

Assignment No. 1

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

Requirements - Python, Python libraries: Tensorflow, pandas, matplotlib etc. Jupyter Notebook.

Theory: Linear Regression -

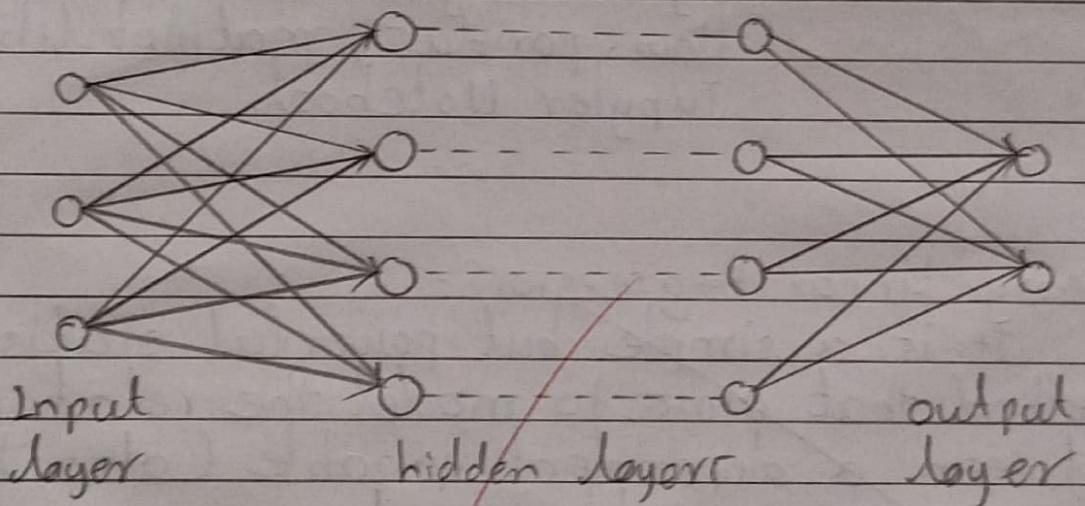
~~It is a simple but powerful statistical method that aims to model the relationship between a dependent variable (also known as response variable) and one or more independent variables.~~

In deep learning, linear regression is used as a basic building block for more complex models and also as a component of more complex models. In neural networks, linear regression can be used as a way to combine input 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 ANNs, Deep Neural Networks can model complete non-linear relationships.

The main purpose of a neural network is to receive a set of inputs, perform progressively complex world problems like classification.



The difference between 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 network, linear regression can be defined as a single layer neural Network with one neuron output, which predicts the dependent variable based on input features.

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 $y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$. The goal is to find values of the weights and bias term that minimize the difference between predicted values and actual values of dependent variables. This is achieved by using a loss function such as mean squared error (MSE) which measures the average squared difference between them.

Algorithm:

- ① Import all python libraries such as tensorflow, numpy, pandas, matplotlib, seaborn etc.
- ② load the dataset and split it into training dataset and testing dataset.

- ③ Conduct exploratory analysis on both training and testing dataset.
- ④ Create Deep Neural Network model, Train and test the created model.
- ⑤ Model Evaluation : Preview the mean value of training and validation data.
- ⑥ Evaluate the model on the test data, plot the loss curve.
- ⑦ View the model predictions.

Mean Squared Error (MSE) :

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

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

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

Advantages of Deep Learning -

- i) Best in class performance on problems
- ii) Reduces need for feature engineering.
- iii) Eliminates unnecessary costs.
- iv) Identifies defects easily that are difficult to detect.

Disadvantages of Deep learning -

- i) Large amount of data required
- ii) Computational expensive to train
- iii) No strong theoretical foundation.

Applications of deep learning -

i) Automated Text Generation:

Corpus of text is learned and from this model new text is generated, word-by-word or character-by-character. Then this model is capable of learning how to spell, punctuate, form sentences, or it may even capture the style.

ii) Image Recognition:

Recognizes and identifies people / objects in images as well as to understand content and context

iii) Healthcare-

Helps in diagnosing various diseases and treating it.

Keras - Dense layer

Dense layer is the regular deeply connected neural network layer. It is most common and frequently used layer. Dense layer does the below operation on the input and return the output.

$$\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias})$$

where,

input represents input data

kernel represents weight data

dot represents numpy dot product of all input and its corresponding weights.

bias represents a biased value used in machine learning to optimize model

Activation represents the activation function

Conclusion:

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

Assignment No. 2

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

Requirements: Python, Python libraries:
TensorFlow, Pandas, numpy,
Jupyter Notebook.

Theory: Classification-

~~Classification~~ is a form of data analysis that extracts models describing data classes. It is a supervised learning technique that is used to identify the category of new observations on the basis of training data. A classifier, predicts categorical classes. Numeric prediction models continuous-value functions. Classification and numeric prediction are the two major types of prediction problems.

