

Practical Assignment No - 1

| | |
|----------|----|
| Page No. | 13 |
| Date | 7 |

Name - shraddha kunwar

class - BE Comp A [A]

Roll no - 20C0063

sub - Laboratory Practice V (Deep Learning)

Aim :-

Linear Regression by Using deep neural network implementation boston house Pricing Prediction problem by linear Regression Using deep neural network Use Boston house Prediction dataset.

Dataset Description :-

The Boston housing dataset contains information about various features of houses in Boston along with their corresponding prices. There are 506 instances & 14 attributes in the dataset.

Prerequisite :-

python , Jupyter Notebook , Numpy , Pandas ,
matplotlib , scikit - learn , Tensorflow .

Theory :-

Linear Regression :-

It is a statistical method used to model the relationship between a dependent variable & one or more independent variables. The goal of the linear regression is to find the line of the best fit that describes the relationship between the variables.

Deep Neural Network :-

They are a type of Machine learning algorithm that are modelled after the structure & function of human brain. They are made up of multiple layer processing & transferring the input data in a non-linear way.

To implement linear regression & deep neural network models to predict the houses in Boston.

Here are the steps:-

- ① Load the boston dataset from scikit-learn. You can use the 'load_boston()' function from the 'sklearn.datasets' module to load the datasets.
 - ② Split the dataset into training & testing sets. You can use the 'train-test-split()' function for the 'sklearn.model_selection' module to split the dataset.
 - ③ Scale the features from standard scalar from scikit-learn scaling the features its important because it ensures the each feature has a similar range of values).
 - ④ Reshape the target variable to a 2D array.
→ the target variable is currently a 1D array, but it needs to be reshaped to a 2D array for compatibility with the deep neural network model.
 - ⑤ Implement a linear regression model using scikit learn & train it on the training set.
 - ⑥ Implement the deep neural network model using Tensorflow & train it on the training set.
You can use the keras API in tensorflow to implement the deep neural network model.
- Here are the 3 steps to follow:-

- Define the model architecture using the 'sequential' function from the tensorflow `Model` module.
- add the input layers using the 'Dense()' function from the 'tensorflow' layer module.
- Add one or more hidden layer using 'Dense()'
- Add one output layer using 'Dense()' function.
- Compile the model using 'compile()' function.
- Train the model using the 'fit()' function.
- ① predict the prices of houses in test set Using `Predict()` function.
- ② Evaluate & Compare performances of both model.

Conclusion :-

In this assignment we learned how to implement linear regression & deep neural network models to predict the price of houses in boston.

Practical Assignment No - 2

| | |
|----------|----|
| Page No. | 16 |
| Date | |

Name - Shradha Kunwar,

Class - BE [Comp A]

Roll no - 20co063

Subject - Laboratory Practice - V [Deep learning]

Aim :- Classification Using Deep Neural Network.

Binary classification using Deep Neural Network.

Networks : classify movie reviews into "positive" reviews & "negative" reviews just based on the text content of the reviews use IMDB dataset.

Dataset Description :-

We will use the IMDB datasets which contains 50,000 movie reviews that are labelled as "positive" or "negative". The dataset is split into 25000 reviews for training & 25000 reviews for testing.

Prerequisites :-

Python , Jupyter Notebook Numpy , scikitlearn , Keras , TensorFlow .

Theory :-

Binary classification is a type of machine learning problem where the task is to classify data into two categories. In this Practical assignment we will use deep neural networks to perform binary classification of movie reviews based on their text content. Deep neural network are a type of machine learning model that are capable of learning complex patterns of data.

STEPS :-

- ① Loads the dataset using built in function in keras.
- ② Preprocess the dataset by converting the integer sequences into a binary matrix using one-hot encoding.
- ③ Split the training dataset into training & validation sets.
- ④ Implement a deep neural network with following architecture :-
 - An embedding layer to convert the integer sequence into dense vectors of fixed size.
 - Two dense layers with ReLU activation functions.
 - A final dense layer with a sigmoid activation function to output probability for "positive" or "negative".
- ⑤ Train the model using the Adam optimizer & binary cross-entropy loss function.
- ⑥ Evaluate the model on the test dataset & report the accuracy & loss.
- ⑦ Experiment with different hyperparameters such as number of hidden units & learning rate & evaluate the model performance.
- ⑧ Save the trained model for future use.

