A New Cluster-Aware Regularization of Neural Networks

Tolga Ahmet Kalaycı[1[0000-0001-5706-6455]], Umut Asan[1[0000-0002-0838-1421]], Murat Can Ganiz[2[0000-0001-8338-991X]] and Aydın Gerek[2[0000-0001-9875-7041]]

[1] Istanbul Technical University, Department of Industrial Engineering, Maçka, İstanbul, Turkey
[2] Marmara University, Department of Computer Engineering, Göztepe, Istanbul, Turkey

kalaycit@itu.edu.tr, asanu@itu.edu.tr, murat.ganiz@marmara.edu.tr, aydin.gerek@marmara.edu.tr

Abstract. Inherent clusters formed by observations used for the training of a classification model is a frequently encountered case. These clusters differ in certain characteristics, however in classical modelling techniques no information on these differences is fed into the model. Differentiations in purchasing styles of e-commerce customers may be a good example for this case. While some customers like to do research and comparisons on price, functionalities and comments, some others may need a shorter examination to decide on their purchase. In a similar manner, purchasing journey of a deal seeker customer would differ from a luxury buyer customer. In this paper, we propose a neural network model which incorporates different cluster information in its hidden nodes. Within the forward propagation and backpropagation calculations of the network, we use a non-randomized Boolean matrix to assign hidden nodes to different observation clusters. This Boolean matrix shuts down a hidden node for observations which do not belong to the cluster that the node is assigned to. We performed experiments for different settings and network architectures. Also, analyses are conducted to study the influence of alternative application patterns of the Boolean matrix on the results – expressed in terms of iterations and epochs for an Adam (adaptive moment estimation) optimization. Empirical results demonstrate that our proposed method works well in practice and compares favorably to fully randomized alternatives.

Keywords: Neural networks, clustering, classification, regularization, machine learning, e-commerce.

1 Introduction

Classification tasks in real-life applications tend to be performed on large volume of data [1]. As the volume of training data increases, groups which are formed by observations that have similar characteristics become more evident. For example, various studies have demonstrated that purchasing journeys and motivations of e-commerce customers significantly differ from each other [2]. Kau, Tang and Ghose [3] has clas-

sified online shoppers into six groups based on their information seeking patterns as well as their motivations and concerns for online shopping. In another study, Rohm and Swaminathan [4] developed a typology which is categorizing online shoppers as convenience shoppers, variety seekers, balanced buyers, and store-oriented shoppers. Jayawardhena, Tiu and Dennis [5] segmented online retail customers based on their purchase orientation. In their study, customers were clustered into five different purchase orientation such as active shoppers, price sensitives, discerning shoppers, brand loyal and convenience-oriented.

Purchase propensity modeling, which is mainly a classification task, is a common marketing tool used in the e-commerce domain. It is assumed that inherent groupings among observations of a training set carry valuable information about critical patterns that drive conversion. However, ordinary classification techniques are not usually aware of such groupings or at least they must learn them on their own. In this paper, we propose a new cluster aware regularization for neural networks which incorporates this group information via a non-randomized Boolean matrix that is designed according to clusters formed by training observations. Also, through the alternating usage of randomized and non-randomized Boolean matrix, our algorithm successfully fulfills the regularization task in a neural network model.

The main contributions of this paper are (i) to recommend a neural network model which embeds the information of different training observation clusters in hidden nodes, (ii) to propose a new cluster-aware regularization method for neural networks, and (iii) to show favorable performance of the proposed method against dropout method.

This paper is structured as follows: a brief literature search on neural networks and regularization is given in Section 2. Afterwards in section 3, the proposed method is explained. The proposed method is evaluated against dropout and the findings are summarized in Section 4. Conclusions and future works are given in Section 5.

2 Neural Networks and Regularization

Neural networks are biologically-inspired programming paradigms which enable a computer to learn from observational data [6]. They are very powerful prediction systems. However, overfitting is a serious problem for these models. To address this issue, various approaches have been proposed in the literature for regularizing neural networks. One of the most popular and recognized approaches is dropout. It has been firstly introduced by Hinton et al. [7] in 2012. The key idea of the dropout method is to randomly drop hidden units in desired hidden layers during training. Although it has been around for a few years, it is a widely accepted regularization technique. Apart from dropout, L1 / L2 regularization, Soft Weight Sharing and Elastic Norm Regularization may be listed as the best-known regularization techniques. There are also intuitive regularization approaches like Early Stopping that basically aims to stop training when the validation error has not sufficiently improved for a certain number of iterations. In addition to these classical techniques, new regularization approaches were recently introduced. Li et al. [8] proposed a smooth group regularization tech-

