2018 Joint 10th International Conference on Soft Computing and Intelligent Systems and 19th International Symposium on Advanced Intelligent Systems

# Bayesian Optimization of Customer Churn Predictive Model

Kyungtae Kim, Jee-Hyong Lee
Department of Electrical and Computer Engineering
Sungkyunkwan University
Suwon, Republic of Korea
{kkt922, john}@skku.edu

*Abstract*— **This paper optimizes the customer churn predictive model using Bayesian Optimization. The customer churn predictive model is used as an important tool in Customer Relationship Management. However, hyperparameters must be set appropriately for high accuracy. In this paper, we optimize seven hyperparameters of the customer churn predictive model using Recurrent Neural Network. The experiment shows that the accuracy of the predictive model can be significantly improved. In addition, we plot the effect of each hyperparameter on the accuracy of the predictive model. This leads to the characterization of each hyperparameter in the customer churn predictive model.**

***Keywords—Bayesian Optimization, Customer Churn Predictive Model, Hyperparameter***

## I. INTRODUCTION

Customer Relationship Management (CRM) has become a major managerial strategy in many highly competitive organizations. The aim of CRM is to understand customers profitability and retain profitable customers as well [1]. Customer churn, which is defined as the propensity of customers to cease doing business with a company in a given time period, has become a significant problem and is one of the prime challenges many companies worldwide are having to face [2]. In order to effectively manage customer churn for companies, it is important to build a more effective and accurate customer churn predictive model. In literature, statistical and data mining techniques have been used to create the prediction models [3].

However, their performance critically relies on the proper setting of numerous hyperparameters. Manual tuning by an expert researcher has been a traditionally effective approach, however it is becoming increasingly infeasible as models become more complex and machine learning systems become further embedded within larger automated systems. Bayesian optimization has recently been proposed as a strategy for intelligently optimizing the hyperparameters of deep neural networks and other machine learning systems; it has been shown in many cases to outperform experts, and provides a promising way to reduce both the computational and human time required [4].

In this paper, we will optimize the hyperparameter of the customer churn predictive model [5] by applying Bayesian optimization. This predictive model uses Long Short Term Memory (LSTM) and Fully Connected Neural Network (FCNN). We derive the optimal value of each hyperparameter and the influence of each hyperparameter on the performance of the customer churn predictive model.

## II. BACKGROUND

### A. Bayesian Optimization

Bayesian optimization is a powerful tool for solving black-box global optimization problems with computationally expensive function evaluations [6]. Most commonly, this process begins by evaluating a small number of randomly selected function values, and fitting a Gaussian process (GP) regression model to the results. The GP posterior provides an estimate of the function value at each point, as well as the uncertainty in that estimate. We then choose a new point at which to evaluate the function by balancing exploration (high uncertainty) and exploitation (best estimated function value). This is done by optimizing an acquisition function, which encodes the value of potential points in the optimization and defines the balance between exploration and exploitation. A common choice for the acquisition function is expected improvement (EI), which measures the expected value of the improvement at each point over the best observed point. Optimization then continues sequentially, at each iteration updating the model to include all past observations [7].

Bayesian optimization has recently become an important tool for optimizing machine learning hyperparameters [8], where in each iteration a machine learning model is fit to data and prediction quality is observed [7].

Wu [9] showed how Bayesian optimization can exploit derivative information to find good solutions with fewer objective function evaluations. Because, unlike most optimization methods, Bayesian optimization typically does not use derivative information.

Snoek [10] explored the use of neural networks as an alternative to GP to model distribution over functions. Since GP scales cubically with the number of observations, it has been challenging to handle objectives whose optimization requires many evaluations, and as such, massively parallelizing the optimization.

### B. Customer Churn Predictive Model

Kim [5] proposed customer churn predictive models based on Deep Learning using behavior patterns of customers. As shown in Figure 1, Kim creates attributes that express customer behavior based on time. In particular, the behavior pattern of the customer is segmented and expressed based on a specific attribute (Attribute_standard). Using the preprocessed data, customer behavior patterns are extracted with LSTM as shown in Figure 2, and the results are applied to FCNN to predict customer churn.

