CS583A: Course Project

Sattvik Sahai

December 1, 2019

1 Summary

I participated in an active competition of predicting the energy usage of buildings using historical usage data as well as some supporting weather data and information about the building. The final model used was LightGBM trained for 1000 rounds with 500 leaves. The implementation was done using the Python api for the LightGBM framework by Microsoft `https://github.com/microsoft/LightGBM`. Training was done on a laptop with one Intel i7 processor with 16 GB of memory and one NVIDIA GeForce GTX 1070 Max-Q GPU and Google Colab. Performance is evaluated on the Root Mean Squared Logarithmic Error. In the public leaderboard, my score is 0.98; I rank 285 among the 2846 teams. The result on the private leaderboard is not available until December 19 2019.

2 Problem Description

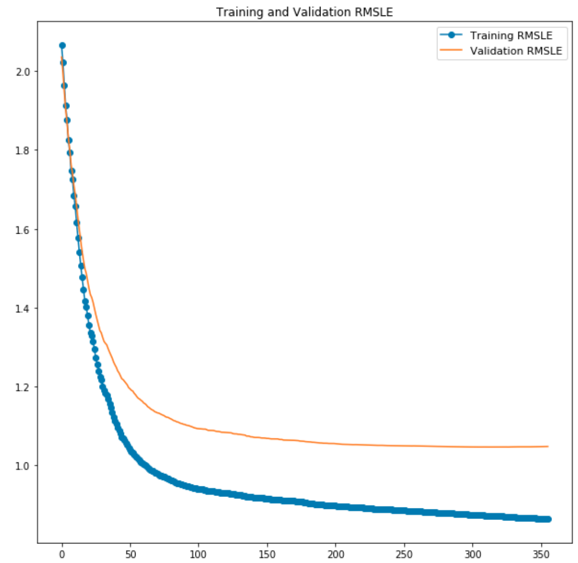
Problem. The problem is to predict the energy usage of buildings using historical usage data as well as some supporting weather data and information about the building. This is a regression problem. The competition is at `https://www.kaggle.com/c/ashrae-energy-prediction`.

Data. The data is a mix of categorical and numerical features and contains the following information: building id, meter type, timestamp, site id, building id, property type, square feet area, built year, floor count, air temperature, cloud coverage, dew temperature, precipitation depth, sea level pressure in the area, wind direction, wind speed. The number of training samples is $n = 20216101$.

Challenges. The data has a lot off missing/erroneous values.

3 Solution

Model. The model finally chosen is the LightGBM [1], a fast gradient boosting decision tree. A description of LightGBM is online: `https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc`.



(a) The Root Mean Squared Error on the training set and validation set.

Figure 1: The convergence curves.

Implementation. The lightgbm model was implemented using the LightGBM framework by Microsoft. My code is available at `https://github.com/sattviksahai/Deep-Learning-Project/blob/master/lightgbm.html`. The code was run on an Asus Zephyrus laptop with one Intel i7 processor with 16 GB of memory and one NVIDIA GeForce GTX 1070 Max-Q GPU and on Google Colab.

Settings. The model was trained for a 1000 iterations with a learning rate of 0.05. The number of leaves were restricted to 500 and 85% of features were used for every iteration. The maximum depth was set to 10 and maximum bins to 9. L2 regularization was added with λ =3.5.

Cross-validation. A 3-fold cross-validation scheme was used to tune hyper-parameters. Figure 1 plots the the convergence curves of one of the three folds. The final Validation error is higher than the Training error. This indicates that the model in its current form is overfitting the training data and has scope to generalize better.

