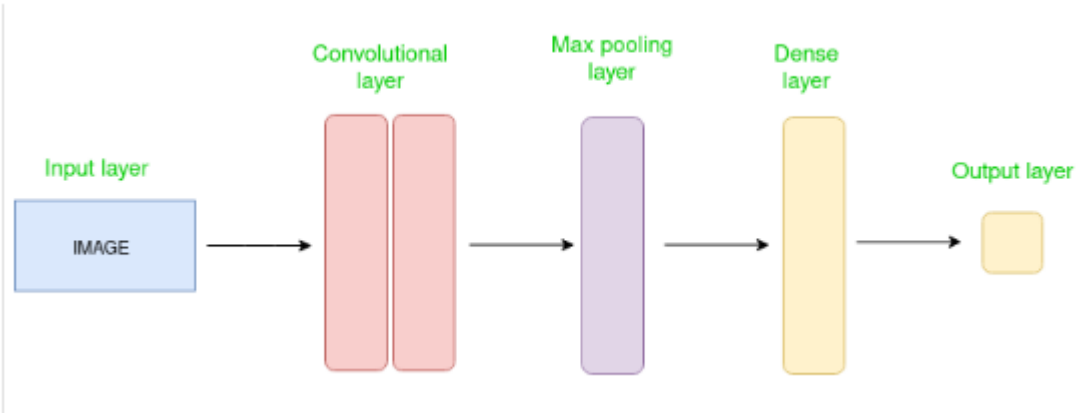



---

### ❖ 1.2 Convolutional Neural Network (CNN)

- Specially designed for **image processing** and **computer vision** tasks.
- Uses **filters/kernels** to automatically learn spatial hierarchies.
- **Key layers**:
  - Convolutional Layer
  - Pooling (MaxPooling)
  - Fully Connected Layer

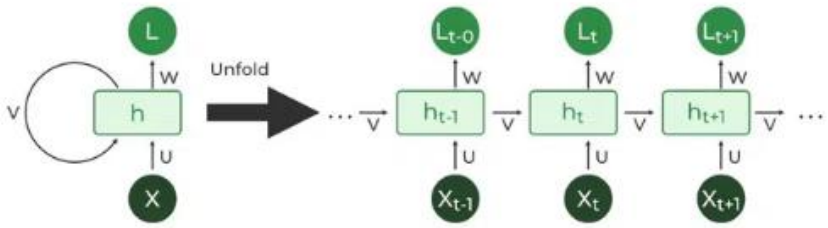
🔍 *Applications:* Face recognition, medical image analysis, traffic sign detection, object recognition.




---

### ◆ 1.3 Recurrent Neural Network (RNN)

- Designed for **sequence data** — like time-series, text, speech.
- Has **feedback loops**, so it can "remember" previous outputs.
- **Challenges**:
  - Vanishing gradients, short memory span.
- **Example Use**: Language translation, stock price forecasting.

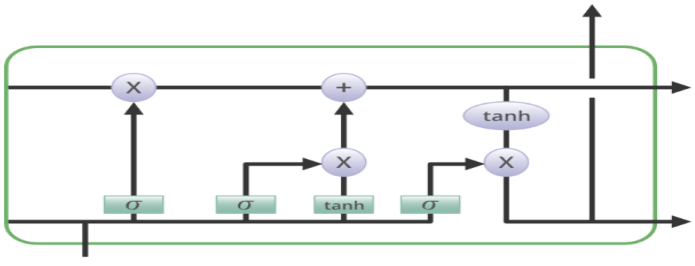



---

### ◆ 1.4 Long Short-Term Memory (LSTM)

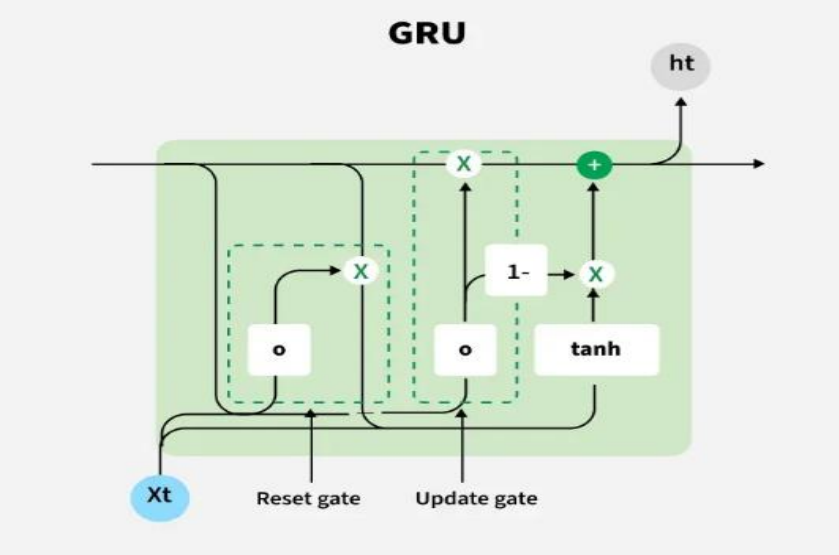
- A special type of RNN designed to handle **long-term dependencies**.
- Solves vanishing gradient problem.
- Uses gates (input, forget, output) to **control memory flow**.

