Analysing the performance of neural networks over same data set

Biswajit Jena

Email - biswajitbits@gmail.com

AITS Link-https://www.ai-techsystems.com

AI Tech Systems
Goa,India

***Abstract*—This project was an assignment given by AI Tech Systems during its internship. The objective of the project was to make two neural networks with different neurons and then compare both of their predictions.**

Keywords—neural networks, keras, accuracy, mean squared error, validation set, test set

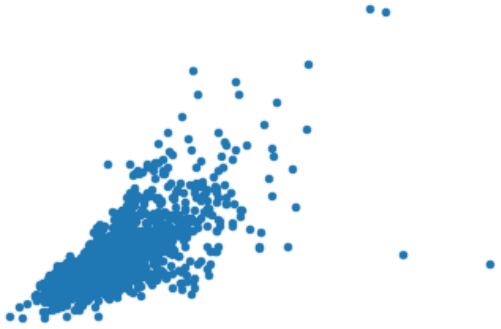
I. Introduction

The neural networks were prepared on the dataset given in a competition of Kaggle named House Prices: Advanced Regression Techniques. The first neural network comprises of three hidden layers and the second one contains five hidden layers. Both the codes were written on keras.

II. DATASET VISUALISATON AND ANALYSIS

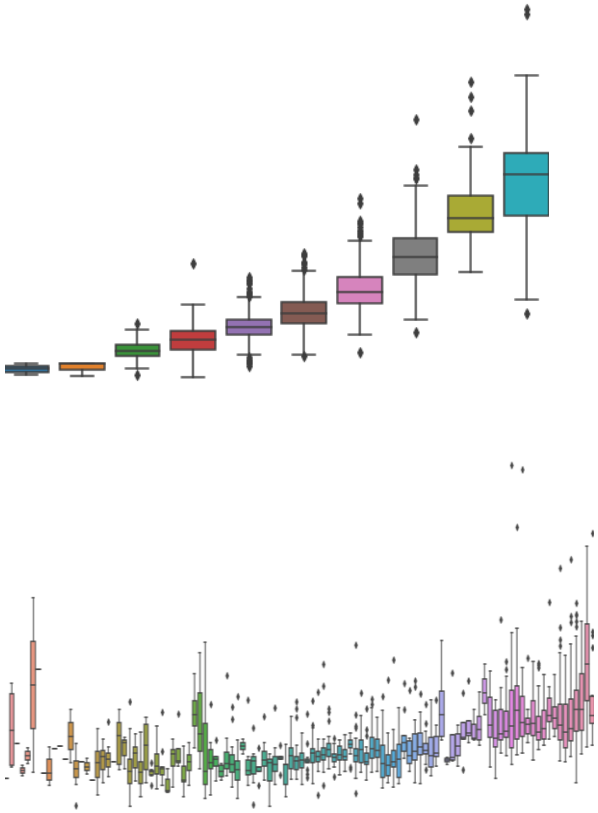
The salesprice show:-
1)Deviate from the normal distribution.
2)Have appreciable positive skewness.
3)Show peakedness.

After analyzing the dataset, following are relationship between variables and salesprice:-
1)SalesPrice and GRlivarea seem to be in a linear relationship.



2)SalesPrice and TotalBMSFarea seem to be in a linear-exponential relationship.

OverallQual and YearBuilt also seem to be related with 'SalePrice'. The relationship seems to be stronger in the case of OverallQual, where the box plot shows how sales prices increase with the overall quality.



XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

III. Removing.Outliers

I used IsolationForest class from skicit learn library for removing the outliers from dataset.
Number of Outliers: 146
Number of rows without outliers: 1314

IV.1ST Neural network

A. *Structure of first neural network*

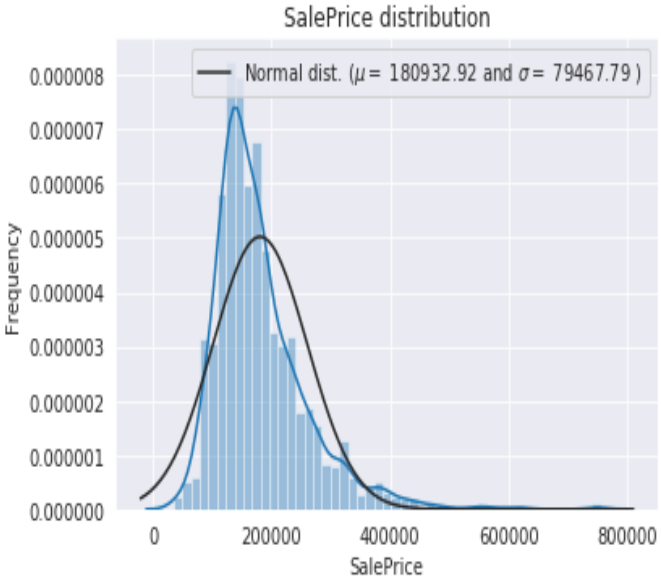
Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 128)	4736
dense_2 (Dense)	(None, 256)	33024
dense_3 (Dense)	(None, 128)	32896
dense_4 (Dense)	(None, 1)	129

Total params: 70,785

Trainable params: 70,785

Non-trainable params: 0

The three hidden layers consisted of 128,256,128 neurons respectively.



The target variable is right skewed. As (linear) models love normally distributed data , we needed to transform this variable and make it more normally distributed.

We first pre-processed data by converting all sales price by applying MinMaxScaler with the help of sklearn library and then converting back from model predictions.

B. *Performance of first neural network*

Epoch 50/50 - 0s - loss: 0.0058 - mean_squared_error: 0.0058 - val_loss: 0.0095 - val_mean_squared_error: 0.0095

We can see that the neural network ran for 50 epochs with batch size of 100. The validation loss was approximately equal to training loss signifying that the model was neither overfitting nor underfitting.
This was achieved by adding dropout layers between the hidden layers.
The dying relu problem was solved by using LeakyRelu as an activation function.
Also the loss was very much less as the sales price value is of order 100000.

V 2ND NEURAL NETWORK

A. *Structure of 2nd Neural Network*

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 128)	4736

dense_2 (Dense) (None, 256) 33024

_____

dense_3 (Dense) (None, 512) 131584

_____

dense_4 (Dense) (None, 256) 131328

_____

dense_5 (Dense) (None, 128) 32896

_____

dense_6 (Dense) (None, 1) 129

==========

Total params: 333,697

Trainable params: 333,697

Non-trainable params: 0

_____

The three hidden layers consisted of 128,256,512,256,128 neurons respectively.

We first pre-processed data by converting all sales price by applying MinMaxScaler with the help of sklearn library and then converting back from model predictions.

B. Performance of second neural network

Epoch 50/50 - 0s - loss: 0.0023 - mean_squared_error: 0.0023 - val_loss: 00048 - val_mean_squared_error: 0.0048

We can see that the neural network ran for 50 epochs with batch size of 100. The validation loss was approximately equal to training loss signifying that the model was neither overfitting nor underfitting.
This was achieved by adding dropout layers between the hidden layers.
The dying relu problem was solved by using LeakyRelu as an activation function.

VI. ANALYSIS OF BOTH NEURAL NETWORKS

Model 2 performed better that model 1 as it consisted of more number of layers and more number of trainable parameters. Neither of the models overfitted the dataset given as can be seen by validation and training losses.

Model 2 scored 0.13 on Kaggle submission whereas model 1 scored 0.145 on Kaggle showing that model 2 performs way better than model 1.

Acknowledgment

I would like to that's AI Tech systems to provide me such an opportunity to work on project under them and their constant guidance.

REFERENCES

https://www.kaggle.com/c/house-prices-advanced-regression-techniques