📓 Day 15 – Types of Neural Networks and Activation Functions

🎗️ Overview

On Day 15 of the AI/ML training at **A2IT InternEdge**, Mohali, we took a deep dive into some of the most essential building blocks of deep learning:

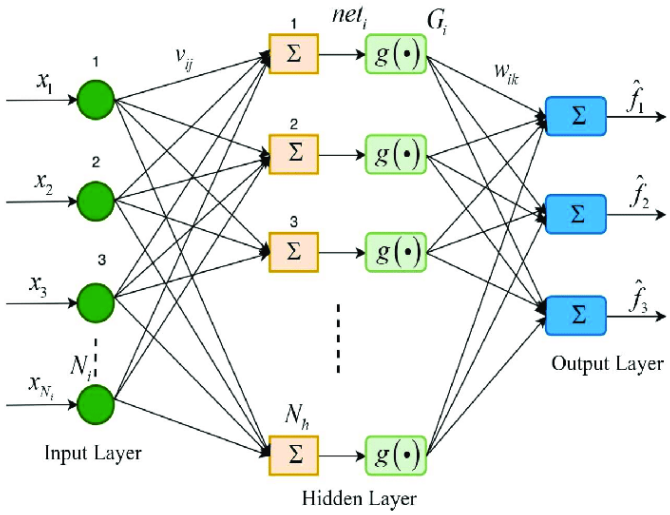
- Different types of Neural Networks (NNs)
- Activation functions and their role
- Graphical visualization of activations using Python

🔹 1. Types of Neural Networks

Neural Networks are inspired by the structure and working of the human brain. They're the backbone of deep learning and can be structured in various ways depending on the task.

🔷 1.1 Feedforward Neural Network (FNN)

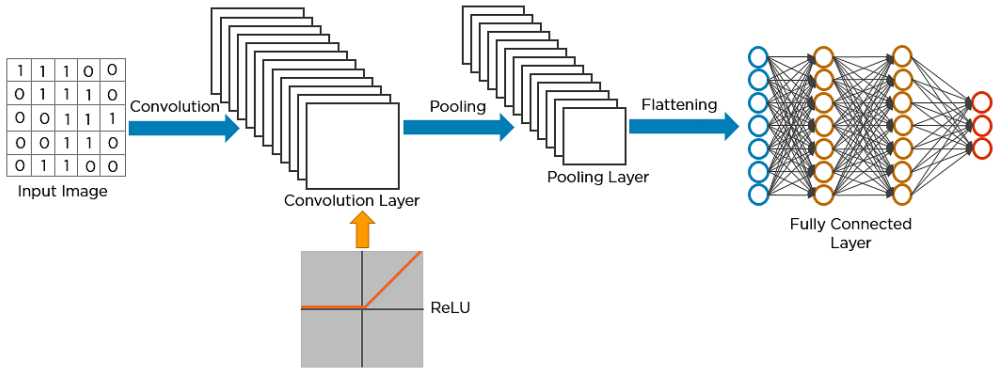
- **Structure**: Data moves in one direction — from input to output — with no loops.
- **Key Layers**: Input → Hidden → Output
- **Limitation**: Cannot process sequential data or remember past input.
- **Applications**: Spam filters, digit recognition, regression tasks.



🔷 1.2 Convolutional Neural Network (CNN)

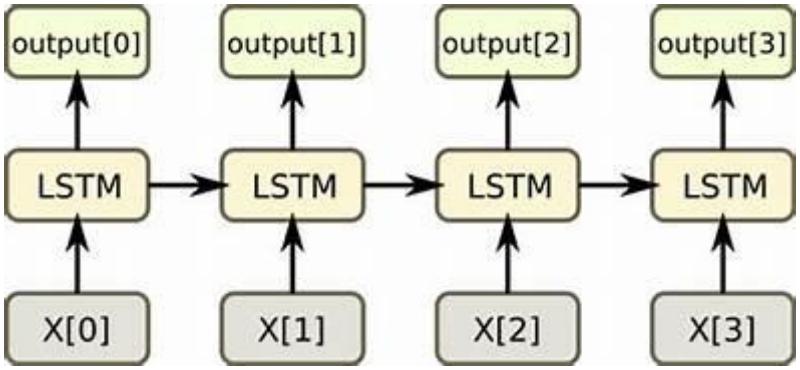
- Best suited for image and video data.
- Learns spatial features using **kernels** and **filters**.

- Includes Convolution → ReLU → Pooling → Fully Connected Layers.
- **Applications**: Face detection, medical imaging, object classification.



