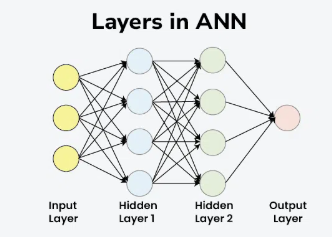
**DIFFERENT TYPES OF HIDDEN LAYER IN DEEP LEARNING**

**ASSIGNMENT:** 01 **NAME:** SRIRANJANI.S **REG\_NUMBER :**122011173026

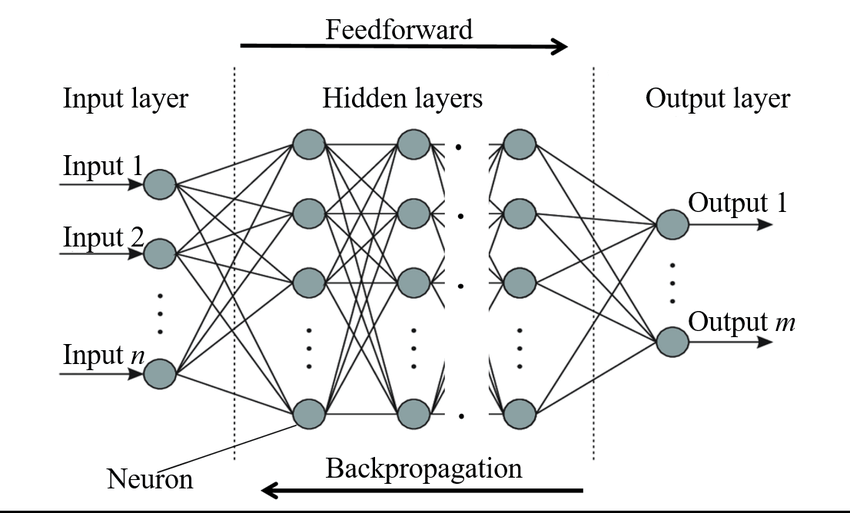
**COURSE\_NAME:** APPLIED DEEP LEARNING **COURSE\_CODE:** XAI602C

**INTRODUCTION**

* Deep learning is a type of artificial intelligence (AI) that allows computers to learn from data and develop sophisticated algorithms. It uses multiple layers of mathematical computations – known as ‘hidden layers’ – to recognize patterns in large datasets and make decisions or predictions based on these patterns.
* In an [ANN](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/), data flows from the input layer, through one or more hidden layers, to the output layer. Each layer consists of neurons that receive input, process it, and pass the output to the next layer. The layers work together to extract features, transform data, and make predictions.
* An ANN typically consists of three primary types of layers:
  + Input Layer
  + Hidden Layers
  + Output Layer
* Each layer is composed of nodes (neurons) that are interconnected. The layers work together to process data through a series of transformations.
* The number of hidden layers needed for any given deep learning model largely depends on the complexity of the problem it must solve. Typically, more complex problems may involve a deeper network with more hidden layers, while simpler problems may only require few if any at all. Most deep learning approaches generally involve between one and 15 hidden layers; though deep networks with over 150 exist depending on the application requirements

**TYPES OF HIDDEN LAYERS IN ARTIFICIAL NEURAL NETWORKS**

**1. Dense (Fully Connected) Layer**

A dense layer is the most common type of hidden layer in an ANN. Every neuron in a dense layer is connected to every neuron in the previous and subsequent layers. This layer performs a weighted sum of inputs and applies an activation function to introduce non-linearity. The [activation function](https://www.geeksforgeeks.org/activation-functions-neural-networks/) (like [ReLU](https://www.geeksforgeeks.org/why-is-relu-used-as-an-activation-function/" \t "_blank), [Sigmoid](https://www.geeksforgeeks.org/derivative-of-the-sigmoid-function/), or Tanh) helps the network learn complex patterns.

**Key Points:**

* **Role**: Learns representations from input data.
* **Function**: Performs weighted sum and activation.
* **Example**: Common in fully connected neural networks.

