House Price Prediction using Backpropagation

I. Neural Network - Backpropagation

The foundation of neural network training is backpropagation. It is a technique for adjusting a neural network's weights depending on the error rate (cost function) recorded in the previous epoch (i.e., iteration). By properly tweaking the weights, you may lower error rates and improve the model's reliability by broadening its applicability.

The term "backward propagation of errors" is shortened to "backpropagation" in neural networks. It is a common technique for developing artificial neural networks. With regard to each weight in the network, this technique aids in calculating the gradient of a loss function.

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \nabla_{\mathbf{w}} J(\mathbf{w})$$

Here,

-> Cost function calculations, C (w) is done first.

-> Gradient descent of C(w) with regard to all of your neural network's weights w, and biases, b. (NN)

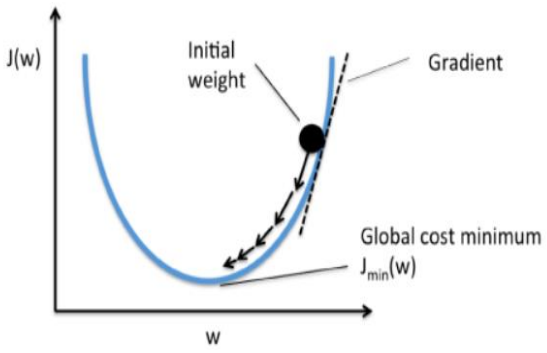
$$\text{Gradient} = \frac{\partial C}{\partial w}$$

->In proportion to the magnitude of their slopes, adjust the w and b.

The goal is to minimize C(w) cost function of our model.

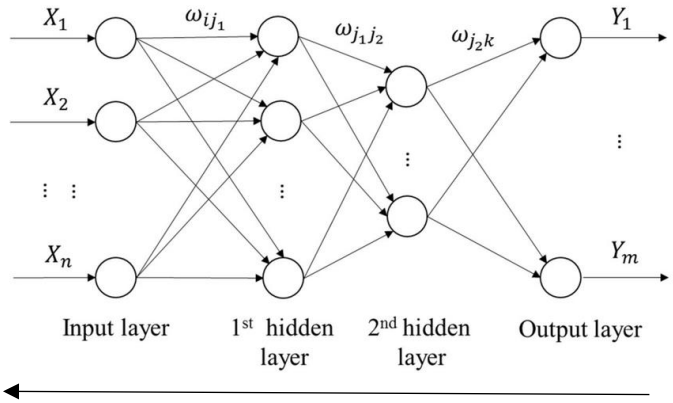
New weight = old weight – learning rate* gradient descent.

Gradient Descent-



The backpropagation approach generates the gradient and avoids repeating the computation of intermediate terms in the chain rule by using the chain rule, to determine the gradient descent of the loss function with respect to each weight and iterating backward from the final layer.

In the modern sense, a neural network is an artificial neural network made up of artificial neurons. A neural network may therefore be either a biological neural network consisting of biological neurons, or an artificial neural network intended to address artificial intelligence (AI) issues. Artificial neural networks mimic biological neuron connections as weights between nodes. An excitatory link is represented by a positive weight, whereas an inhibitory connection is represented by a negative weight. Each input is given a weight before being added together. A linear combination is the term used to describe this action. Finally, an activation function regulates the output's amplitude.

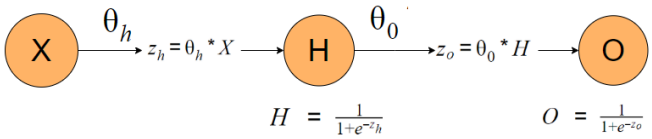


BACK PROPAGATION

A. Mathematical Understanding of GMM:

The 3 equations of back propagation are:

Output Layer Error	$E_o = (O - y) \cdot R'(Z_o)$
Hidden Layer Error	$E_h = E_o \cdot W_o \cdot R'(Z_h)$
Cost-Weights Deriv	$LayerError \cdot LayerInput$



Here X is the input layer, H is the hidden layer and O is the output layer.

