House Price Prediction
(3 and 5 layer Neural Networks)

Arun Kumar
Machine Learning Intern
AI Tech Systems (www.ai-techsystems.com)
arunksharma1998@gmail.com

***Abstract*— while buying something everyone wants to pay as low as possible. Same is the case while buying a house. House price depends upon a lot of factors so it's hard to guess the price of a house by analyzing lot of factors. In this project I used a Neural Network model to predict house prices based on a dataset. 3 and 5 layers neural networks are built and later on their performance is compared. Comparing factor used is mean absolute percentage error.**

Keywords— Machine Learning, Regression, Deep Learning, Atrificial Neural Networks

I. INTRODUCTION

Why? Why we need to predict the house prices? This is the first question that arises in our mind. What makes predicting house prices an area of interest.

The answer is simple house or broadly speaking "Real Estate" is a field where we can invest. Since investing is related with money so it becomes important to value the property as close as possible so as to minimize the loss and therefore maximize the profit that an investor can make from buying that piece of property. Demand in general is one the the key factor in increase or decrease of price of property. In this article I have analysed house price dataset from a website called kaggle. Data is analysed to identify the important factors contributing to the price of a property using different data analysis techniques and then I built two powerful neural networks one with 3 layers and other with 5 layers to train on this dataset and later on predict the prices of some unpredicted houses. At last I compared the percentage error produced by both neural networks to compare their performance.

II. METHODOLOGY

I have used python3 as the primary language to code the neural networks used to predict the prices of houses. Different libraries were used like pandas, numpy, seaborn, matplotlib, keras, scikit-learn. Uses of these libraries in building the model.

NumPy and Pandas are used to create data structures so as to store data in nd-arrays and dataframes and data manipulations were done using these libraries.

Matplotlib and Seaborn are used to plot the data to analyse the data in graphical form. Heat maps, Histograms etc. were plotted to get deep insights of data. Plotting also helped in detecting outliners present in different features of the correlated data.

Scikit-learn is used to pre-process the data before putting it into the neural network for processing.

Keras is a powerful library which is used to build the neural network.

A. ***Dataset***

The dataset used to train the neural network is The Ames Housing dataset was compiled by Dean De Cock for use in data science education. It's an incredible alternative for data scientists looking for a modernized and expanded version of the often cited Boston Housing dataset.

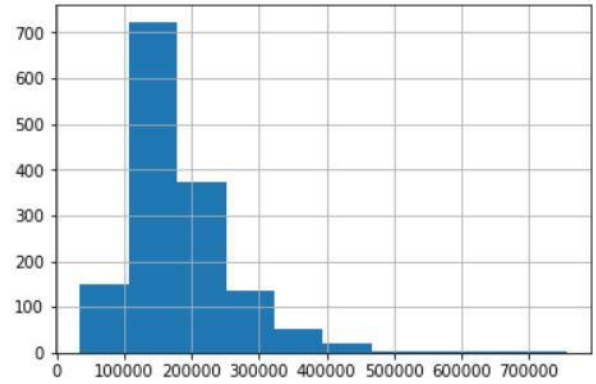
The dataset has 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa. The dataset contains 1460 data points on which training can be done and 1440 data points on which we can test our neural network. This dataset contains both numerical and categorical data. Some of the data columns are year built, garage area, living room area, street, overall quality, sale condition, pool area and in total there are 79 data columns. There are 36 numerical columns and 43 categorical columns. The dependent variable is "Sale Price" which is present in training set data on which we can train our neural network.

B. *Data Analysis*

Data analysis is the process of evalutating data using analytical and statistical tools to discover useful information form the dataset.

