Page No.
Date

Lab Assignment No. 1

LP-II : Deep Learning

Aim :

Linear Regression by using Deep Neural Network. Implement Boston housing price prediction problem by linear regression using Deep Neural Network. Use Boston House price prediction dataset.

Requirements :

64-bit Windows O.S., Python, Python libraries : TensorFlow, pandas, matplotlib, etc. Jupyter Notebooks.

Theory :

Linear Regression :

It is a simple but powerful statistical method that aims to model the relationship between a dependent variable (also known as response variable) and one or more independent variables.

In deep learning, linear regression is used in as a basic building block for more complex models and also as a component of more complex model. In neural networks, linear regression can be used as a way to combine inputs features to generate a single output.

Deep Neural Network :

A Deep Neural Network is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANN's, Deep Neural Networks can model

complex non-linear relationships.

The main purpose of a neural network is to recieve a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification.

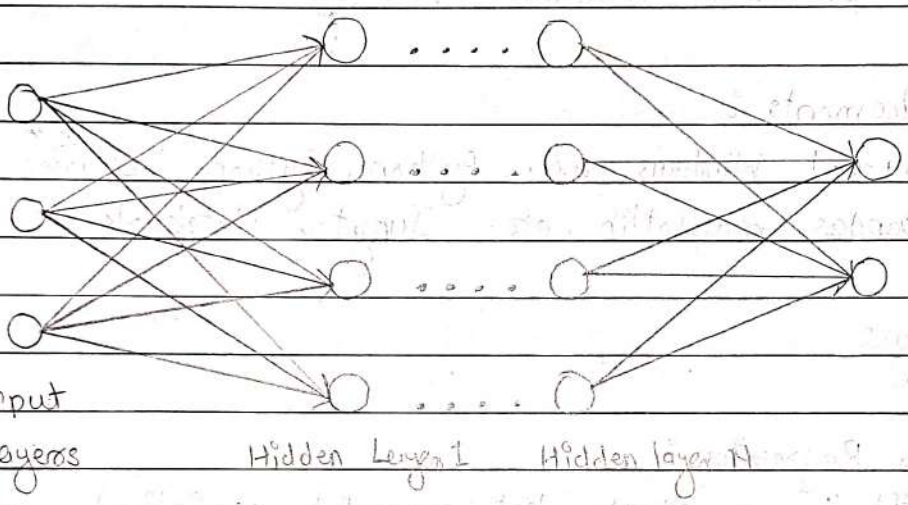

fig. Architecture of ANN

The difference between neural network and deep neural network is that of a chess game, and computer. A chess game is programmed such that it has algorithms to determine tactics, depending on your moves and actions. Whereas, the computer can be able to learn from you and other players, by playing with them and become invincible.

In the context of deep neural networks, linear regression can be defined as a single-layer neural network with one neuron output, which predicts the dependent variable based on input features.

Page No.
Date

A linear regression neural network takes in a vector of input features multiplies each feature by a weight, adds up the weighted inputs, and then passes the result through linear activation function to obtain predicted value of dependent variable mathematically as, $y = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$.
The goal is to find values of the weights & bias term that minimize the difference between predicted values and actual values of dependent variable. This is achieved by using a loss function such as mean squared error (MSE), which measures the average squared difference between them.

Algorithm:

Step 1: Import all python libraries required such as tensorflow, numpy, pandas, matplotlib, seaborn, etc.

Step 2: Load the dataset and split it into training dataset and testing dataset.

Step 3: Conduct exploratory analysis on both training and testing datasets. such as:
① Check data shape & type.
② Converting data to dataframe using pandas library
③ View the datasets
④ Perform preprocessing on the datasets.

Step 4: Create Deep Neural Network, model, Train and Test the created model.

Page No.
Date

Step 5 : Model Evaluation :

Previeu the mean value of training and validation data. ~~data~~ Evaluate the model on the Test data. Plot the loss curves.

Step 6 : View the model predictions.

