Predicting House Prices with basic Informtion and Images of the House[1]

Hesam Damghanian
SBU, Computer Science Dept.
h.damghanian@mail.sbu.ac.ir

***Abstract*— One way to implement regression in neural networks is to use only one neuron in the last layer so that output of that neuron gets closer to the target value in each iteration. With this given dataset, a simple way to predict house prices only based on the structured data which rather straightforward. Another way to do so is to create a ConvNet model along having mentioned mlp and concatenating output of these two layers. As conducted experiments suggests, simple mlp regression model performs better than the combined model.**

Keywords—regression, CNN, structured data, mixed data

I. Dataset

The given dataset consists of a 15474*8 table and 15474 images each corresponding to a house. Columns of the table are *image_id (int64), street (string), city (string), n_city (int64), bed (int64), bath (float64), sqft (int64)* and *price (int64).* Image_id is the key to related image for the house, n_city is simply the id for each city, bath column can be divided into to separate columns, full_bath and half_bath. Columns did not have null entry and there was no duplicated entry in the entire table.

II. Preprocessing

Outliers lying in the sqft and the price column were normalized with IQR method. Bath column has been divided into two new columns as mentioned and price and sqft column had been scaled to (0, 1). City and n_city columns also could be bucketized but since the size of each bucket would be small, I decided not to, same logic applies for street column (12401 unique streets!). so city, n_city, street, and obviously image_id columns were dropped and resulting table now has only 5 columns.

Fig. 1

Fig. 1. Shows a sample house image from training dataset.

Preprocessing house images

Given house images were not all demonstrating the same view of the house and some of the even were locations on the map or ‘no image’ pictures which is usually displayed in real state websites when there is no uploaded image. So, we should not expect seeing the accuracy of the prediction being improved with these pictures. However, for augmenting the images I used an imageGenerator from keras libraries to augment (shear, translate, rotate etc.) the images.

III. Model Architechture

Figure 2 depicts the overall implemented structure which consists of a cnn model for fitting images and a mlp model to fit the given table. The output layer of these two models should be of the same size to concatenate them. After the concatenation layer there is a fully connected layer for predicting the price (regression).

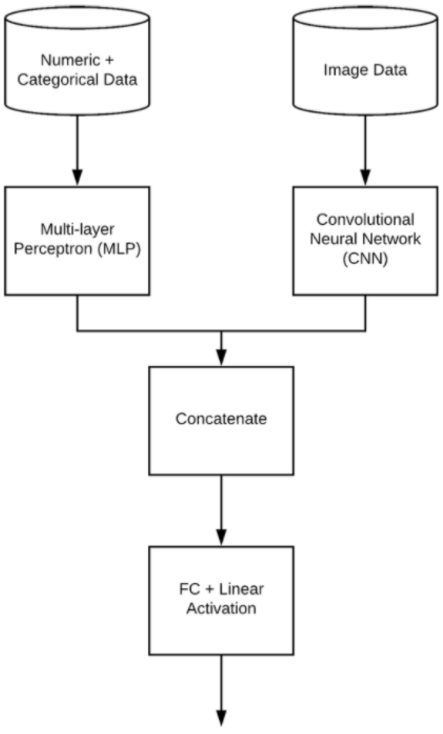
Numeric + Categorical Data
Image Data
Multi-layer Perceptron (MLP)
Convolutional Neural Network (CNN)
Concatenate
FC + Linear Activation

Fig. 2

Specific details of these two models are provided in attached notebook.

IV. Results

Chosen loss function is mean absolute percentage error and can be calculated by:

1. Technical notes are in attached notebook

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Eq. 1

Where M is mean absolute percentage error, *n* is number of times the summation iteration happens, A_t is actual value and F_t is forecast value.

The results are in the notebook but they are incredibly low and inaccurate.

Another mlp regressor model were conducted which is also in the attached notebook, which suggests that images didn’t result in any improvement in the model accuracy.

(predicted values of the last model were in 2 decimal precision which I think is the reason that in the last dataframe, the *diff* column was all zeros)