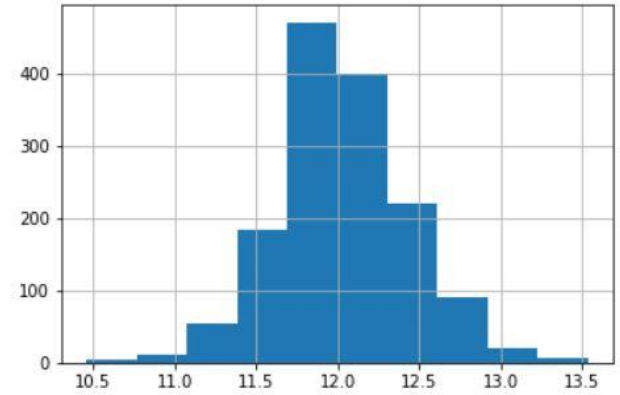
At first histogram of the dependent variable SalePrice was plotted and skewness was calculated.

```
Skew is : 1.8828757597682129
```



Then SalePrice was visualized in order to find its distribution using log transformation and skew is calculated.

```
Skew is : 0.12133506220520406
```



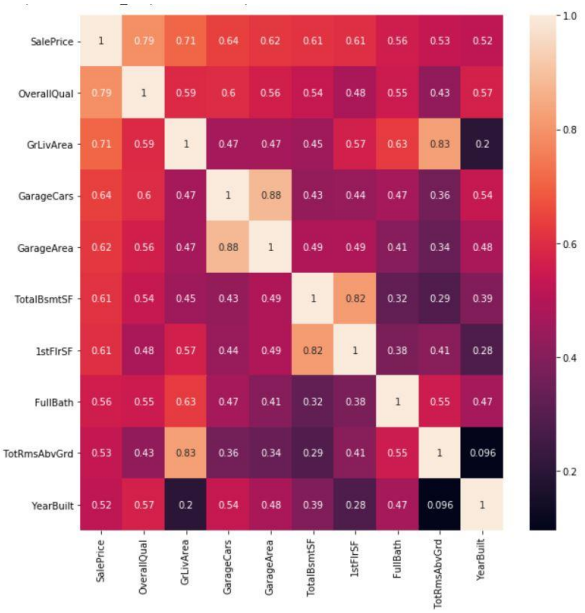
We got skew as 0.12 which is good and we can see great normal distribution.

Plotting Heatmap

To find those features which are most correlated with our dependent variable SalePrice we plotted a heatmap using seaborn library to plot 10 most correlated features.

After plotting the heatmap we found that the features listed below we most correlated with target variable with these correlation coefficients.

OverallQual	0.790982
GrLivArea	0.708624
GarageCars	0.640409
GarageArea	0.623431
TotalBsmtSF	0.613581
1stFlrSF	0.605852
FullBath	0.560664
TotRmsAbvGrd	0.533723
YearBuilt	0.522897
YearRemodAdd	0.507101



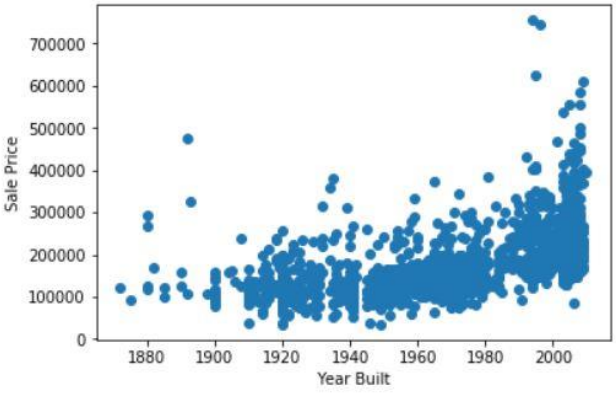
Here we can see 'OverallQual' is most correlated feature to SalePrice with correlation coefficient 0.79

Investigation of top most correlated features with target variable SalePrice.