**2. Convolutional Layer**

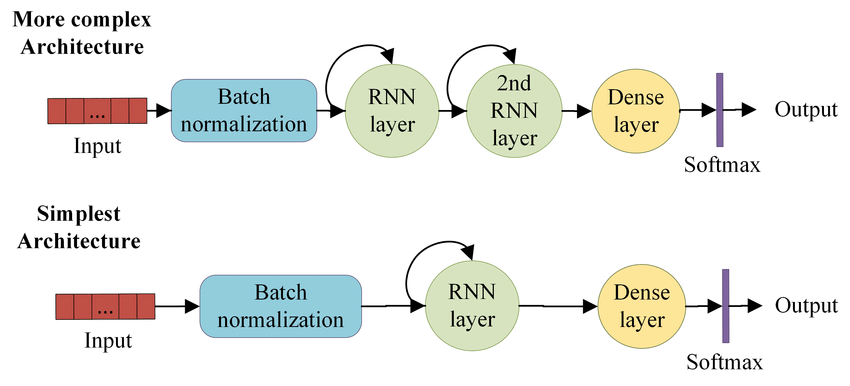
Convolutional layers are primarily used in [Convolutional Neural Networks (CNNs)](https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/) for image processing tasks. They apply convolution operations to the input, capturing spatial hierarchies in the data. Convolutional layers use filters to scan across the input and generate feature maps. This helps in detecting edges, textures, and other visual features.



**Key Points:**

* **Role**: Extracts spatial features from images.
* **Function**: Applies convolution using filters.
* **Example**: Detects edges and textures in images.

**3. Recurrent Layer**

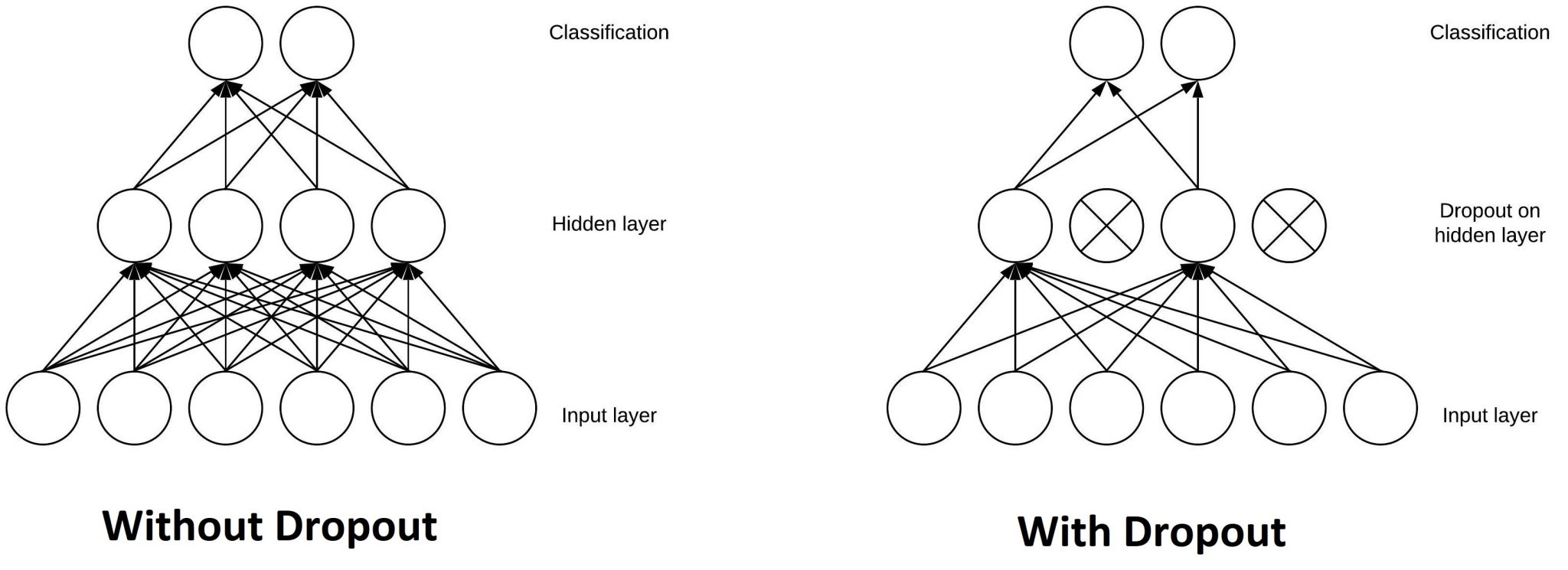
Recurrent layers, such as [Long Short-Term Memory (LSTM)](https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/) and [Gated Recurrent Unit (GRU),](https://www.geeksforgeeks.org/gated-recurrent-unit-networks/) are used in [Recurrent Neural Networks](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/) (RNNs) for sequence data like time series or natural language. They have connections that loop back, allowing information to persist across time steps. This makes them suitable for tasks where context and temporal dependencies are important.