◆ 1.3 Recurrent Neural Network (RNN)

- Processes sequences like time-series, text, and speech.
- Maintains a hidden state to remember past inputs.
- Suffers from short-term memory and vanishing gradient issues.
- **Applications**: Sentiment analysis, music generation.



◆ 1.4 Long Short-Term Memory (LSTM)

- A special kind of RNN that remembers long-term dependencies.
- Uses **memory cells** and **gates** (input, forget, output).
- Solves the vanishing gradient problem effectively.
- **Applications**: Chatbots, translation, speech recognition.

◆ 1.5 Generative Adversarial Networks (GANs)

- Consist of two networks:
    - **Generator**: Creates fake data.

- **Discriminator**: Detects real vs fake.

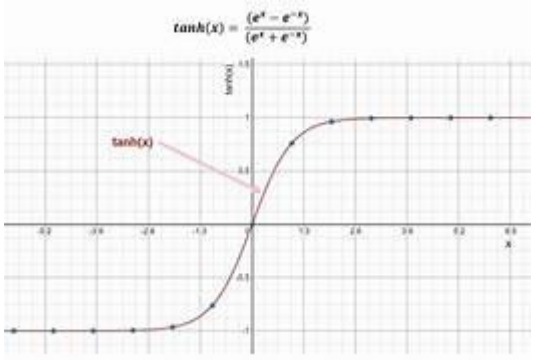
- Used to generate realistic fake images, deepfakes, synthetic voices, etc.

🔷 2. Activation Functions

Activation functions bring **non-linearity** into neural networks, allowing them to learn from complex and non-straightforward patterns.

🔹 Key Functions:

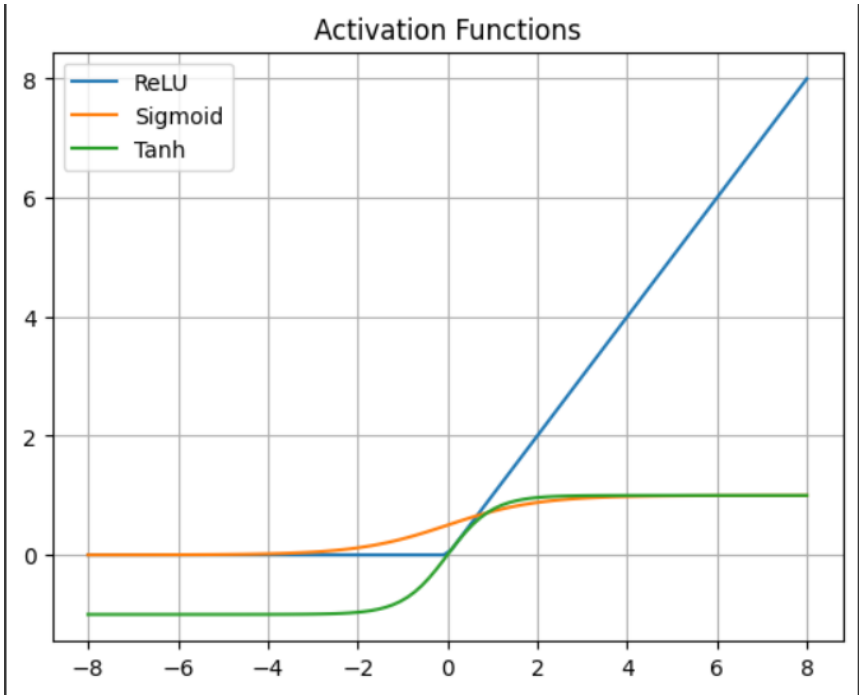
Function	Formula	Notes
ReLU (Rectified Linear Unit)	f(x) = max(0, x)	Fast, popular, doesn't activate negative values
Sigmoid	f(x) = 1 / (1 + e^(-x))	Compresses values between 0 and 1; used in binary classification
Tanh	f(x) = tanh(x)	Range: (-1, 1); centered around 0; better for some tasks than sigmoid



🔷 3. Python Code – Visualizing Activation Functions

```
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.title("Activation Functions")
plt.legend()
plt.grid(True)
plt.show()
```



- ReLU: 0 for negatives, linear for positives.
- Sigmoid: S-shaped, maps input between 0 and 1.
- Tanh: Smooth curve between -1 and 1.

🔷 4. Why Do We Need Activation Functions?

- They help the network **learn complex patterns**.
- Make deep learning models **powerful and flexible**.
- Without them, even multi-layered models behave like simple linear regression.

✅ Learning Outcomes

- Differentiated between various NNs like FNN, CNN, RNN, LSTM, and GAN.
- Understood when to use each type based on the problem/data.
- Explored and visualized how activation functions behave.
- Gained hands-on coding experience using Python + Matplotlib.
