This is the project on prediction of HDB resale price based on past resale prices. The resale price are adjusted by RPI to standardized the data. It is adjusted based on 2024,4th quarter RPI. Then data is cleaned, preprocessed .

Before Preprocessing,it is 180154 rows x 11 columns .

After Preprocessing, input shape is : 175672x51 .

The original model consists of input layer, 2 hidden layers and an output layer.

2nd layer made of 32 neurons ,and its total parameters with bias are : 32x51+32=1664 .

3rd layer made of 16 neurons , and its total parameters with bias are : 32x16+16=528

Output layer is a single neuron as it is regression problem : total parameter : 16x1+1=17

Model: "sequential_6"

Layer (type)	Output Shape	Param #
dense_23 (Dense)	(None, 32)	1,664
dense_24 (Dense)	(None, 16)	528
dense_25 (Dense)	(None, 1)	17

```
                                Train      Test
Root Mean Squared Error       85004.52  84948.47
Mean Aboslute Error           63869.84  63767.65
Mean Aboslute Percentage Error    10.63     10.63
R2 score                           0.82      0.82
total training time :28.773163318634033 seconds
```

Epochs and batch_size stay the same across models to test the performance of layers and neuron difference.

Variety of layers and neurons were tested .

	ModelA_2layers	ModelB_ 2layers	ModelC_3layers	ModelD_5layers	ModelE_unorder Random_5layers	Model F_ 5layers small Neuron counts	Model G_ 6layers	ModelH_10Layers
Training Time	27.760	27.699	29.095	35.060	31.799	38.215	36.149	1015.808
Root Mean Squared Error (Train)	77738.989	65573.440	62114.599	57249.736	57301.822	56973.513	58463.045	55269.725
Mean Absolute Error (Train)	57049.074	47695.177	44906.774	41248.695	41800.522	40873.389	42875.342	40198.110
Mean Absolute Percentage Error(Train)	9.315	7.766	7.314	6.703	6.873	6.619	7.044	6.587
R2 Score(Train)	0.852	0.895	0.905	0.920	0.920	0.920	0.916	0.925
Root Meean Squared Error(Test)	77708.188	65411.323	61946.604	57290.897	57404.134	57049.426	58865.929	55737.892
Mean absolute Error(Train)	57019.103	47613.837	44839.218	41326.716	41898.467	40938.628	43184.569	40623.636
Mean Absolute Percentage error(Test)	9.331	7.768	7.318	6.733	6.906	6.644	7.104	6.672
R2 Score(Test)	0.852	0.895	0.906	0.920	0.919	0.920	0.915	0.924
hidden layers	[128, 32]	[256, 128]	[128, 64, 32]	[256, 128, 64, 32, 16]	[32, 153, 12, 352, 12]	[70, 60, 50, 40, 30]	[512, 256, 128, 64, 32, 16]	[8172, 4086, 2048, 1024, 512, 256, 128, 64, 32, 16]
Total Params	10817	46337	17025	57089	17386	14251	201729	44988079

Deeper layers make training time longer and use more computing resources. In my case , training shallow models were fine but in deeper model , warnings like “ Registers are spilled to local memory “ were raised .

Same number of Layers with grater neuron quantity have better metrics. Eg. initial layers and Model B , C). In bigger layers like Model D,E and F , changes in neuron counts doesn’t seem to make big difference. Memory warnings started to appear in Model G and H .It might be the reason why model G have lower score. In Model F , the accuracy is still improving .

In conclusion , Deep layers and greater neurons number really improved the models’ accuracies and their metrics as the network can learn more complex and hierarchical features from the data. Therefore Deep learning is a viable solution for regression if there is enough computing resources.