**Key Points:**

* **Role**: Processes sequential data with temporal dependencies.
* **Function**: Maintains state across time steps.
* **Example**: Language modeling, time series prediction.

**4. Dropout Layer**

[Dropout layers](https://www.geeksforgeeks.org/dropout-in-neural-networks/) are a [regularization](https://www.geeksforgeeks.org/regularization-in-machine-learning/) technique used to prevent overfitting. They randomly drop a fraction of the neurons during training, which forces the network to learn more robust features and reduces dependency on specific neurons. During training, each neuron is retained with a probability ppp.

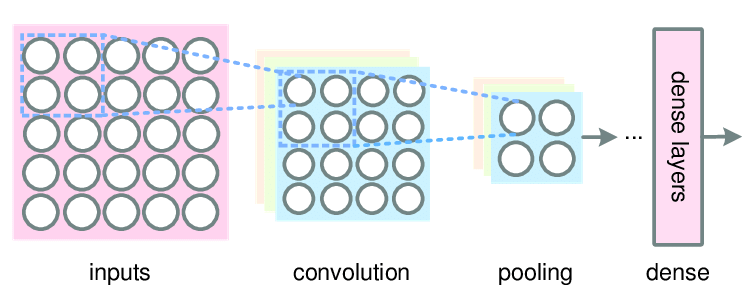


**Key Points:**

* **Role**: Prevents overfitting.
* **Function**: Randomly drops neurons during training.
* **Example**: Common in[deep learning](https://www.geeksforgeeks.org/deep-learning-tutorial/) models to improve generalization.

**5. Pooling Layer**

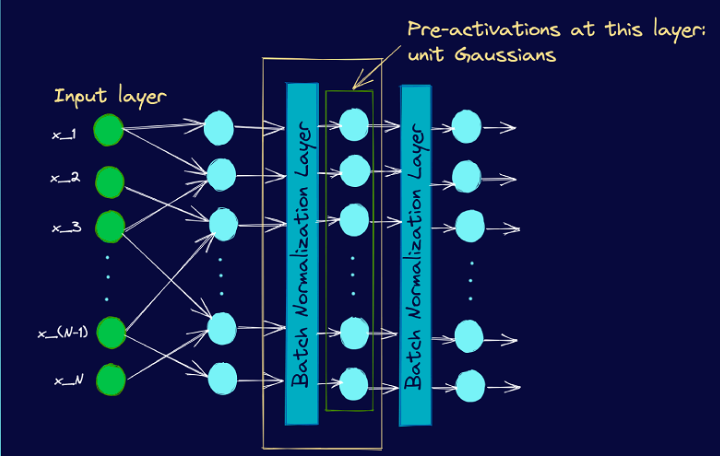
A [**Pooling Layer**](https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/) is used to reduce the spatial dimensions of the data, thereby decreasing the computational load and controlling overfitting. Common types of pooling include Max Pooling and Average Pooling.



**Use Cases:** Dimensionality reduction in CNNs

**6. Batch Normalization Layer**

A [**Batch Normalization Layer**](https://www.geeksforgeeks.org/what-is-batch-normalization-in-cnn/) normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. This helps in accelerating the training process and improving the performance of the network.



**Use Cases:** Stabilizing and speeding up training

**REFERENCE**

[Topic DL01: Activation functions and its Types in Artifical Neural network | by abhigoku10 | Medium](https://abhigoku10.medium.com/activation-functions-and-its-types-in-artifical-neural-network-14511f3080a8)

[Fully connected (dense) artificial neural network. | Download Scientific Diagram](https://www.researchgate.net/figure/Fully-connected-dense-artificial-neural-network_fig1_358145060)