1. SalePrice vs OverallQual

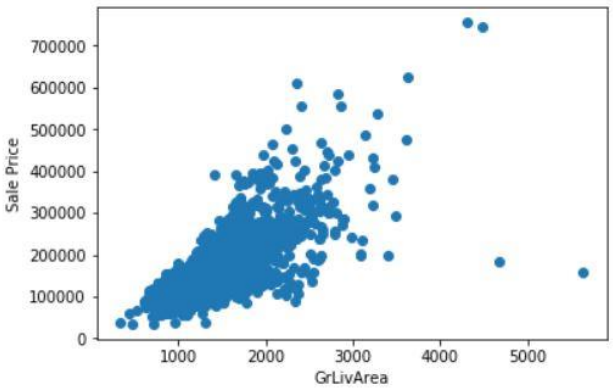
Here we can see higher overall quality leads to higher sale price of property.

2. SalePrice vs YearBuilt

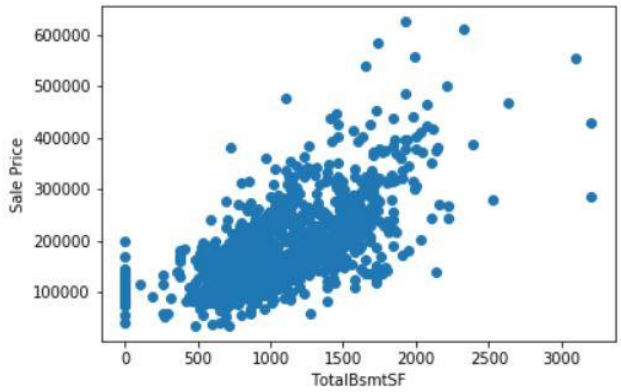


We observed that older houses have lower sale price.

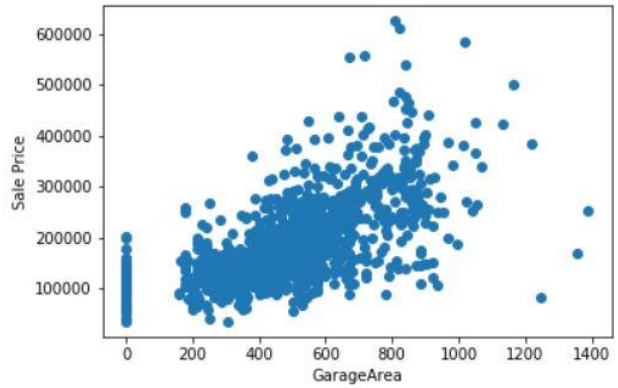
3. SalePrice vs GrLivArea



4. SalePrice vs TotalBasementArea



5. SalePrice vs GarageArea



In these observations we can clearly see some outliners. We removed these outliners so that they can't affect performance of our neural network.

C. *Data Preprocessing*

Data preprocessing is important because in real world data, we may come across the situations where some of the data is missing and training our model with such data will leave huge negative impact on the performance of our model. So, to handle this situation in our case we replaced numerical missing values with mean of the remaining values in that column using basic imputation methods from scikit-learn library.

We know that machine works on numbers. So, it becomes necessary to convert categorical data to numbers and for this we used dummy variables technique.

Some the features(LotArea, Saleprice etc.) in this dataset had large values. So, to reduce execution time we scaled the values using MinMaxScaler form scikit-learn. After doing all the data preprocessing steps we got 287 columns which will be used in our neural network as input variables in input layer.

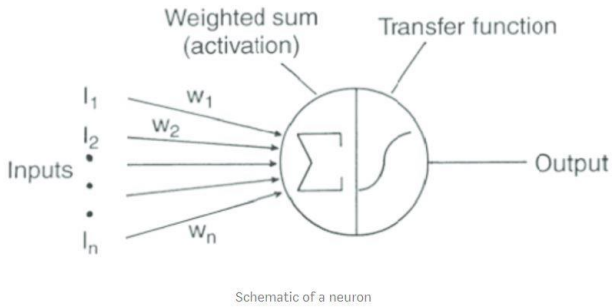
D. *Neural Network Model*

Neural network is a powerful deep learning model which consists of input layer, hidden layers and output layer.

