Steps to run the code:

A. **Pre-Processing:**
   To run data pre-processing, open the PREPROCESSING.ipynn file and load the source data from the source data folder.
   Note: Renaming of file has been done, so check for correction in the code for file name only.
   You will get 3 pre-processed files which as been stored in the Pre-Processed Data Folder.

B. **Splitting the Data:**
   The splitting of data has been done on year basis and happens in Day 2 of the PREPROCESSING.ipynn. The result has been stored in the Test_Train_Data Folder.

C. **CNN Model 1:**
   Load the Train and test data from the Test_Train_Data Folder and run the code to get the desired results. The result is based on Rainfall prediction in millimetres.

D. **CNN Model 2:**
   Load the Train and test data from the Test_Train_Data Folder and run the code to get the desired results. The result is based on Rainfall outcome as zero and ones.

E. **MLP Model/ANN Model :**
   Load the Train and test data from the Test_Train_Data Folder and run the code to get the desired results. The result is based on Rainfall outcome as zero and ones.

Experimentation:

In this project, Data Collection has been done from Bureau of Meteorology, Australia. Rainfall in millimetres , Maximum Temperature and Minimum Temperature were collected from the Bureau of Meteorology, Australia.
Additional feature parameters such as SOI, Nino 1.2, Nino 3.0, Nino 4.0, DMI, Sunspot, IPO, TPI etc could not be included in the model due to variance of the source data. Thus, the complete model was built on three features or attributes: Daily Rainfall in millimetres , Maximum Temperature and Minimum Temperature.

S.No	Attribute	Source
1	Daily Rainfall in mm	Bureau of Meteorology, Australia
2	Maximum Temperature	Bureau of Meteorology, Australia
3	Minimum Temperature	Bureau of Meteorology, Australia

Data Pre-Processing:

Data acquired from the site had to be pre-processed in order to use as input features. Following Pre-Processing Steps were taken:

1. Downloading the Source File
2. Reading the Source File
3. Dropping the un-necessary columns in the Source File
4. Only data after year 1957 was considered due to negligible invariance.
5. Data was checked for null values.
6. The null values were replaced by average values
7. Three csv files were obtain, one for each feature, which was joined using "index-Joining"
8. A split was made to divide the data into test and training groups based on the year 2010.

Pre-Processing in every Model:

Following pre-processing steps were used in every model code:

1. Splitting of Train and Test data into X_train, X_test, y_train,y_test.
2. Month were made as categorical variable and One Hot Encoding was done on the month column
3. Then the data was fed to the model.

Convolutional Neural Network – I:

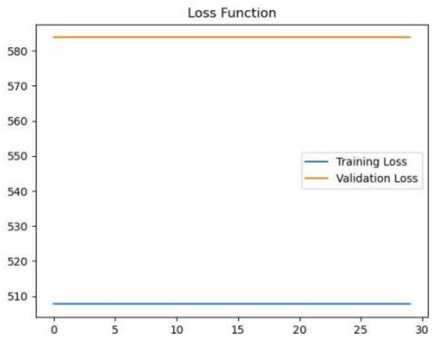
The first CNN was trained using the outcome as Rainfall in mm, i.e the measure of the exact rainfall on that day.

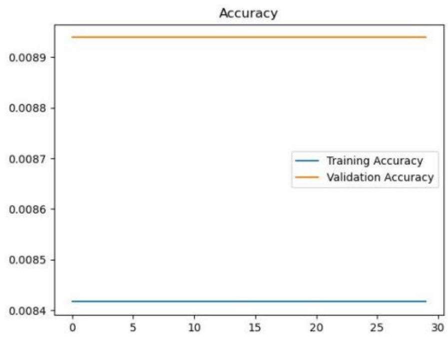
The architecture was as following:

Convolutional Layer – I → Pooling Layer – I → Convolutional Layer – II → Pooling Layer – II → Flatten → Dense Layer

Layer	Type	Activation	Filter Size	Nb Filter	Pool Size	Neuron
Conv1	Convolutional Layer	Relu	4	24		
Pool1	Pooling Layer				2	
Conv2	Convolutional Layer	Relu	2	32		
Pool2	Pooling Layer				2	
FCl	Fully Connected Layer	Hyperbolic tangent				1

Results:

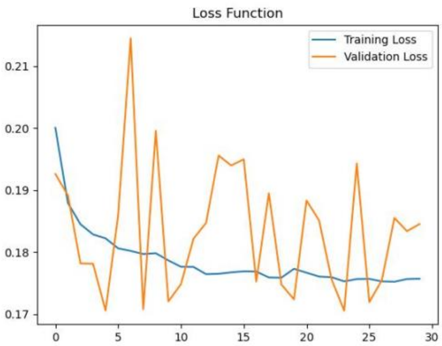


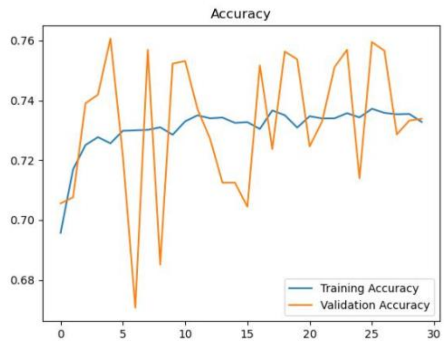


Convolutional Neural Network – II:

The second CNN was trained using the outcome as Rainfall Outcome i.e has their been rainfall on that day. All the details were the same.

Results:

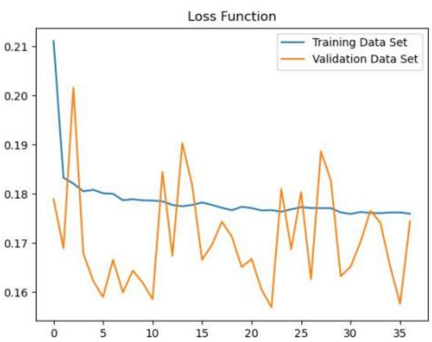


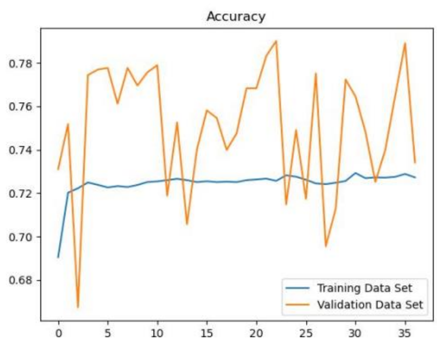


Artificial Neural Network:

The data was also fed to a ANN model to predict the outcome.

Result:





Artificial Neural Networks:

Artificial Neural Networks (ANNs) stand as one of the most transformative advancements in the field of artificial intelligence and machine learning. Inspired by the intricate network of neurons in the human brain, these computational models have revolutionized various industries, from healthcare and finance to autonomous vehicles and natural language processing.

At the heart of artificial neural networks lies the concept of deep learning, a subfield of machine learning that mimics the way the human brain learns and processes information. ANNs consist of interconnected nodes, or "neurons," organized into layers. Each neuron receives input signals, performs a computation, and produces an output signal that is transmitted to subsequent layers.

Architecture of Artificial Neural Networks:

Input Layer: The input layer receives raw data or features that serve as input to the neural network. Each neuron in the input layer represents a feature or attribute of the data.

Hidden Layers: Hidden layers, situated between the input and output layers, perform complex computations and learn hierarchical representations of the input data. Deep neural networks contain multiple hidden layers, enabling them to learn intricate patterns and relationships in the data.

Output Layer: The output layer produces the final prediction or classification based on the processed input. The number of neurons in the output layer depends on the nature of the task, such as regression, classification, or sequence generation.

Convolutional Neural Networks:

Convolutional Neural Networks are a class of deep learning models inspired by the organization and functioning of the human visual cortex. They are specifically designed to process and analyse visual data, making them ideal for tasks such as image classification, object detection, and image segmentation.

Architecture of Convolutional Neural Networks:

1. Convolutional Layers: The core building blocks of CNNs, convolutional layers apply filters or "kernels" to input images, extracting features such as edges, textures, and shapes. These filters slide across the input image, performing element-wise multiplication and aggregation to create feature maps.

2. Pooling Layers: Pooling layers reduce the spatial dimensions of the feature maps, enhancing computational efficiency and promoting translation invariance. Common pooling operations include max pooling, average pooling, and global pooling.

3. Fully Connected Layers: Fully connected layers connect every neuron in one layer to every neuron in the subsequent layer, enabling high-level feature learning and classification. These layers typically reside at the end of the CNN architecture and perform tasks such as image classification or object detection.