Type of classification:

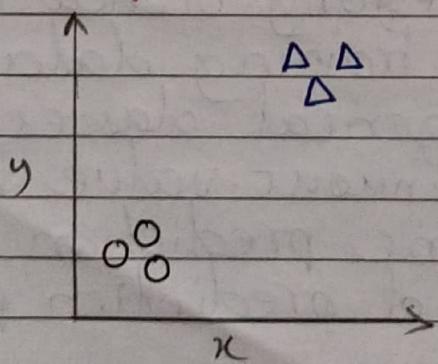
- i) Binary classification
- ii) Multiclass classification

① Binary classification -

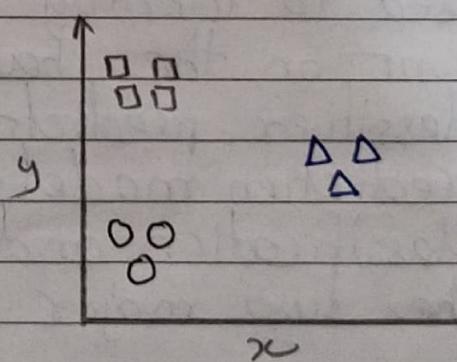
When we have to categorize given data into 2 distinct classes. e.g. - On the basis of given health conditions of a person, we have to determine whether the person has a certain disease or not.

② Multiclass classification :-

The number of classes is more than 2. For example - on the basis of data about different species of flowers, we have to determine which species our observation belongs.



Binary classification



Multiclass classification.

Algorithm:

step 1: Load the dataset using built in function in keras

step 2: Pre-process the dataset by converting the integer sequence into a matrix using one-hot encoding

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

step 4: Implement deep neural Network with following Architecture -

i) An embedding layer to convert the integer sequence into desire vector of fixed size.

ii) One dense layer with RELU Activation function.

step 5: iii) A final dense layer with RELU Activation function to output probability for "positive" or "negative"

step 5: Train the model using the Adam optimizer and binary cross-entropy loss function.

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

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

step 8: solve the trained model for failure

use

step 9: End.

layers used in model:

① Text Vectorization -

It is a pre-processing layer which maps text feature to integer sequences. This layer has basic option for managing text in a hierarchical model. It transforms a batch of strings into either a dict of token indices or a dense representation.

② Conv1D :

This layer creates a convolution kernel that is convolved with the layer input over a single spatial dimension to produce a tensor of outputs.

③ Flatten :

It is used to flatten the input. For example, if flatten is applied to layer having shape ay (batch size, 2×2), then the output shape of the layer will be (batch size, 4)

④ Max Pooling ID -

It downsamples the input representation by taking the maximum value over a spatial window of size pool-size.

⑤ Dense -

Dense layer is the regular deeply connected neural network layer. It is most common and frequently used layer. Dense layer does the below operation on the input and return the output.

output: Activation (dot (input) kernel) + bias)

⑥ LSTM -

LSTM stands for long short memory layer. Based on available runtime hardware and constraints, this layer will choose different implementations.

Conclusion -

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

~~BB~~

Aim: Convolutional Neural Network -

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

Requirements: Python, Python libraries: tensorflow, numpy, pandas, Jupyter Notebook.

Theory: Convolutional Neural Network -

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 hierachical 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 dimensionality 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 /

regression based on the learned features. CNN have been very successfully in a wide range computer vision tasks, including image classification, object detection and semantic segmentation.