978-1-5386-2633-7/18/$31.00 ©2018 IEEE
DOI 10.1109/SCIS-ISIS.2018.00024

| Customer ID | Time | Attribute_ standard | Attribute 1 | Attribute 2 |
|---|---|---|---|---|
| 4175D6K2 | 2017-03-04 | 1 | … | … |
| 4175D6K2 | 2017-03-05 | 1 | … | … |
| 4175D6K2 | 2017-03-06 | 1 | … | … |

| Customer ID | Time | Attribute_ standard | Attribute 1 | Attribute 2 |
|---|---|---|---|---|
| 4175D6K2 | 2017-03-04 | 2 | … | … |
| 4175D6K2 | 2017-03-05 | 2 | … | … |
| 4175D6K2 | 2017-03-06 | 2 | … | … |

Fig. 1. Preprocessed data based on time

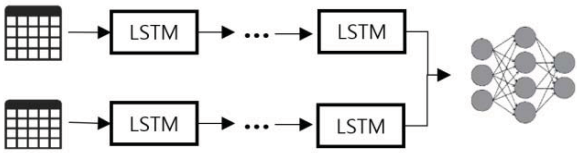



Fig. 2. The predictive model based on LSTM, FCNN

## III. PROPOSED METHOD

To maximize the performance of the predictive model, We tune hyperparameters using Bayesian Optimization. There are many hyperparameters to tune in predictive models (LSTM, FCNN). Among them, We select 7 hyperparameters to tune. The type and dimension of selected hyperparameters are shown in Table 1.

Learning rate is a hyperparameter that controls how much we are adjusting the weights of our network with respect the loss gradient. Dropout rate is the percentage of applying dropout which is a regularization method for reducing overfitting in neural networks. Number of LSTM, FCNN layers and the number of LSTM, FCNN nodes determine the size (complexity) of the entire model. The greater the number

TABLE I. SELECTED HYPERPARAMETERS

| Hyperparameter | Type | Dimension | |
|---|---|---|---|
| | | *Min* | *Max* |
| Learming rate | Real | $10^{-6}$ | $10^{-2}$ |
| Dropout rate | Real | 0.05 | 0.95 |
| # of LSTM layers | Integer | 1 | 5 |
| # of LSTM nodes | Integer | 5 | 512 |
| # of FCNN layers | Integer | 1 | 5 |
| # of FCNN nodes | Integer | 5 | 512 |
| Activation function | Categorical | Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU) | |

TABLE II. INITIAL VALUE OF HYPERPARAMETERS

| Hyperparameter | Initial value |
|---|---|
| Learming rate | $10^{-5}$ |
| Dropout rate | 0.4 |
| # of LSTM layers | 2 |
| # of LSTM nodes | 30 |
| # of FCNN layers | 2 |
| # of FCNN nodes | 150 |
| Activation function | ReLU |

because of the increased complexity. Activation function is non-linear function added to the output of LSTM, FCNN.

The initial value for Bayesian optimization is the optimized value when the person manually tunes. The initial value of hyperparameters is shown in Table 2. We apply Expected Improvement (EI) for acquisition function.

## IV. EXPERIMENT

### A. Training Bayesian Optimization

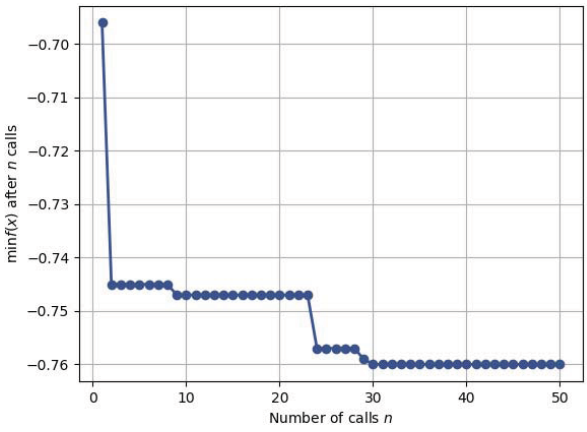
We used Windows 10 and the GeForce GTX 1080 Ti as a Graphics Processing Unit (GPU). We used Scikit-Optimize library in Python. Bayesian Optimization performed a total of 50 iterations. The negative accuracy at each iteration is shown in Figure 3. In this plot, BO converges only in 30 iterations. The total time taken for the experiment is about 30 hours.

### B. Result

The final accuracy, optimized using the BO, is 76%. This is 9.4% better accuracy than the manually optimized accuracy of 69.5%. The values of each optimized hyperparameters are shown in Table 3.

