Effect of Dropout Layer on Classical Regression Problems

Conference Paper · October 2020

CITATIONS
9

READS
2,851

2 authors:

Atilla Özgür
Constructor University Bremen gGmbH
27 PUBLICATIONS **152** CITATIONS
SEE PROFILE

Fatih Nar
Ankara Yildirim Beyazit University
70 PUBLICATIONS **215** CITATIONS
SEE PROFILE

Effect of Dropout layer on Classical Regression Problems

1st Atilla Özgür
Mathematics and Logistics
Jacobs University
Bremen, Germany
ORCID: 0000-0002-9237-8347

2nd Fatih Nar
Department of Computer Engineering
Ankara Yıldırım Beyazıt University
Ankara, Turkey
ORCID: 0000-0002-3003-8136

Abstract—In the last decade, deep learning architectures have provided good accuracy as they become deeper and wider in addition to other theoretical improvements. However, despite their current success, they initially faced with overfitting issue that limits their usage. The first practical and usable solution to overfitting in deep neural networks is a simple approach known as the dropout. Dropout is a regularization approach that randomly drops connections from earlier layers during training of neural nets. Dropout is a widely used technique, especially in image classification, speech recognition and natural language processing tasks, where features created by earlier layers are mostly redundant. Usage of the dropout layer in other tasks is largely unexplored. In this study, we seek an answer to question if the dropout layer is also useful for classical regression problems. A 3 layer deep learning net with a single dropout layer with various dropout levels tested on 8 real regression datasets. According to the experiments, the dropout layer does not help over fitting.

Keywords—deep learning, neural network, regression, overfitting, regularization, dropout

I. Introduction

Deep neural networks have become more popular in the last decade due to mostly their success on wide variety of tasks [1]. Like classical neural networks, deep learning neural nets also suffer from overfitting problem, also known as memorization [2]. Simpler models less tend to over fit to the training data while complex neural network models with many layers and many neurons has more tendency to over fit to the training data. For classical neural networks, different solutions to overfitting is suggested among them early stopping is the popular approach [3]. Also, $L1$-norm and $L2$-norm regularization are employed to prevent overfitting for traditional neural networks where smaller weights are forced to obtain simpler models [4]. For all these approaches, use of cross-validation is suggested to obtain a best bias-variance trade-off to avoid overfitting and achieving generalization [5].

Co-dependency amongst fully connected layers can be formed during the training since all the weights are learned together. This causes weaker connections to be ignored while powerful connections with more predictive capability getting even stronger. Traditional $L1$-norm and $L2$-norm regularization are not effective since their regularization is already based on the predictive capability of the connections. So strong connections get stronger and the weak connections get weaker due to this deterministic approach. This co-dependency that causes overfitting degrades the performance of the learned model in the test phase [6].

For deep neural networks, one of the most popular solutions to overfitting is the dropout approach [7] that is patented by Google [1]. Original dropout article [7] is cited over 17000 times according to Google Scholar in January 2020. In dropout, instead of learning all the weights together a randomly selected fraction of the weights are learned during training. Because of introduced randomness, training process become noisy which forces nodes in each layer to take more or less responsibility for the inputs. This approach enforces a robustness to the learned model by preventing network layers to co-adapt to correct mistakes from prior layers. Therefore, this simple approach resolved the overfitting issue in large networks, especially for deep neural networks [6], [7].

When one searches among the articles that cited original dropout article for regression in title, there seems to be a lot of regression articles such as [8]–[10]. However, these articles focus on the regression on computer vision problems but not on classical regression problems. A recent review [11] also focused on the regression on computer vision problems. Also, search with keyword "dropout deep learning regression" gives a lot of results, but most of them again belong to computer vision regression problems. As an exception, [12] recently proposed a deep learning regression for stream temperature problem that uses early stopping, dropout and its variant DropConnect [13] since dropout itself becomes insufficient for regression.

The aim of this study is to show affect of dropout layer on classical regression problems. A 3 layer deep learning net with a single dropout layer with various dropout levels tested on 8 real regression datasets. According to the experiments, the dropout layer does not help over fitting. On the contrary, in the 7 datasets, the dropout layer reduced test dataset performance. In the remaining one dataset, the dropout layer has at best a negligible effect on the test dataset performance. From these experiments,

[1]Google Patent: US9406017B2
System and method for addressing overfitting in a neural network
https://patents.google.com/patent/US9406017B2/en

