House Price Prediction using 3-Layer and 5-Layer Neural Networks

Nayakam Vishnuvardhan
Machine Learning Intern
AI Tech Systems
Hyderabad, India
vishnu123sai@gmail.com

www.ai-techsystems.com

***Abstract*— In the real time one of the costlier things are lands and houses. The relationship between house prices and the economy is an important motivating factor for predicting house prices. Housing price trends are not only the concern of buyers and sellers, but it also indicates the current economic situation. The price of the houses varies based on lot of parameters like land area in sq. feet's, location of house or land, based on availability of water and transportation and based on availability of facilities. Higher the facilities result always higher cost per sq. feet. This is one of the Machine learning Technique to use to predict the price of the house under some parameters. In This technique we have used 3 Layer and 5 Layer neural networks to compute the price of the House.**

Keywords*— Machine Learning, Deep Learning, Neural Networks, House Price Prediction System, Artificial Intelligence, Prediction.

I. Introduction

The relationship between house prices and the economy is an important motivating factor for predicting house prices .There is no accurate measure of house prices. A property's value is important in real estate transactions. House prices trends are not only the concerns for buyers and sellers, but they also indicate the current economic situations. Therefore, it is important to predict the house prices without bias to help both buyers and sellers make their decisions. There are different machine learning algorithms to predict the house prices. This project will use Neural network to solve this problem. I am going to use two types Neural networks to compare and get the best solution one is with 3 layers of neural network and another one is 5 layers of neural network.

There are many factors affect house prices, such as numbers of bedrooms and bathrooms. In addition, choosing different combinations of parameters in Support Vector Regression will also affect the predictions greatly. This project is guided by these questions: Which features are important for predicting price of houses? How to select those features in the data to achieve a better performance? How to fine tune the parameters? The flow chart of the problem solving is given in II.

II. Flow of the Problem Solving

The Structure of the problem solving is as below diagram.

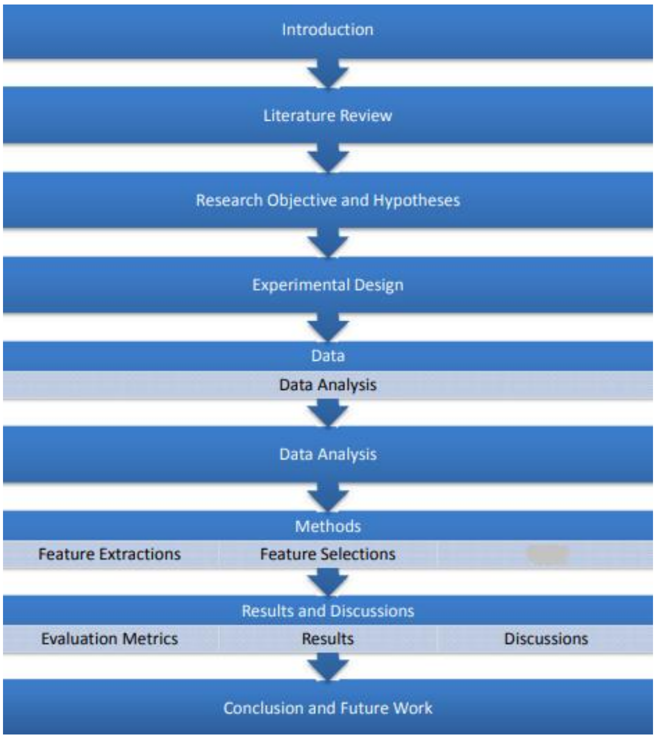


Fig (i)

We will more concentrate on the main steps involved to solve the problem.

A. *Dataset or Data*

Data is, to put it simply, another word for information. But in the context of computing and business, data refers to information that is machine-readable as opposed to human-readable. The dataset should have all of the useful features stand out. For example, if we were to make a model whose job is to detect where the person is on an image, our dataset should consist of images which contain people for who we know their exact location on the image.