🔍 *Applications:* Text prediction, chatbot engines, speech recognition.



### ❖ 1.5 Generative Adversarial Networks (GANs)

- Composed of two networks:
  - **Generator**: Creates fake data.
  - **Discriminator**: Evaluates real vs fake.
- Both improve each other through competition.
- Produces **high-quality synthetic data** like images, voice, etc.




---

## ◆ 2. Activation Functions

Activation functions add **non-linearity** to a model, allowing it to learn **complex patterns**.

Without activation functions, neural networks would act like **simple linear regression**.

### ❖ Common Activation Functions

| Activation Function | Mathematical Formula | Description |
|---|---|---|
| **ReLU** (Rectified Linear Unit) | `f(x) = max(0, x)` | Fast, simple. Preferred for hidden layers. Doesn't saturate for positive values. |
| **Sigmoid** | `f(x) = 1 / (1 + e^(-x))` | Smooth curve. Squeezes output to (0,1). Often used in binary classification. |
| **Tanh** | `f(x) = tanh(x)` | Output between (-1, 1). Centered at zero. Better than sigmoid for some tasks. |

---

### ◆ 3. Code Example: Visualizing Activation Functions

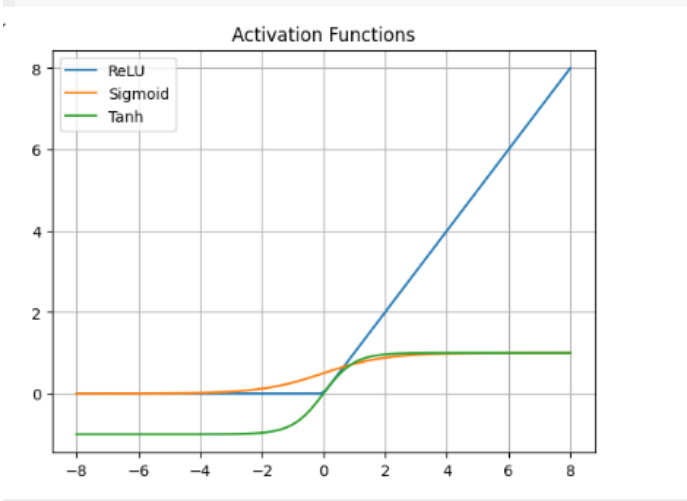
We used Python with NumPy and Matplotlib to plot ReLU, Sigmoid, and Tanh over input values from -8 to +8.

### 📌 Python Code:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-8, 8, 100)
relu = np.maximum(0, x)
sigmoid = 1 / (1 + np.exp(-x))
tanh = np.tanh(x)

plt.plot(x, relu, label='ReLU')
plt.plot(x, sigmoid, label='Sigmoid')
plt.plot(x, tanh, label='Tanh')
plt.legend()
plt.title("Activation Functions")
plt.grid(True)
plt.show()
```




---

### 📊 Graphical Output:

- **ReLU**: Straight line from 0 onward, flat at 0 before that.
- **Sigmoid**: S-shaped curve, compressing values to (0,1).
- **Tanh**: Similar to sigmoid but outputs between -1 and 1.

---

### □ 4. Why Are Activation Functions Important?

- Introduce **non-linear capabilities** so neural networks can learn from **complex data**.
- Help with learning important **features** through backpropagation.
- Choice of activation function affects **accuracy**, **training speed**, and **convergence**.

---

### 📈 Learning Outcomes

- Understood major **types of neural networks** and their architecture.
- Learned the role and mathematical behavior of activation functions.
- Visualized and compared the impact of each function.
- Developed intuition on **when to use** ReLU, Sigmoid, or Tanh.