978-1-7281-7206-4/20/$31.00 ©2020 IEEE

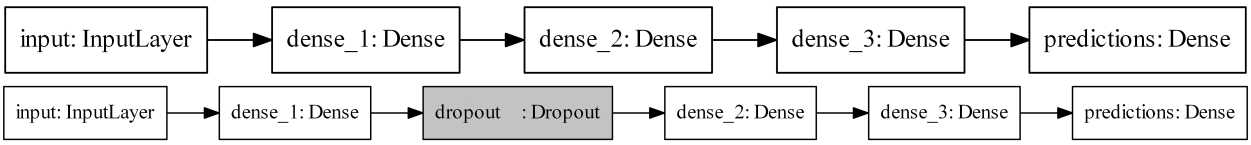


Figure 1: Neural networks in this study

Table I: Dataset information

dataset name	rows	features	Approximate size	download link
Diabetes	442	10	4.420	https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html
Boston	506	13	6.578	https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
Wind	6.574	14	92.036	https://www.openml.org/d/503
California Housing	20.640	8	165.120	https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html
MV Artificial Data	40.768	10	407.680	https://www.dcc.fc.up.pt/~ltorgo/Regression/mv.html
Pol	15.000	48	720.000	https://www.openml.org/d/201
Poker Hand	1.025.010	10	10.250.100	https://archive.ics.uci.edu/ml/datasets/Poker+Hand
BNG(pbc)	1.000.000	18	18.000.000	https://www.openml.org/d/1191

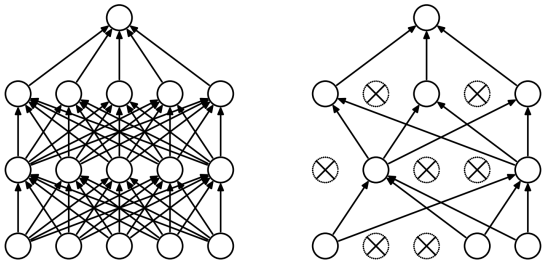
we can conclude that the dropout layer is not useful for classical regression problems.

The structure of the rest of this paper is as follows: Section II introduces the materials and methods, Section III presents experimental results, and finally section IV concludes the paper.

II. Material and Methods

A. Dropout Layer

Dropout layer is a simple technique to reduce over fitting in the neural networks. Dropout layer is used with single parameter named dropout rate, chance of dropping out connections. Dropout rate changes between 0 and 1. If this parameter is 0.5, coming neural connections to dropout layer is dropped with 50% chance only when neural net is training, see Figure 2.



(a) Standard Neural Net (b) After applying dropout.

Figure 2: Dropout in Neural Networks ([7])

To test the effect of dropout layer, 3 layers deep learning neural net with optional dropout layer is used in this study. These deep neural networks can be seen in Figure 1. Dropout rate in this optional layer is changed between 0 and 0.5 with increase of 0.1 rate.

B. Datasets

8 different regression datasets are chosen for the experiments. Table I shows the information for datasets used in the experiments. These datasets come from different domains. Table I download link column shows from where the corresponding dataset is downloaded. Approximate size column is multiplication of rows and features columns to show how complex is the corresponding dataset. For the training dataset, 80% is selected randomly from full dataset, for testing dataset, the remaining 20% is used.

C. Libraries

Following libraries are used in this study: Tensorflow 2 [14], Keras [15], scikit-learn [16] and PLMB machine learning benchmark library [17].

III. Experimental Results

A. Experiments

Table II and Figure 3 show results of the experiments. Normally, in statistics regression algorithms are trained and tested on the same dataset; therefore, R^2 values always are in the range of $[0, 1]$. But, here since we are testing our Deep Learning model on outside test set, we can get values outside the range of $[0, 1]$, thus, Some of the R^2 values in the Table II are negative. Another reason for negative values that trained model is a very bad fit to test dataset [18].

As can be seen in Figure 3, R^2 values either dramatically drops and stays almost the same. These results shows that, at least for these datasets, dropout layer does not help over fitting in deep learning problems.

Figure 4 shows zoomed results for three datasets, Diabetes, Wind and BNG. R^2 results for Diabetes dataset change very slightly. For Diabetes dataset, dropout has a negligible effect on both train and test set results. R^2 results for Wind dataset reduces both on train and test datasets. They start 0.79 and drops to -0.10, showing that for wind dataset trained deep learning models with high

Table II: Experiment Results for R^2 Values According to Drop out Rate and Dataset

dataset name	dropout value R^2	0.0	0.1	0.2	0.3	0.4	0.5
Diabetes		0.464320	0.467957	0.463929	0.462949	0.466320	0.467447
Boston		0.605194	0.461621	0.372973	0.380099	0.239699	0.429705
Wind		0.790589	0.740548	0.702499	0.537453	0.317982	-0.101584
California housing		0.602037	0.629591	0.567010	0.342244	0.189784	0.277773
MV Artificial Data		0.999668	0.996620	0.990862	0.900465	0.759711	0.628739
Pol		0.995619	0.985658	0.979394	0.933576	0.924112	0.876146
Poker Hand		0.792075	0.689202	0.389502	0.209848	0.255563	0.154803
BNG (pbc)		0.385580	0.369965	0.314704	0.177496	0.007424	-0.230664