The Bayesian optimizer works by building a surrogate model of the search-space and then searching this model instead of the real search-space, because it is much faster. Figure 4 shows the last surrogate model built by the Bayesian optimizer where yellow regions are better and blue regions are worse. The black dots show where the optimizer has sampled

Fig. 3. Convergence plot of Bayesian Optimization



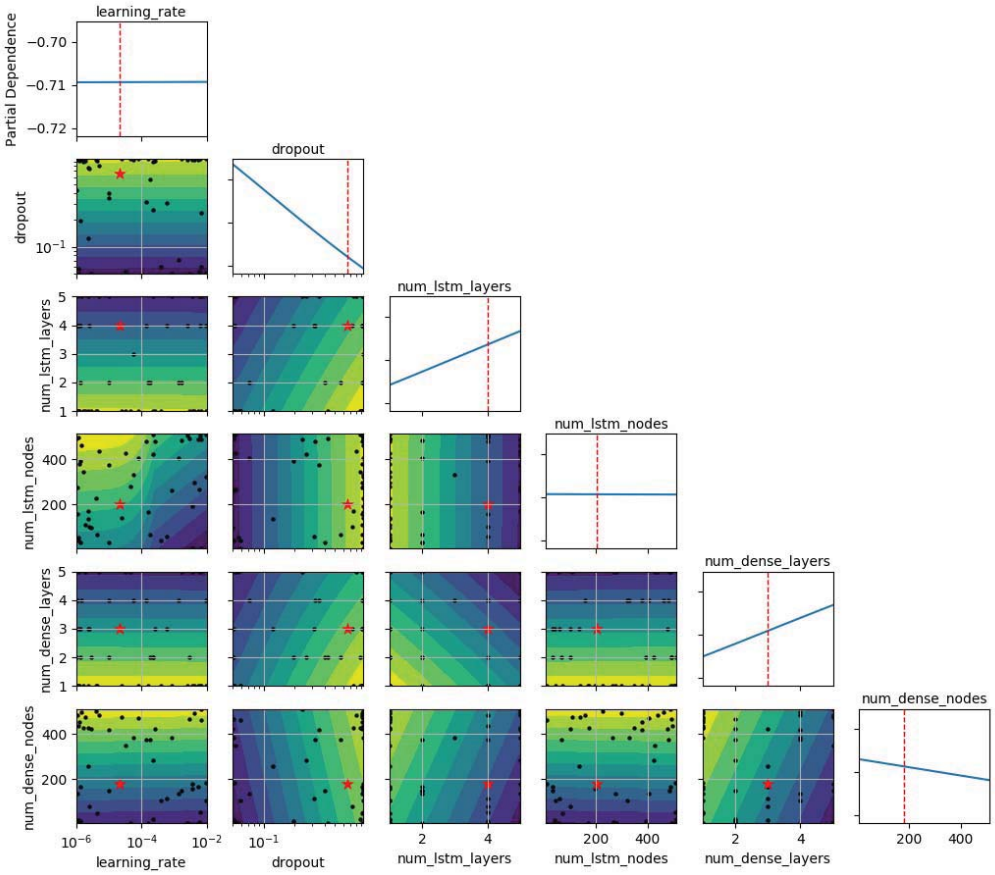



Fig. 4. Partial Dependence plots and the last surogate models built by the BO

the search-space and the red star shows the best parameters found.

The diagonal shows the influence of a single dimension with respect to the objective function. This is a so-called Partial Dependence plot for that dimension. It shows how the approximated fitness value changes with different values in that dimension. The learning rate has the same effect on the objective function regardless of the value. The higher the dropout rate, the smaller the effect on the objective function. The greater the number of LSTM layers and the number of dense layers, the greater the impact on the objective function. As the complexity of the model increases, the objective function seems to be more responsive. The number of dense nodes tend to be opposite.

The plots below the diagonal show the Partial Dependence for two dimensions. This shows how the approximated the objective function value changes when we are varying two dimensions simultaneously.

Figure 5 shows another type of matrix-plot. Here the diagonal shows histograms of the sample distributions for each of the hyper-parameters during the Bayesian optimization. The plots below the diagonal show the location of samples in the search-space and the colour-coding shows the order in which the samples were taken.

TABLE III. THE OPTIMIZED VALU ES OF HYPERPARAMETERS

| **Hyperparameter** | **Initial value** | **Optimized value** |
|---|---|---|
| Learming rate | $10^{-5}$ | $2.08 \times 10^{-5}$ |
| Dropout rate | 0.4 | 0.6561 |
| # of LSTM layers | 2 | 4 |
| # of LSTM nodes | 30 | 203 |
| # of FCNN layers | 2 | 3 |
| # of FCNN nodes | 150 | 181 |
| Activation function | ReLU | ReLU |

## V. CONCLUSION

In this paper, we optimized the customer churn predictive model using Bayesian Optimization. We selected seven hyperparameters to optimize. Expected Improvement was used as the acquisition function of the BO. Experimental results show that the accuracy of the predictive model was 9.4% higher than that of the existing model manually optimized . By showing the partial dependence plot, we confirmed the effect of each hyperparameter on the objective function. This can be