Input Layer consists of independent variables which contributes in predicting dependent variable. Here we have 287 input variables in our case.

Hidden Layer consists of neurons which uses backpropagation to optimize the weights of synapses in order to improver the predictive power of neural network.

Output Layer consists of nodes which contains the output we got after processing data from input and hidden layers.



Schematic of a neuron

1. 3 layers Neural Network

For building neural network we used keras library which uses tensorflow and theanos in backend.

The three layers are:-

Input Layer : 287 input variables

Hidden Layer : 128 neurons with activation function as 'relu'which is rectifier function.

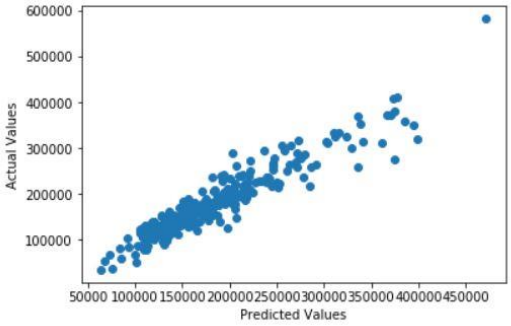
Output Layer : 1 node with activation function as 'linear'. The node in this layer will contain predicted price of house by our neural network.

One dropout layer with 10% dropout was added to reduce overfitting. Dropout is a technique where random neuron are ignored while training.

For compilation part we used ‘adam’ as optimizer function which is extension to stochastic gradient descent and ‘mean_squared_error’ as loss function.

Then this model is fitted to training data with batch_size of 20 and 20 epochs. With these parameters our model produced mean_absolute_precentage error of 7.5457%.

Plot of actual values and predicted values.



2. 5 layers Neural Network

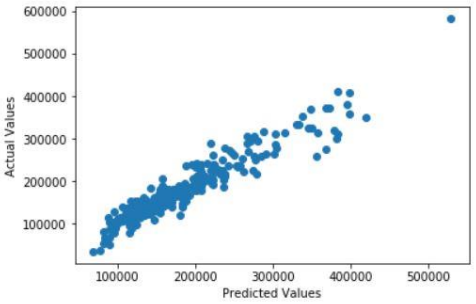
The five layers are:-
Input Layer : 287 input variables
Hidden Layer1 : 128 neurons with activation function as ‘relu’ which is rectifier function.
Hidden Layer2 : 128 neurons with activation function as ‘relu’ which is rectifier function.
Hidden Layer3 : 128 neurons with activation function as ‘relu’ which is rectifier function.
Output Layer : 1 node with activation function as ‘linear’. The node in this layer will contain predicted price of house by our neural network.

Three dropout layes with 5% dropout were added to reduce overfitting.

For compilation part we used ‘adam’ as optimizer function which is extension to stochastic gradient descent and ‘mean_squared_error’ as loss function.

Then this model is fitted to training data with batch_size of 20 and 20 epochs. With these parameters our model produced mean_absolute_precentage error of 6.9489%.

Plot of actual values and predicted values.



III. CONCLUSION

The purpose of this project was to build two neural networks one with 3 and other with 5 layers and predict house prices. All the necessary step were followed before building the neural network like handling missing values, encoding categorical variables, scaling the data and splitting the data to train and test set. After training both neural networks and evaluating their performance we got the results as 5 layers neural network outperformed 3 layers neural network. The mean_absolute_percentage_error of 3 layers neural network was 7.5457 % and that of 5 layers neural network was 6.9489 %.

IV. REFERENCES

[1]Alejandro Baldominos , Iván Blanco ,Antonio José Moreno , Rubén Iturrarte , Óscar Bernárdez and Carlos Afonso, “Identifying Real Estate Oppurtunities Using Machine Learning, November 2018

[2]Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, wayan Firdaus Mahmudy, “Modeling House Price Prediction using Regression Analysis and Particle Swarm Optimization”, Vol. 8, No. 10, 2017

--x--