II. Experiment Motivation

Consider a real estate company that has a dataset containing the prices of properties in the Delhi region. It wishes to use the data to optimise the sale prices of the properties based on important factors such as area, bedrooms, parking, etc.

III. METHODOLOGY

The current paper focuses on using a special machine learning model called Neural Network - Backpropagation in addition to extracting significant non-correlated features from the dataset. The model will proceed according to the standard procedures, which include data extraction followed by exploratory data analysis, data preprocessing, model construction, and model parameter evaluation.

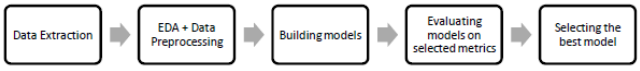


Fig 1. Flowchart describing Methodology

A. Dataset:

I used a dataset from Kaggle which contains 545 observations and 13 columns. The columns are:

i. Price – Price of houses, integer

ii. Area – Area of the house per sqare feet, integer

iii. bedroom – Number of bedrooms in the house, integer

iv. Bathrooms – Total bathrooms, integer

v. Stories – Number of stories, integer

vi. Main-road – If the house is near mainroad, object

vii. Guestrooms – Presence of the guestroom, object

viii. Basement – Presence of the basement, object

ix. hotwaterheating – presence of the steamer is available, object

x. Parking – No of Parking lots alloted, integer

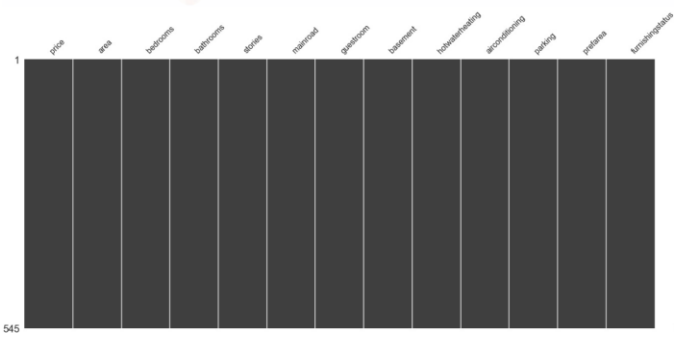
xi. furniture – Displays if the house is semi-furnished and Furnished, object

xii. Prefarea – Preference of the buyers, Object

B. Exploratory Data Analysis and data preprocessing:

The data has around 545 rows and 13 columns. The view of the subset of the dataset is given below:

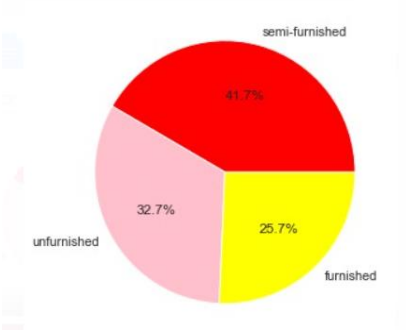
	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished



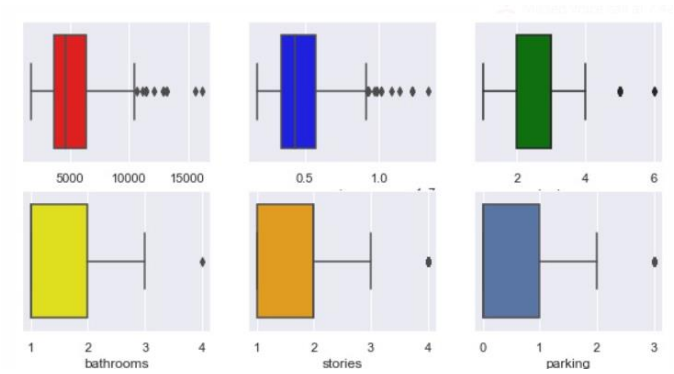
The shows a distribution of null values in the data frame. As you can see there are no null values.

	price	area	bedrooms	bathrooms	stories	parking
count	545.000000	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4766729.247706	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1870439.615657	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1750000.000000	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3430000.000000	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4340000.000000	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5740000.000000	6360.000000	3.000000	2.000000	2.000000	1.000000
max	13300000.000000	16200.000000	6.000000	4.000000	4.000000	3.000000