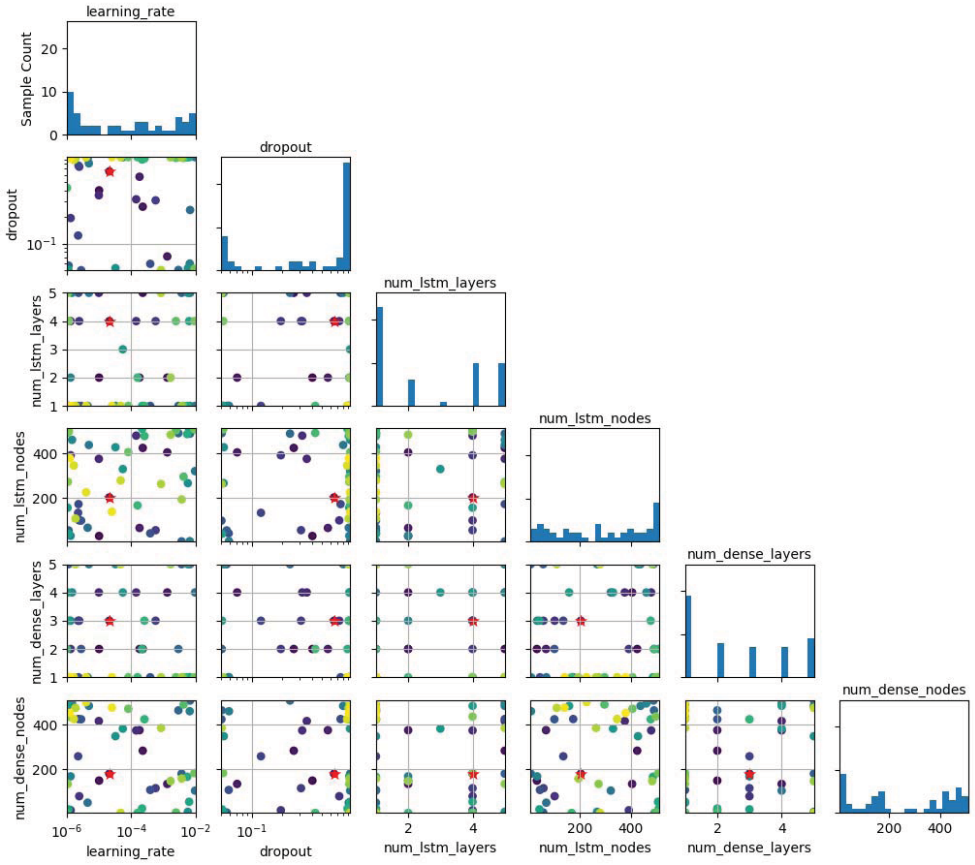



Fig. 5. The histograms of the sample distributions and the location of samples in the search-space

used to identify the nature of each hyperparameter in the customer churn predictive model.

## ACKNOWLEDGMENT

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7069440).

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014M3C4A7030503).

## REFERENCES

[1] Hung, Chihli, and Chih-Fong Tsai. "Market segmentation based on hierarchical self-organizing map for markets of multimedia on demand." Expert systems with applications 34.1 (2008): 780-787.

[2] Chandar, M., Arijit Laha, and P. Krishna. "Modeling churn behavior of bank customers using predictive data mining techniques." National conference on soft computing techniques for engineering applications (SCT-2006). 2006.

[3] Tsai, Chih-Fong, and Yu-Hsin Lu. "Customer churn prediction by hybrid neural networks." Expert Systems with Applications 36.10 (2009): 12547-12553.

[4] Swersky, Kevin Jordan. Improving Bayesian Optimization for Machine Learning using Expert Priors. Diss. 2017.

[5] Kyungtae Kim, and Jee-Hyong Lee. "Predictive Models for Customer Churn using Deep Learning and Boosted Decision Trees." Journal of Korean Institute of Intelligent Systems 28.1 (2018): 7-12.

[6] Jones, Donald R., Matthias Schonlau, and William J. Welch. "Efficient global optimization of expensive black-box functions." Journal of Global optimization 13.4 (1998): 455-492.

[7] Letham, Benjamin, et al. "Constrained Bayesian optimization with noisy experiments." arXiv preprint arXiv:1706.07094 (2017).

[8] Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. "Practical bayesian optimization of machine learning algorithms." Advances in neural information processing systems. 2012.

[9] Wu, Jian, et al. "Bayesian optimization with gradients." Advances in Neural Information Processing Systems. 2017.

[10] Snoek, Jasper, et al. "Scalable bayesian optimization using deep neural networks." International Conference on Machine Learning. 2015.