After successfully collecting data, it should be converted into a format that our model understands. In general, the input data, whether it’s texts, images, videos, or sounds, is turned into vectors and tensors to which linear algebra operations can be applied. The data needs to be normalized, standardized and cleaned to increase its effectiveness. Until then, the data is called Raw data (unprocessed data).

B. *Data Analysis and Feature selection*

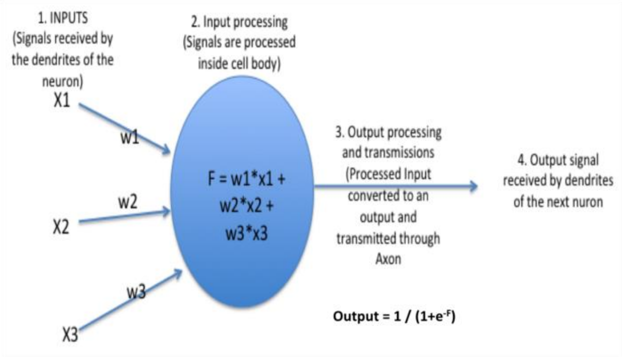
Feature extraction involves reducing the number of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power, also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction.

Data analysis is a process of inspecting, cleansing, transforming and modelling data with the goal of discovering useful information, informing conclusions and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively

C. *Neural Networks*

Artificial Neural Network (ANN) uses the processing of the brain as a basis to develop algorithms that can be used to model complex patterns and prediction problems.

In our brain, there are billions of cells called neurons, which processes information in the form of electric signals. External information/stimuli is received by the dendrites of the neuron, processed in the neuron cell body, converted to an output and passed through the Axon to the next neuron. The next neuron can choose to either accept it or reject it depending on the strength of the signal.


1. INPUTS (Signals received by the dendrites of the neuron)
2. Input processing (Signals are processed inside cell body)
X1
w1
X2
w2
X3
w3
F = w1*x1 + w2*x2 + w3*x3
3. Output processing and transmissions (Processed Input converted to an output and transmitted through Axon
4. Output signal received by dendrites of the next nuron
Output = 1 / (1+e^-F)

Fig(ii)

Here, w1, w2, w3 gives the strength of the input signals

As you can see from the above, an ANN is a very simplistic representation of a how a brain neuron work.

The network architecture has an input layer, hidden layer (there can be more than 1) and the output layer. It is also called MLP (Multi Layer Perceptron) because of the multiple layers. The hidden layer can be seen as a “distillation layer” that distills some of the important patterns from the inputs and passes it onto the next layer to see. It makes the network faster and efficient by identifying only the important information from the inputs leaving out the redundant information. The activation function serves two notable purposes:

1. It captures non-linear relationship between the inputs
2. It helps convert the input into a more useful output

D. *Model Selection and Methods*

Model selection is the process of combining data and prior information to select among a group of statistical models “Mγ” belongs to “M”. In building a model, decisions to include or exclude covariates as well as uncertainty in how to code the covariates in the design matrix “Xγ” for any given model “Mγ” are based both on the prior hypotheses and the data. With many potential covariates, these decisions become difficult. Some algorithms will select variables to be included in the model, but only return a single “best” model.

III. METHODOLOGY

In This Part we will look at the data set, pre-processing steps, building neural network, making a model, testing model and predicting predictions.

A. *Dataset*

For this problem we are using data set of Kaggle problem “**House Prices: Advanced Regression Techniques**”. Which consists 80 different features.

File descriptions:

- train.csv - the training set
- test.csv - the test set
- data_description.txt
- sample_submission.csv

Data fields :

Here's a brief version of what you'll find in the data description file.

- MSSubClass: The building class
- MSZoning: The general zoning classification
- LotFrontage: Linear feet of street connected to property
- LotArea: Lot size in square feet
- Street: Type of road access
- Alley: Type of alley access
- LotShape: General shape of property
- LandContour: Flatness of the property
- Utilities: Type of utilities available
- LotConfig: Lot configuration
- LandSlope: Slope of property
- Neighborhood: Physical locations within Ames city limits etc..