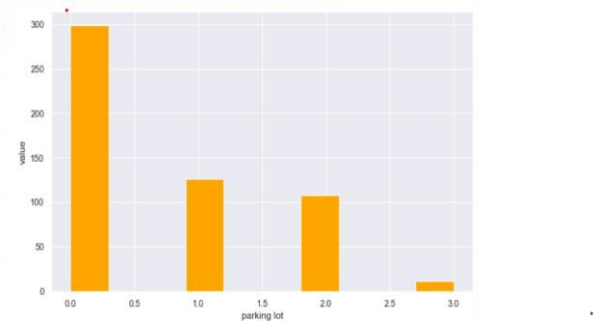
The table above shows the statistical description of the entire data-frame.



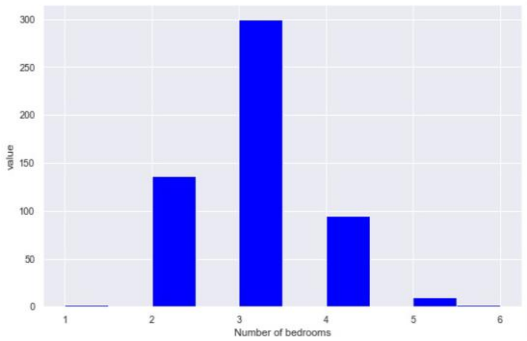
This pie chart shows the condition in which the houses are marketed. For example, 41.7 % of the houses marketed are semi-furnished.



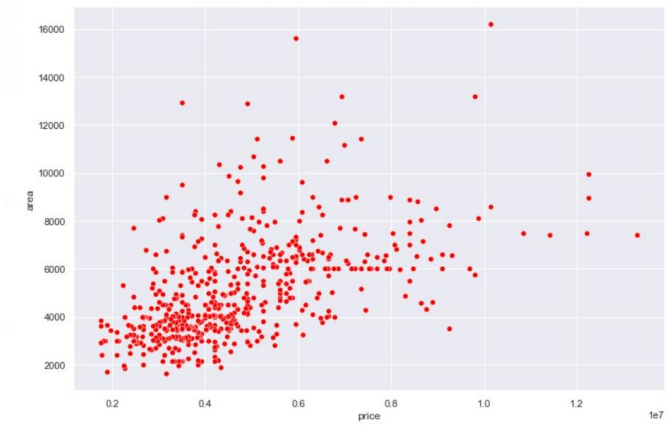
The boxplot shows the distribution of the variables in the data frame which included median values of bathroom, stories, parking, price area and bedrooms



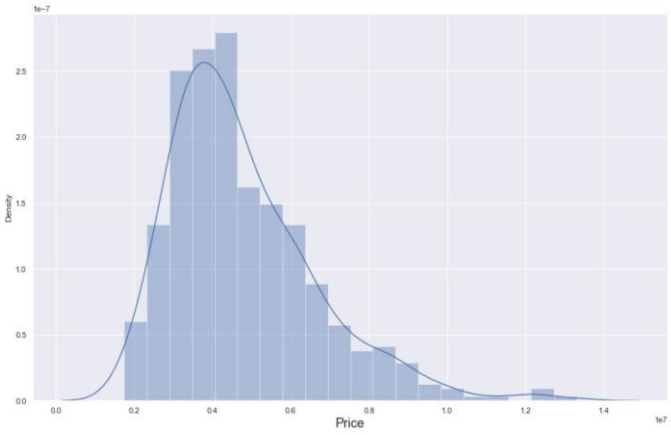
This graph shows the relation between the number of houses and the number of parking available with that household.



This graph shows the distribution of houses with respect to number of bedrooms. As you can see this feature is normally distributed.



The graph shows how prices and area are related. As you can see high area lean towards high price and vice versa.



The graph above is a distplot of price column in the dataframe.

