Assignment - 4

Name: Anurag Pathak
Class: A_aiml
Roll.no: A3_52

Aim: Write a Report on Neural Network Architectures

Objective: Explore the design and use cases of various neural network architectures.

1. What is a neural network?

Neural networks are machine learning models that mimic the complex functions of the human brain. These models consist of interconnected nodes or neurons that process data, learn patterns, and enable pattern recognition and decision-making tasks.

2. Components of a Simple Neural Network

A fundamental neural network consists of several essential elements:

a) Layers

Neural networks are built using different layers:

- **Input Layer**

  The input layer is the first layer of a neural network and serves as the entry point for raw data. Each neuron in this layer corresponds to a specific feature of the input, such as individual pixels in an image, words in a text document, or numerical values in a dataset. The number of neurons in the input layer is determined by the dimensionality of the dataset, ensuring that all relevant information is passed to the subsequent layers for processing. This layer does not perform any computations; instead, it simply transmits the data to the hidden layers, where transformations and learning take place

- **Hidden Layer(s)**

  These layers perform most of the computational heavy lifting. A neural network can have one or multiple hidden layers. Each layer consists of units (neurons) that

transform the inputs into something that the output layer can use.

1. **Fully Connected (Dense) Layer:** Each neuron in this layer connects to every neuron in the previous and next layers, allowing full interaction.
2. **Convolutional Layer (CNNs):** Used in image processing tasks to detect patterns like edges and textures.
3. **Recurrent Layer (RNNs):** Designed for sequential data, such as time-series forecasting or language modeling.

- **Output Layer**

  This is the final layer of the network, responsible for producing predictions.

  The number of neurons in this layer depends on the type of problem being solved:

1. **For binary classification:** A single neuron with a **sigmoid activation function** outputs values between 0 and 1.
2. **For multi-class classification:** Multiple neurons with a **softmax activation function** assign probabilities to different classes.

3. **For regression tasks:** A single neuron with no activation function (or a linear function) outputs continuous values.

b) Activation Functions

Activation functions introduce non-linearity, enabling the network to capture complex relationships. Some widely used activation functions include:

- **Sigmoid**: Maps values between 0 and 1, commonly used in binary classification.
- **ReLU (Rectified Linear Unit)**: Converts negative inputs to zero while allowing positive values to pass unchanged, helping deep networks train efficiently.
- **Softmax**: Converts raw scores into probabilities, making it useful for multi-class classification problems.

c) Loss Functions

Loss functions measure how different the network's predictions are from actual values. Some commonly used loss functions include:

- **Mean Squared Error (MSE)**: Used in regression problems to measure prediction errors.

- **fCross-Entropy Loss**: Frequently applied in classification tasks to assess the difference between predicted and actual probabilities.

d) Optimization and Learning Process

Neural networks learn by modifying their internal weights using a process called backpropagation, which works with optimization algorithms like **Stochastic Gradient Descent (SGD)** or **Adam** to minimize error.

3. Comparison between shallow networks and deep networks.

Feature	Shallow Networks	Deep Networks
Number of Layers	One or two hidden layers	Multiple hidden layers
Feature Extraction	Requires manual selection of features	Automatically identifies relevant features
Complexity	Simple architecture, easier to train	More complex but better at capturing patterns
Performance	Suitable for basic tasks	Effective for complex problems

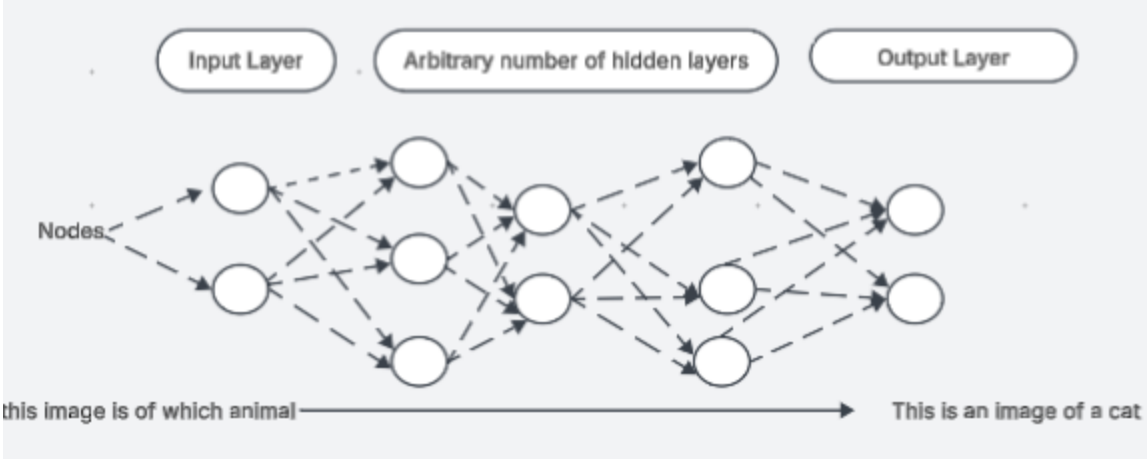
		like image and speech recognition
Training Time	Faster	Requires more computational resources

4. Applications of Simple Feedforward Neural Networks

Feedforward neural networks (FNNs) are the simplest form of neural networks, where data flows in a single direction from input to output. Some practical applications include:

- **Handwriting Recognition:** Used in document digitization and banking systems.
- **Spam Detection:** Helps filter unwanted emails by analyzing content patterns.
- **Medical Diagnosis:** Assists doctors in detecting diseases from medical records.
- **Stock Market Prediction:** Identifies financial trends based on historical data

5.Include diagrams to illustrate a simple neural network.



6. Conclude with a summary of how Keras/TensorFlow simplifies neural network design.

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense


# Create a basic feedforward neural network

model = Sequential([

    Dense(16, activation='relu', input_shape=(10,)),  # Hidden layer with 16 neurons

    Dense(1, activation='sigmoid')  # Output layer for binary classification

])
```

```
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

model.summary()
```