

CS583A: Course Project

Siyuan He

May 19, 2019

1 Summary

I participate an active competition of predict house prices based on house information data. The information contains both text and numerical data. The final model I choose is a deep convolutional neural network architecture, which contains 9 dense layer with batch normalization and dropout layer. It takes 1×299 one-dimension arrays as input and the output is the prices of houses. I implement the convolutional neural network using Keras and run the code on a Dell xps15 with one 7th Gen Intel i7 CPU and 32 GB memory. Performance is evaluated on the mean squared error. In the public leaderboard, My score is 0.11639; I rank 633 among the 4581 teams.

2 Problem Description

Problem. The problem is to predict house prices based on house information data including Lot Area, Year Built, Room Style etc. This is a regression problem. The competition url is <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview>.

Data. The data are 1×299 one-dimension arrays. The number of training samples is $n = 1460$. The price of each house is around 200.000.

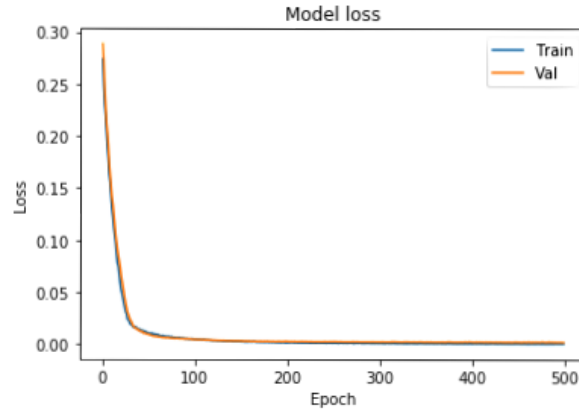
Challenges. The training set is too small. The prediction target, price of the house, is too big.

3 Solution

Model. The model we finally choose is a deep convolutional neural network. It contains 9 dense layer with batch normalization and dropout layer.

Implementation. I use the CNN sequential model provided by Keras. My code is available at https://github.com/Siyuanhee/CS583Final_House_Prices. I run the code on a Dell xps15 with one 7th Gen Intel i7 CPU and 32 GB memory. It takes 6 min to train the model.

Settings. The loss function is mean square error. The optimizer is RMSprop. The learning rate is 0.0001. The epochs is 500. The batch size is 128.



(a) The loss on the training set and validation set.

Figure 1: The convergence curves.

Advanced tricks. I have used Dropout layer, Batch Normalization and make the CNN model deeper. This help my model to decrease the mean square error from 0.0174801 to 0.00041818.

Cross-validation. I use train test split in sklearn to partition the training data to 80%-20% for hyperparameter tuning. Figure 1 plots the the convergence curves on 80% training data and 20% validation data. When the epoch increase, the loss of both train and validation are decrease. After about 200 epochs, the loss of train and validation don't change obviously.




















4 Compared Methods

Least Square Fitting. Least Square Fitting is the base line of my project. It is a basic linear regression model. In this model, the setting of loss is mean square error and the parameter of cross-validation is 5. Its mean square error 0.0174801 is the base mean square error of my project. The score is 0.31561 in public leaderboard.

Lasso Regression. Lasso Regression is a regression model. Here is a description of Lasso online: [https://en.wikipedia.org/wiki/Lasso_\(statistics\)](https://en.wikipedia.org/wiki/Lasso_(statistics)). In this model, the setting of loss is mean square error, parameter of cross-validation is 5 and alpha is 0.0005. Its mean square error is 0.0126106. The score is 0.20878 in public leaderboard.

Ridge Regression. Ridge Regression(Tikhonov regularization) is a regression model. Here is a description of Ridge Regression online: https://en.wikipedia.org/wiki/Tikhonov_regularization. In this model, the setting of loss is mean square error, parameter of cross-validation is 5 and alpha is 12. Its mean square error is 0.0126809. The score is 0.11792 in public leaderboard.

Random Forest. Random Forest is an ensemble learning method. Here is a description of Random Forest online: https://en.wikipedia.org/wiki/Tikhonov_regularization. The score

Overview	Data	Kernels	Discussion	Leaderboard	Rules	Team	My Submissions	Submit Predictions
620	victorwp0904		0.11624	1	10d			
621	ZhuYuFei		0.11625	21	14d			
622	Ichern		0.11626	2	10d			
623	Aleksey Petrov		0.11627	4	16d			
624	olleuler		0.11629	16	12d			
625	Radhika T S		0.11631	6	2mo			
626	AT081179		0.11631	10	2mo			
627	GilSapir		0.11631	3	5d			
628	gelhart		0.11635	1	1mo			
629	Olivia Li		0.11635	6	2d			
630	Wesley2018		0.11636	26	1mo			
631	Dimitri K.		0.11638	9	2d			
632	Siyuan		0.11639	11	now			
Your Best Entry 								
Your submission scored 0.11658, which is not an improvement of your best score. Keep trying!								
633	asv325		0.11642	3	2mo			
634	Brett WangZihan		0.11642	2	2mo			
635	sgtbaur		0.11646	1	17d			
636	Asabur		0.11647	8	2mo			
637	Nazarko		0.11647	17	2mo			

(a) Public leaderboard.

Figure 2: Our rankings in the leaderboard.

is 0.15471 in public leaderboard.

5 Outcome

I participated in an active competition. My score is 0.11639 in the public leaderboard. I rank 632/4576 in the public leaderboard. The screenshots are in Figure 2.

[1]

References

- [1] Rio de Janeiro. Regression:top 20% with a very simple model-lasso. <https://www.kaggle.com/goldens/regression-top-20-with-a-very-simple-model-lasso>, 2016.