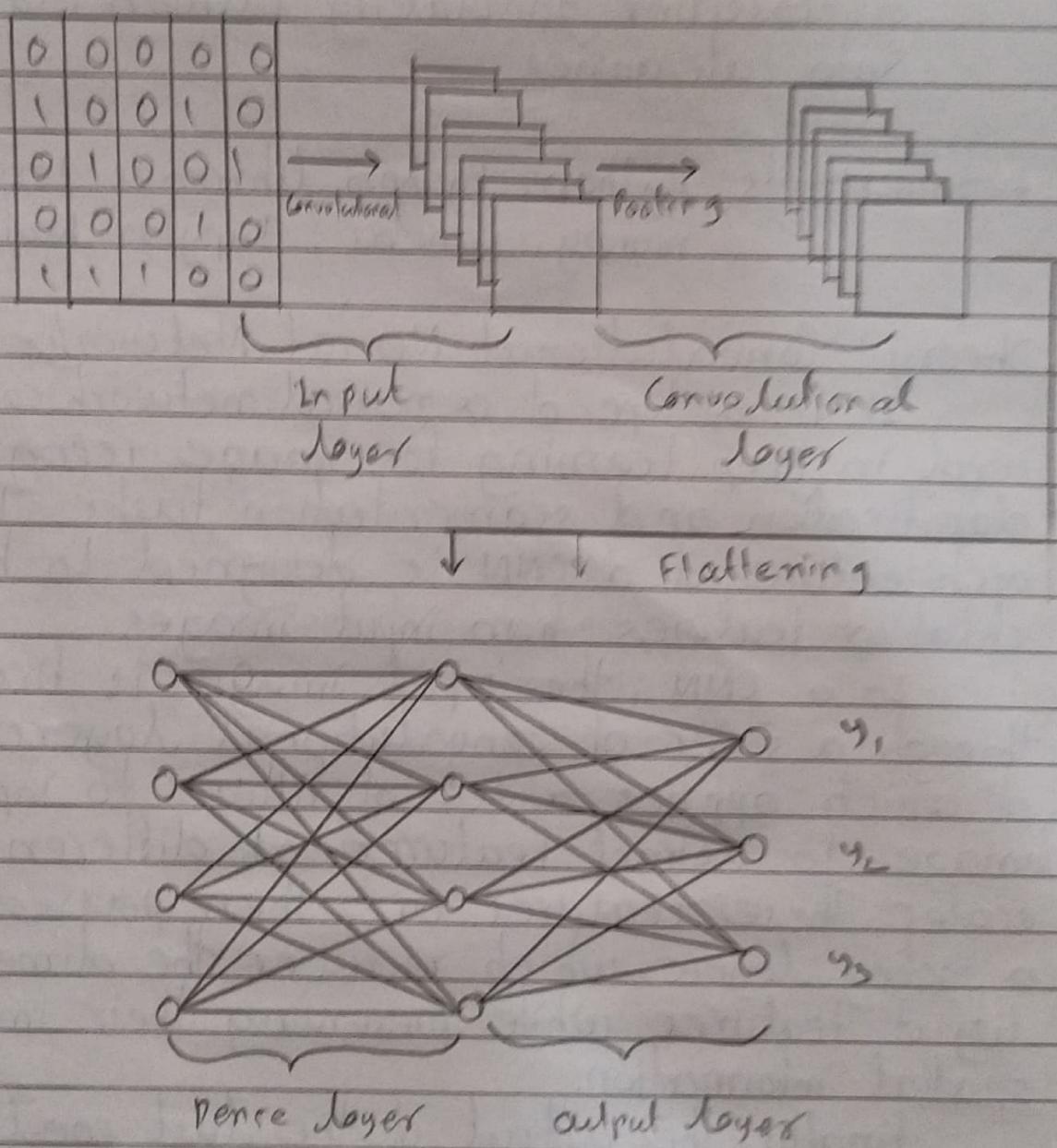


Fig. Architecture of CNN

① Convolutional layer -

This layer is the first layer that is used to extract various features from the input. In this layer, the mathematical operation of convolutional is performed between the input image and a filter of a particular size. It is by sliding the filter over the input image.

② Pooling layer -

The pooling layer follows the convolutional layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational cost. This is performed by decreasing the connections between layers and it is independently operated on each feature map.

③ Fully connected layer -

The fully connected layer consists of the weights and biases along with the neurons and is used to connect the neurons between the two different layers.

④ Activation function -

Activation functions are used to learn and approximate any kind of continuous and complex relationship between variables of network.

Algorithm -

step 1: start

step 2: Import all necessary libraries like tensorflow, pandas, numpy etc

step 3: load the dataset and split it into training and testing dataset

step 4: Normalize the data to values between 0 and 1.

step 5: Explore the data.

i) View the shape of the training data.

ii) View the shape of the testing data.

step 6: process the data:

Verify that the data is in correct format and that you are ready to build and train the model.

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:

i) Feed the training data to the model.

ii) The model learns to associate images and labels.

iii) You ask the model to make predictions about a test set.

iv) Verify that the predictions match the labels from testing dataset.

step 9: Feed the model:

To feed the start training, call `model.fit()` method, so called because it 'fits' the model to 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 images.

step 12: End.

Conclusion:

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

Aim: Recurrent Neural Network (RNN) :
use the google stock prices dataset and
design a time analysis and prediction
system using RNN.

Requirements: Jupyter Notebook, Python libraries:
tensorflow, pandas, matplotlib, etc.

Theory: Recurrent Neural Network:

Recurrent Neural Network (RNN) is a
type of neural Network where the output
from the current step.

In traditional neural network, all the
inputs and outputs are independent of each
other, but in cases like when it is required
to predict the next word of sentence,
the previous words are required and hence
there is a need to remember the previous
words.

The main and most important feature
of RNN is hidden state, which remembers some
information about a sequence.

output

Hidden
layer

Input

RNN have a 'memory' which remembers all information about what has been calculated

It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output.

Thus reduce the complexity of parameters, unlike other neural networks.

RNN Working.

The working of RNN can be understood with help of below example:

Suppose there is a deeper network with one input layer, three hidden layers and one output layer. Then like other neural networks each hidden layer will have its own set of weights and biases. Let's say for hidden layer 1 the weights and bias are (w_1, b_1) , for second hidden layer (w_2, b_2) and (w_3, b_3) for third hidden layers. This means that each of these layers is independent of other i.e. they do not memorize the previous outputs.

Conclusion:

Hence, we have performed recurrent neural network on google price stock database.