nique for feedforward neural networks. The main advantage of their techniques is that it can remove some redundant weights of the surviving nodes as well as redundant nodes. Korchi and Ghanou [9] introduced a regularization method that sets all the weak weights within all the layers of a neural network to zero. They called their technique as DropWeak and compared its results against dropout using the MNIST data set. In another study, Iosifidis et al. [10] proposed an extension of Extreme Learning Machine algorithm for neural network training that uses dropout and DropConnect regularization in its optimization steps. They concluded that their proposed approach can lead to better network output weights without much additional computational costs.

3 Proposed Method

Traditional classification models do not take account of inherent clusters that represent separate groups of training observations with similar characteristics. However, in most real-life problems this type of clusters occur and they tend to affect the accuracy of predictions. Feed of such an information can be provided by adding an independent variable to the model. However, a single variable is not expected to make a significant difference in a model with a large number of independent variables. An alternative way for taking advantage of this information could be training separate models for each cluster. Increased training cost and other potential side effects such as incomparability of results between models should be considered as the drawbacks of this alternative.

In this section, we describe our proposed method that aims to address this issue. It basically depends on clustering the observations before the training phase and then allocating specific nodes in the hidden layers of the network to one of these clusters during the training.

Fig. 1. summarizes how our method assigns hidden nodes to two different clusters observed in the training set. As seen in the figure, since we have two clusters, we have two Boolean vectors which are both of size 1 x m. The first one includes “1” for observations that belong to first cluster and “0” for observations that do not belong to first cluster. The second Boolean vector has just the opposite sequence. It is composed of “1” for observations that belong to second cluster and of “0” for observations that do not belong to the second cluster. As depicted in Fig. 1, activation vectors of the first and second hidden nodes of the first hidden layer are multiplied by the first Boolean vector before they are forwarded to the next layer. Similarly, activation vectors of the third and fourth hidden nodes are multiplied by the second Boolean vector to compute the resulting activation matrix of the first hidden layer. In this way, we allocate the first two hidden nodes to the first cluster, and they transfer information to next layer only for this group of observations. For all other observations always “0” value is generated by these nodes. In the same manner, third and fourth nodes generate a non-zero value only for observations that belong to the second cluster.

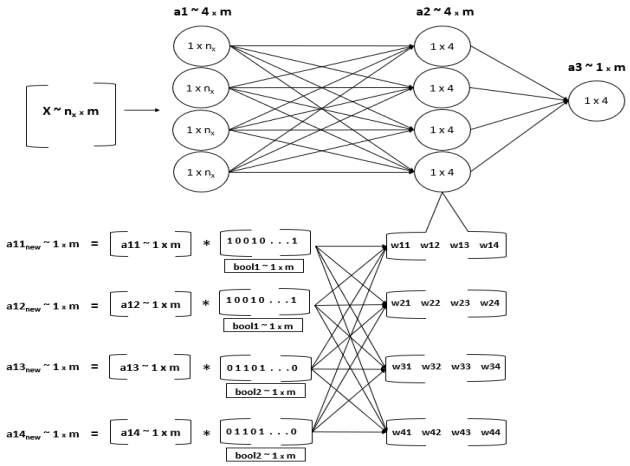

Fig. 1. The sketch describes the use of a Boolean matrix to assign hidden nodes of first hidden layer to two different clusters that are formed in the input data set X. "n_x" is the number of independent variables, "m" is the number of observations, a1 – a2 and a3 are the activation matrices of related layers, a11 – a12 – a13 – a14 are the activation vectors of related hidden nodes, $a11_{new}$ – $a12_{new}$ – $a13_{new}$ – $a14_{new}$ are the updated activation vectors of related hidden nodes after Boolean matrix multiplication

From this point on, we will call our non-randomized Boolean matrix as "**Cluster Info Matrix**". An important part of our algorithm is that in consecutive iterations we switch between our Cluster Info Matrix and a completely randomized Boolean matrix. In other words, for iterations 1, 3, 5, …, 2n-1 we multiply the activations by the Cluster Info matrix and for iterations 2, 4, 6, …, 2n we multiply the activations by a completely randomized Boolean matrix. Later, we will give the details on why we have incorporated such an alternate use.