Conclusion :- Hence we have successfully implemented binary classification for IMDB dataset.

Practical Assignment No - 3

| | |
|----------|----|
| Page No. | 18 |
| Date | |

Name - Shraddha Kunwar.

Roll no - 20C0063

Class - BE Comp [A]

Sub :- LP [V] : (Deep Learning)

Aim :- Convolutional Neural Networks (CNN), (Any one from the following)

- Use MNIST Fashion dataset & Create a classifier to classify fashion clothing into categories.

Requirements:-

64 bit operating systems, Python installation, Jupyter Notebook, Python libraries, TensorFlow, Numpy & Pandas.

Theory :-

Convolutional Neural Network (CNN) :-

It is a type of neural network commonly used in deep learning for image recognition, classification & segmentation tasks. The architecture of a CNN is designed to automatically & adaptively learn spatial hierarchies of features from input images.

To a CNN, the input image is first passed through a series of convolution layers, each of which applies a set of filters to the input image to extract features at different spatial scales. These features are then passed through a pooling layer which reduces the dimensionality of features while preserving their important spatial information.

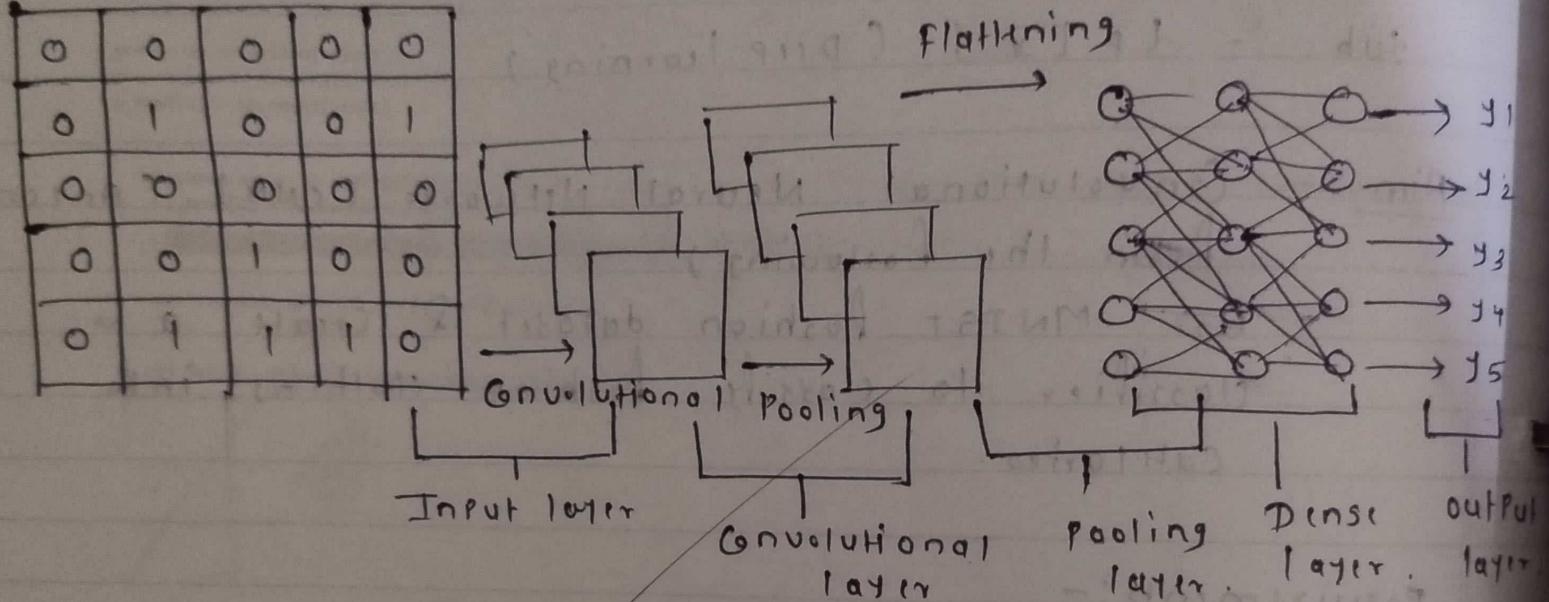


Fig : Architecture of CNN

Algorithm :-

STEPS :-

- ① START
- ② Import all the necessary libraries like Tensorflow etc.
- ③ Load the dataset & split it into train & test dataset.
- ④ Normalize the data to values between 0 & 1.
- ⑤ Explore the data.
- ⑥ Preprocess the datasets.
- ⑦ Build the model.
- ⑧ Train the model.
- ⑨ Feed the model.
- ⑩ EVALUATE accuracy.
- ⑪ make prediction.
- ⑫ STOP.

Conclusion :-

Hence we prepared a model to create a CNN classifier using MNIST fashion Dataset to classify fashion ~~clothing~~ into categories.

Page :-

DL [Experiment No-4]

Date : 20

Aim :- Recurrent Neural Network (RNN) :-

Use the Google Stock Prices dataset and design a time series analysis system using RNN.

Requirements :- 64 bit operating systems Python installation, Jupyter Notebook, Python libraries.

Theory :-

Recurrent Neural Network (RNN) :-

A Recurrent Neural Network (RNN) is a type of artificial neural network designed to handle sequential data or time series data. Unlike traditional deep neural networks that assume inputs & outputs are independent, RNNs have a unique ability to remember information from prior inputs to influence the current input & output. This memory feature allows RNNs to process sequences of data where the order is crucial, making them suitable for tasks like language translation, natural language processing, speech recognition & image captioning.