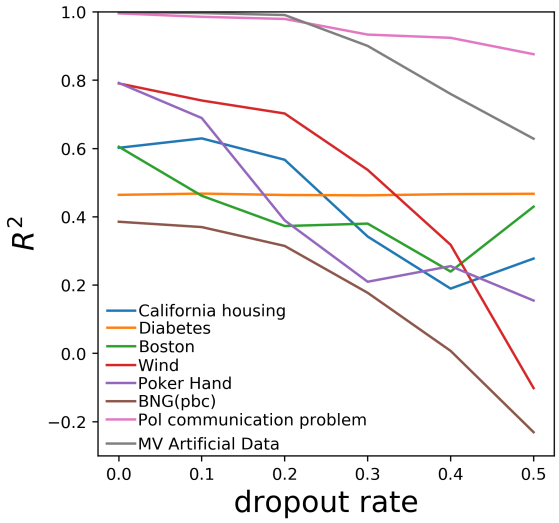
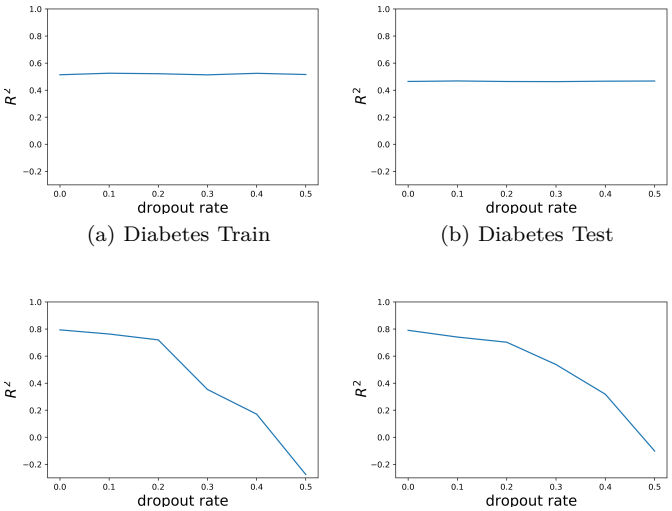


Figure 3: Dropout vs R^2 in All 8 Datasets



(a) Diabetes Train (b) Diabetes Test

(c) Wind Train (d) Wind Test

(e) BNG (pbc) Train (f) BNG (pbc) Test

Figure 4: Dropout vs R2 in 3 Datasets

dropout rates are very bad fit to test dataset [18]. Similar results can also be seen in BNG dataset.

We also show critical difference diagrams (Figure 5) proposed by Demšar [19] for comparison of machine learning algorithms over multiple datasets. Even though, critical difference diagrams are proposed for classifiers, it can still be used for R^2 measure comparison for regression problems. According to this diagram, there are significant differences between pair wise testing on statistical test (Wilcoxon-Holm method) on all algorithms. That is, even though low dropout rate DL methods are better than others, statistical difference is very low for these datasets.

IV. Conclusion

In this study, we investigated the usage of the dropout regularization approach in classical regression problems which is largely unexplored. Dropout is a widely used and well-performing technique in image classification, speech recognition, and natural language processing tasks. However, our experiments and investigations show that the dropout approach is not an effective technique for classical regression problems. Its usage in classical regression problems has a rarely positive effect while generally it lowers the test performance or at most have a negligible effect.

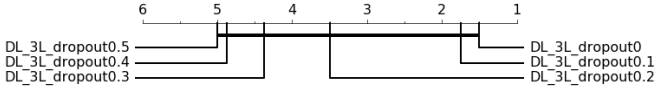


Figure 5: Critical Difference Diagram (Figure generation code due to [20])

Future investigations with more regression datasets and more deep neural networks are planned so that conclusions drawn from this study can be made more general.

Acknowledgments

Authors thank the anonymous reviewers whose comments/suggestions helped improve and clarify this manuscript.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, 2015.

[2] S. S. Haykin, Neural Networks And Learning Machines. Pearson Education, 2009.

[3] L. Prechelt, "Early stopping - but when?" in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1998.

[4] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in Proceedings of the Twenty-First International Conference on Machine Learning. Association for Computing Machinery, 2004, p. 78.

[5] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning: data mining, inference and prediction, 2nd ed. Springer, 2009.

[6] P. Baldi and P. J. Sadowski, "Understanding dropout," in Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2814–2822.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, 2014.

[8] H.-I. Suk, S.-W. Lee, and D. Shen, "Deep ensemble learning of sparse regression models for brain disease diagnosis," Medical Image Analysis, 2017.

[9] H.-I. Suk and D. Shen, "Deep ensemble sparse regression network for alzheimer's disease diagnosis," in Machine Learning in Medical Imaging. Springer International Publishing, 2016.

[10] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in The IEEE International Conference on Computer Vision (ICCV), 2015.

[11] S. Lathuiliére, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "A comprehensive analysis of deep regression," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[12] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Impact of deep learning-based dropout on shallow neural networks applied to stream temperature modelling," Earth-Science Reviews, vol. 201, p. 103076, 2020.

[13] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in Proceedings of the 30th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, vol. 28, no. 3. PMLR, 2013, pp. 1058–1066.

[14] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[15] F. Chollet et al., "Keras," https://keras.io, 2015.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, 2011.

[17] R. S. Olson, W. L. Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore, "PMLB: a large benchmark suite for machine learning evaluation and comparison," BioData Mining, 2017.

[18] J.-M. Dufour, "Coefficients of determination," McGill University, Tech. Rep., 1983.

[19] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," The Journal of Machine Learning Research, 2006.

[20] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," Data Mining and Knowledge Discovery, vol. 33, no. 4, pp. 917–963, 2019.