4 Compared Methods

Sequence to Sequence Network with LSTMs. A seq2seq LSTM was evaluated which predicts the meter reading given the current values of the categorical data points, the past 200 values of all numerical data values, and the past 200 values of the meter reading. Figure 2 depicts the architecture of this model. This model achieved an average training RMSLE of 0.17 and a validation RMSLE of 0.40. However it received a score of 4.6 on the Public LB. As the model was trained with true previous values of meter readings, it probably could not generate reasonable predictions

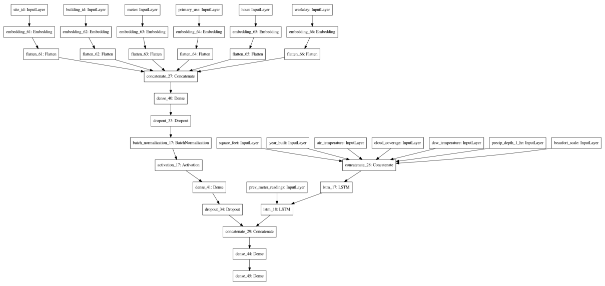


Figure 2: Sequence to Sequence Model.

on test data where its own previous predictions were used as input. The code is available at `https://github.com/sattviksahai/Deep-Learning-Project/blob/master/Keras%20LSTM.html`

Fully-connected neural network. A 7-layer fully-connected neural network was evaluated for this task. The widths of the layers (from bottom to top) are respectively 512, 256, 128, 64, 32, 16, and 1. Dropout and batch normalization were applied for each layer. The training and validation RMSLE are respective 1.0694 and 1.0301. This model received a score of 2.84 on the Public LB. The code is available at `https://github.com/sattviksahai/Deep-Learning-Project/blob/master/dense_NeuralNet_with_embeddings.html`

Advanced tricks. The finally adopted method is a LightGBM with 500 leaves and L2 regularization.

- Feature Engineering and Scaling. Explicit feature added to mark holidays, as it is likely to heavily influence energy usage. Scaled *square feet area* and *primary usage* by using log scaling and min-max scaling respectively. This is done in order to improve the conditioning of the problem.

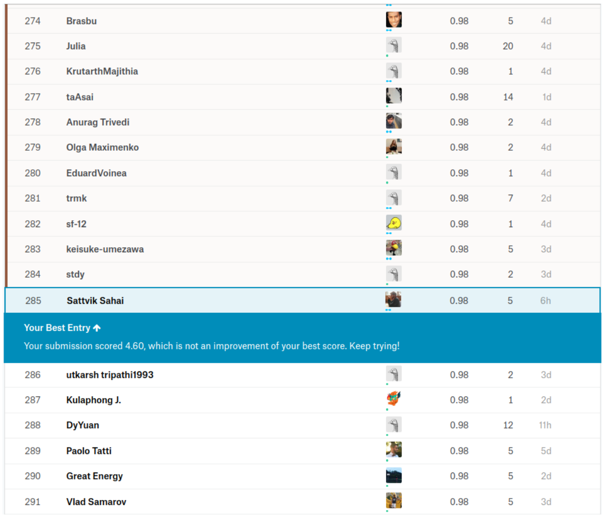
- Data Interpolation. Performed interpolation on *air temperature*, *cloud coverage*, *dew temperature*, *sea level pressure*, *wind direction*, *wind speed*, and *precipitation depth* using averaging.

5 Outcome

I participated in an active competition. My score is 0.98 in the public leaderboard. I rank 285/2846 in the public leaderboard. The results of the private leaderboard will not be available until December 19 2019. The screenshots are in Figure 3.

References

[1] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.



274	Brasbu	0.98	5	4d
275	Julia	0.98	20	4d
276	KrutarthMajithia	0.98	1	4d
277	taAsai	0.98	14	1d
278	Anurag Trivedi	0.98	2	4d
279	Olga Maximenko	0.98	2	4d
280	EduardVoinea	0.98	1	4d
281	trmk	0.98	7	2d
282	sf-12	0.98	1	4d
283	keisuke-umezawa	0.98	5	3d
284	stdy	0.98	2	3d
285	**Sattvik Sahai**	0.98	5	6h

Your Best Entry ↑

Your submission scored 4.60, which is not an improvement of your best score. Keep trying!

286	**utkarsh tripathi1993**	0.98	2	3d
287	**Kulaphong J.**	0.98	1	2d
288	**DyYuan**	0.98	12	11h
289	**Paolo Tatti**	0.98	5	5d
290	**Great Energy**	0.98	5	2d
291	**Vlad Samarov**	0.98	5	3d

(a) Public leaderboard.

Figure 3: Rankings in the leaderboard.