B. *Data Analysis and Feature Selection*

Data Analysis and Feature selection plays major role in getting correct or nearer predictions. Best way to get the related features which exactly effects price of the house is correlation function.

```
corr_matrix =train_set.corr()

corr_matrix['SalePrice'].sort_values(
ascending=False)
```

```
output:

 SalePrice        1.000000

 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

 GarageYrBlt      0.486362

 MasVnrArea       0.477493

 Fireplaces       0.466929

 BsmtFinSF1       0.386420

 LotFrontage      0.351799

 WoodDeckSF       0.324413

 2ndFlrSF         0.319334

 OpenPorchSF      0.315856

 HalfBath         0.284108

 LotArea          0.263843

 BsmtFullBath     0.227122

 BsmtUnfSF        0.214479

 BedroomAbvGr     0.168213

 ScreenPorch      0.111447

 PoolArea         0.092404
```

```
MoSold           0.046432

3SsnPorch        0.044584

BsmtFinSF2      -0.011378

BsmtHalfBath    -0.016844

MiscVal         -0.021190

Id              -0.021917

LowQualFinSF    -0.025606

YrSold          -0.028923

OverallCond     -0.077856

MSSubClass      -0.084284

EnclosedPorch   -0.128578

Name: SalePrice, dtype: float64
```

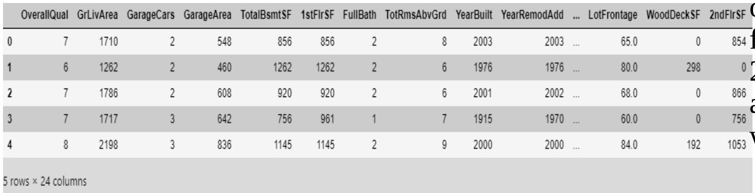
Choose the features which correlates more with price of the house and make a model using those features so that we may get the effective predictions. Select the features which correlation rate more than 0.1.

```
needed_columns = ["OverallQual",'GrLivArea','GarageCars','GarageArea','TotalBsmtSF','1stFlrSF','FullBath','TotRmsAbvGrd','YearBuilt','YearRemodAdd', 'GarageYrBlt','MasVnrArea','Fireplaces','BsmtFinSF1','LotFrontage','WoodDeckSF','2ndFlrSF','OpenPorchSF','HalfBath','LotArea','BsmtFullBath','BsmtUnfSF','BedroomAbvGr','ScreenPorch'];

X = train_set[needed_columns];

y = train_set['SalePrice'];

X.head();
```



	OverallQual	GrLivArea	GarageCars	GarageArea	TotalBsmtSF	1stFlrSF	FullBath	TotRmsAbvGrd	YearBuilt	YearRemodAdd	...	LotFrontage	WoodDeckSF	2ndFlrSF
0	7	1710	2	548	856	856	2	8	2003	2003	...	65.0	0	854
1	6	1262	2	460	1262	1262	2	6	1976	1976	...	80.0	298	0
2	7	1786	2	608	920	920	2	6	2001	2002	...	68.0	0	866
3	7	1717	3	642	756	961	1	7	1915	1970	...	60.0	0	756
4	8	2198	3	836	1145	1145	2	9	2000	2000	...	84.0	192	1053

5 rows × 24 columns

Handling Nan values in dataset and scaling done as follows. Nan values are filled with median of the feature and scaling features for the best results using MinMaxScaler scaling.

1. Fillinf Nan Values:

```
median = int(X['MasVnrArea'].median())

X['MasVnrArea'].fillna(int(median),inplace=True)

median = X['LotFrontage'].median()

X['LotFrontage'].fillna(int(median), inplace=True)

median = X['GarageYrBlt'].median()

X['GarageYrBlt'].fillna(int(median), inplace=True)
```

2. Feature Scaling :

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X_t = pd.DataFrame(scaler.fit_transform(X));