- How RNN's work :-
- Information flow :- To an RNN, information cycles through a loop, allowing it to consider both the current input & what it has learned from previous inputs.
- Memory state :- RNNs have a hidden state that remembers information about a sequence, allowing them to retain & utilize past inputs.

- Internal Memory :- RNN's have internal memory which enables them to remember important aspects of the input they receive, making them precise in predicting future data points.
- Looping Structure :-
RNNs loop over a single hidden layer multiple times, standardizing activation-functions and weights to maintain memory across different time steps.

How Deep Neural Network Work on classification:-

Deep Neural Network (DNNs) can be used for classification tasks Using Recurrent Neural Network Through a process called sequence classification.

Here's an overview of how DNN's work for classification Using RNN's .

- 1) Data Representation :- Input data is sequential , like txt or time series represented as vectors
- 2) RNN Architecture :- RNN's process sequences with recurrent connections , updating hidden states at each time step.
- 3) Training :- Input sequences are fed into the RNN & parameters are adjusted to minimize loss using backpropagation through time (BPTT)

4> **OUTPUT Layer :-**

RNN output layer computes class probabilities using softmax activation.

5> **Inference / Prediction :-** Trained RNN predicts class labels for new sequences by selecting the class with the highest probability.

6> **Evaluation :-** Model performance is assessed using metrics like accuracy, precision, recall on a separate validation dataset.

Recurrent Neural Networks find applications across various domains due to their ability to model sequential data.

1> **Natural Language Processing :-**

- language modeling :- predicting the next word in a sequence of text.

- sentiment Analysis :- classifying the sentiment of text (positive, negative, neutral)

2> **Time series Analysis & forecasting :-**

- stock price prediction :- forecasting future stock prices based on historical data.

- weather forecasting :- Predicting future weather conditions using historical weather data.

3) Healthcare :-

- Predictive maintenance :- forecasting equipment failure or maintenance needs in health care equipment like MRI machines.

Conclusion :-

Hence , We have successfully designed a time series analysis system Using RNN , leveraging google stock prices dataset.

~~WTF~~