Mean Squared Error (MSE) : Activation Function —

It measures how does a regression line is to a set of data points. Mean square error is calculated by taking average, specifically mean, of error squared from data as it relates to a function.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

where, n is number of data points
$\hat{y}_i$ is predicted value by model
y_i is actual value of datapoint i.

Conclusion :

Hence, in this assignment we learned how to implement linear regression and deep neural network models to product predict the price of house in Boston using Boston House price prediction dataset.

Page No.
Date

Lab Assignment No. 2

LP-II Deep Learning

Aim :

Classification using Deep Neural Network : Binary classification using Deep Neural Networks : Classify movie reviews into "positive" reviews and "negative" reviews, just based on the text context of the reviews. Use IMDB dataset.

Dataset Description :

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

Requirements :

64-bit Windows O.S., Python, Python libraries : Numpy, scikit learn, keras, Tensorflow, Jupyter Notebook.

Theory :

Binary Classification :

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 review based on their text context.

Page No.
Date

Deep Neural Network are a type of machine learning model that are capable of learning complex patterns of data.

Algorithm :-

Step 1 : Load the dataset using built in function in Keras.

Step 2 : Pre-process the dataset by converting the integer sequences into a binary matrix using one-hot encoding.

Step 3 : Split the training dataset into training and validation sets.

Step 4 : Implement a deep neural network with following architecture -
- An embedding layer to convert the integer sequence into desire vectors of fixed size.
- Two dense layers with RELU activation function.
- A final dense layer with a sigmoid activation function to output probability for "positive" or "negative".

Step 5 : Train the model using the Adam optimizer and binary cross-entrophy loss function.

Page No.
Date

Step 6: Evaluate the model on the test dataset and report the accuracy and loss.

Step 7: Experiment with different hyperparameters such as number of hidden units and learning rate and evaluate the model performance.

Step 8: Save the trained model for future use.

Step 9: END.

Conclusion :

Hence, we have successfully implemented binary classification for IMDB dataset.

Page No.
Date

Lab Assignment No. : 3

LP-II Deep Learning

Aim :

Convolutional Neural Network :

Use MNIST fashion dataset and create a classifier to classify fashing clothing into categories

Requirements :

64-bit Windows O.S., Python, Python libraries : Tensorflow, numpy, pandas, Jupyter Notebook.

Theory :

Convolutional Neural Network (CNN) :

It is a type of a neural network commonly used in Deep Learning for image recognition, classification, and segmentation tasks. The architecture of a CNN is designed to hierarchial of features from input images.

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

Finally, the output of the last pooling layer is passed through one or more fully connected layers, which perform classification or regression based on the learned features.

CNN have been very successful in a wide range of computer vision tasks, including image classification, object detection and semantic segmentation.