Data preprocessing steps:

i. Performed correlation matrix and removed all non-correlated values.

ii. Changed the names of certain columns.

iii. Label encoding of all the categorical variables to be used in modeling.

IV. Model Building

I discovered the model's parameters and trained it for back propagation using the sklearn general library. I tried the suggested technique, but it did not help to lessen the mistake:

```
model = Sequential([
    tf.keras.layers.InputLayer(input_shape = (6,)),
    Dense(units = 40, activation = 'linear'),
    Dense(units = 1, activation = 'linear')
])
```

```
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.5)
    loss=tf.keras.losses.MeanSquaredError()
)
```

I used 40 layers to model this experiment with epoch ranging 10 to 25 finding the optimum epoch.

A. Evaluation Metrics:

i. R-squared value:

The adjusted completeness(model accuracy) statistic of the multilinear model indicates the proportion of the variation in the target field that can be ascribed to the input or inputs:

$$Adjusted\ R^2 = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$$

R^2 = The coefficient of determination
N = total sample size
p = no. of independent variables

ii. Root mean squared error (RMSE):

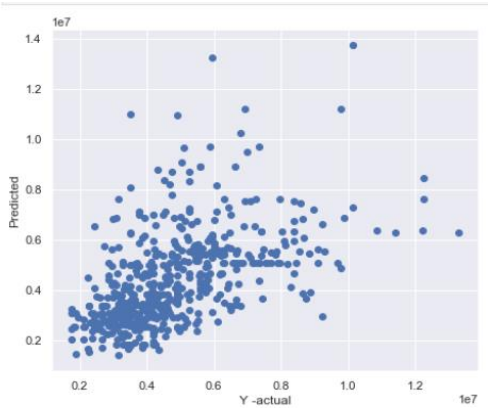
In statistics, the square root of the mean square, or the arithmetic mean of the squares of a set of values, is known as the root mean square (RMS). RMS is a specific example of the generalized mean, whose exponent is 2, and is sometimes known as a quadratic mean.

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

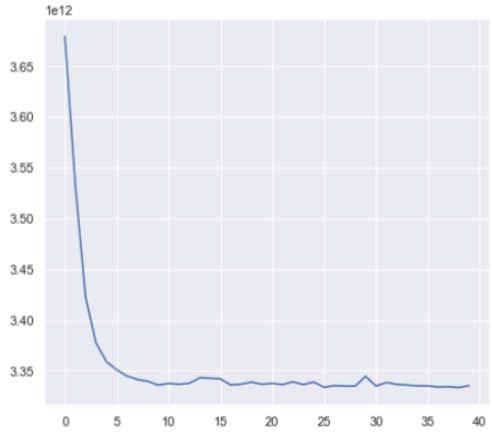
n is the number of non-missing observations
y_i is an actual observation
$\hat{y}_i$ is predicted observation

V. Evaluation Results

The below diagram shows how the predicted and the actual values are related:



As you can see the loss function, there is an abrupt change in loss.



I changed the learning rate to get a better RMSE and R-squared value. Below are the results of my experiments:

Learning rate	Error rate	Accuracy
0.01	0.54	62%
0.02	0.33	74%
0.1	0.21	78%
0.2	0.2087	78.5%
0.5	0.205	79.9%

The RMSE value and the R-squared values of this model is close to perfect:

RMSE	500
R-squared	80%

As you can see that the R-squared value is close to 100% and the rmse value is near zero even the graph of actual vs prediction formed a straight line stating that the values predicted are close to perfect.

VI. Conclusion

To conclude we can say that the neural network backpropagation performed good to predict the price of house. I split the training and testing data and performed

backpropagation using 40 layers and using the learning rate of 0.5.

VII. Acknowledgement

I would like to express my special gratitude to professor Changyou Chen, for teaching the underlying concepts behind every machine learning model. I would also like to thank the Teaching Assistants for clearing doubts and providing support when needed.

VIII. REFERENCES

i. Changyou Chen, CSE 574 Lecture Slides(2022).
ii. https://www.kaggle.com/datasets/ashydv/housing-dataset
iii. https://en.wikipedia.org/wiki/Backpropagation