X_te = pd.DataFrame(scaler.fit_transform(test_X))
```

The above process also done for test set.

C. *Making Neural Network*

In this project we built two neural networks to solve the problem one is 3 layer neural network and another is 5 layer neural networks using Keras.

1. 3 Layer Neural Network :

Three layer Neural network consists of one input layer, one hidden layer and one output layer. Input layer has 100 neurons, relu functions as activation and input_dimension of 24, the hidden layer has 50 neurons with activation relu and output layer has one neuron with linear activation.

three_layer_model = Sequential()

three_layer_model.add(Dense(100, activation='relu',input_dim=X.shape[1:][0]))

three_layer_model.add(Dense(50,activation ='relu'))

three_layer_model.add(Dense(output_dim=1 ,activation='linear'))

three_layer_model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])

Model is compiled with loss function "mean squared error" and optimizer "adam".

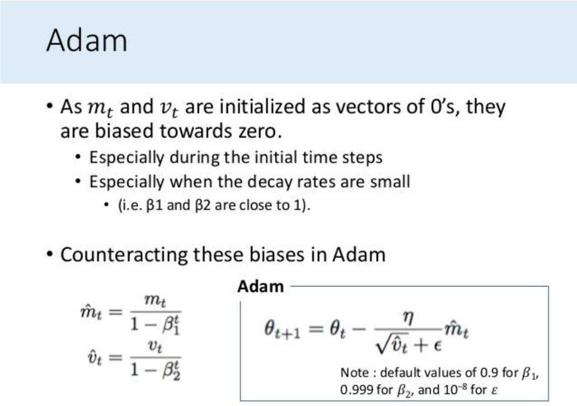
$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$*\ n$ is the number of data points

$*\ Y_i$ represents observed values

$*\ \hat{Y}_i$ represents predicted values

Fig(iii)



Adam

- As m_t and v_t are initialized as vectors of 0's, they are biased towards zero.
  - Especially during the initial time steps
  - Especially when the decay rates are small
    - (i.e. β1 and β2 are close to 1).
- Counteracting these biases in Adam

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Adam

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t$$

Note : default values of 0.9 for β_1, 0.999 for β_2, and 10^{-8} for ε

Fig(iv)

Predictions with 3 layer neural network arer as follows

Predictions :

```
[[0.20121609]
 [0.23529924]
 [0.2953675 ]
 [0.29329547]
 [0.25579336]]
```

Actual values :

```
0.224208
0.248687
0.243157
0.237507
0.199642
```

Model Info:

```
Batch size: 32
No. of Epochs: 30
Loss function : MSE
Optimizer : Adam
```

2. 5 Layer Neural Network

` Five layer Neural network consists of one input layer, three hidden layer and one output layer. Input layer has 300 neurons, relu functions as activation and input_dimension of 24, the hidden layers has 100,50 and 10 neurons respectivly with activation relu and output layer has one neuron with linear activation.

five_layer_model.add(Dense(300, activation='relu',input_dim=X.shape[1:][0]))

five_layer_model.add(Dense(100,activation ='relu'))

five_layer_model.add(Dense(50,activation=' relu'))

five_layer_model.add(Dense(10,activation=' relu'))

five_layer_model.add(Dense(output_dim=1, activation='linear'))

five_layer_model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])

Model is compiled with loss function "mean squared error" and optimizer "adam".

Predictions :

```
[[0.18765602]
 [0.21876758]
 [0.27828076]
 [0.28479898]
 [0.25641984]]
```

Actual values :

```
0.224208
 0.248687
0.243157
0.237507
0.199642
```

Model Info :

```
Batch size: 32
No. of Epochs: 30
Loss function : MSE
Optimizer : Adam
```

IV. Conclusion

In this project we tried to predict the house price based on some features using 3 layer neural network and 5 layer neural network. Using 24 Features which correlates the price of the house are considered for this project. 5 Layer neural network gives more accurate result compare to 3 Layer neural network.

References

[1] Hands on Machine learning wih scikit-learn, keras and tensorflow, II Edition, Aurelien Geron.

[2] Housing Price Prediction, Paper by An Nguyen, March 20, 2018

[3] Housing Price prediction Using Support Vector Regression, Paper by Jiao Yang Wu, San Jose State University.

[4] Modeling House Price Prediction using Regression Analysis and Particle Swarm Optimization, Paper by Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, Brawijaya University

..