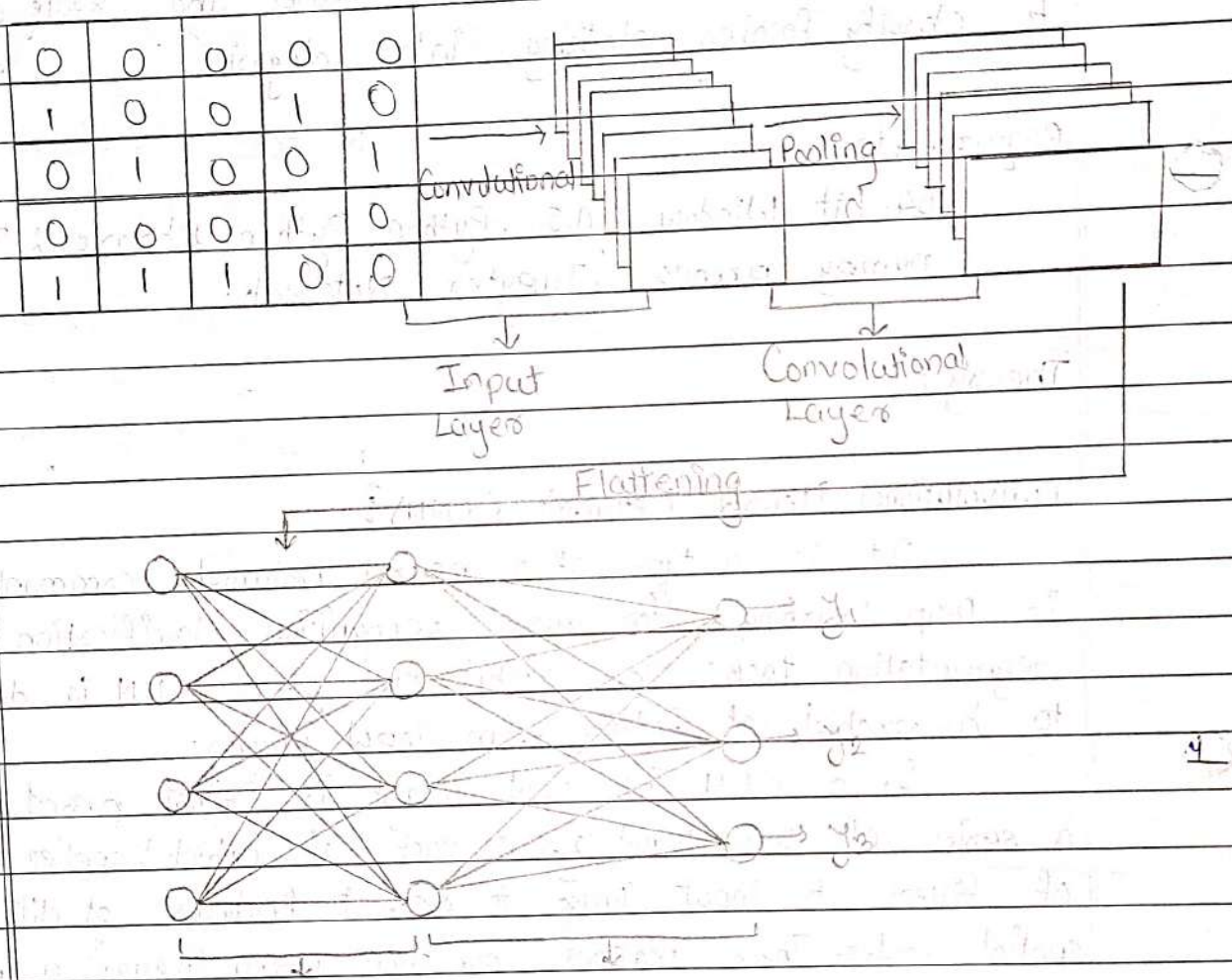

fig: Architecture of CNN

Page No.
Date

Algorithm :

Step 1: START

Step 2: Import all the necessary libraries like tensorflow, pandas, numpy etc.

Step 3: Load the dataset and split it into training and testing dataset.

Step 4: Normalise the data to values between 0 and 1.

Step 5: Explore the data:
① View the shape of the training data.
② View the shape of the testing data.

Step 6: Preprocess the data:
Verify that the data is in correct format and that you are ready to build and train the networks.

Step 7: Build the model:
Building the model for neural network requires configuring the layers of the model, then compiling the model using
model.compile(..)

Step 8: Train the model:
① Feed the training data to the model.

Page No.
Date

② The model learns to associate images & labels.
③ You ask the model to make predictions about a test set.
④ Verify that the predictions match the labels from testing dataset.

Step 9: Feed the model:
To start training, call the model.fit() method - so called because it "fits" the model to the training data.

Step 10: Evaluate accuracy:
Compare how the model performs on test dataset.

Step 11: Make predictions:
With the model trained, you can use it to make predictions about some images.

Step 12: END

Conclusion:
Hence, we prepared a model to create a CNN classifier using MNIST fashion dataset to classify fashion clothing into categories.