Obviously, for different network architectures and more than two clusters, different strategies for node to cluster allocation and different hidden layer choices for Cluster Info Matrix application are possible. These scenarios are explained in the next section.

4 Experiments

In this section, we present how we have evaluated our cluster-aware regularization method. We conducted our experiments on a real-life binary classification data set consisting of 80524 observations (60463 training, 20061 test), 248 independent variables and one dependent variable. Forward propagation and backpropagation steps were coded in Python 3.6.4. During our experiments, we used four different network architectures which are 3x3, 4x4, 3x3x3, 4x4x4. In all experiments ADAM (adaptive moment estimation) with mini batches was used as the optimization algorithm. The number of clusters and the number of hidden units of the layer on which the Cluster Info Matrix is applied are kept the same in all experiments. In the next two subsec-

tions, we first explain how we clustered the training data and then we present the results of our experiments.

4.1 Clustering the training data

For the 3x3 and 3x3x3 network architectures we used Hierarchical clustering with Complete L1 distance and divided the data into three clusters consisting of 12287, 48152 and 24 observations. For the 4x4 and 4x4x4 architectures we implemented the K-Means algorithm and divided the data into four clusters consisting of 8368, 185, 12287 and 39623 observations.

4.2 Application of proposed method

In this part, we apply the proposed method to different layers in different network architectures and compare the results with those of dropout regularization method which incorporates a completely randomized Boolean matrix. To allow for a valid comparison between the proposed method and dropout, all the model parameters and random seeds were kept the same for both methods and in each experiment the same layers were multiplied by the corresponding Boolean matrix. We selected the area under the receiver operating characteristic curve (AUC) as our main comparison metric since it is a widely-accepted threshold-free performance indicator for classification models. During our experiments, we also examined the value of the cost function in both training and test sets.

In our preliminary trials, we observed that if we apply the Cluster Info Matrix regularly in all iterations, the value of the training cost function tends to increase after a certain number of iterations. This is obviously an unexpected behavior in a convergence process. To fix this problem, we suggest the alternating use of the Cluster Info Matrix and randomized Boolean matrix in our methodology as stated in section 3. The refined method ensures a monotonic decrease of the cost function (see Fig. 2).

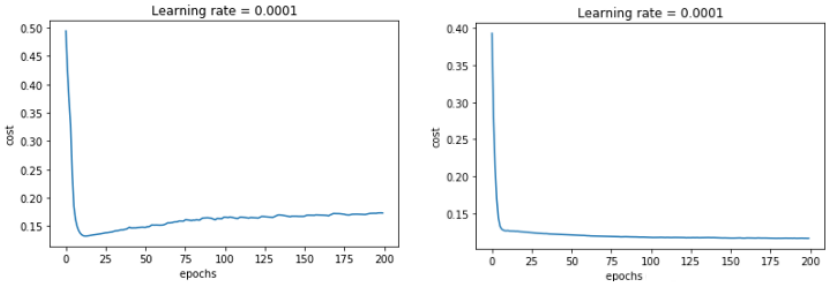

Fig. 2. Increase in training cost after a certain number of iterations (left image) and monotonic decrease after the alternating use is adopted (right image).

We further evaluated our algorithm by using the Cluster Info Matrix on each hidden layer separately in four different architectures. Depending on the results of the experiments, we can conclude that when applied to the first hidden layer, the proposed

method clearly outperforms the dropout method. Table. 1 outlines the results of eight different trials with different coefficient initializations and network architectures. As seen in the "Increase in AUC" column of the table, in all trials our proposed method resulted in higher Test AUC values compared to the dropout method. This result perfectly matches with our intuitive expectations because what we basically do here is to allocate a separate entrance point to the network for each cluster. In this way, the cluster information is passed to the rest of the network by distinguishing between activations of the first hidden layer.

Moreover, we conducted some other trials by defining the Cluster Info Matrix in a fully randomized way. These trials consistently resulted in lower AUC values compared to the proposed method. This indicates that allocating a separate entrance node to network for each cluster and feeding this information to the next hidden layer yields a difference.

Table 1. Test AUC comparison results between proposed method and dropout for first hidden layer application. Each trial was conducted with a different random initialization seed.

Trial Number	Network Architecture	AUC (Proposed Method)	AUC (Dropout)	Increase in AUC
1	3x3	88,271%	86,143%	2,128%
2	3x3	79,877%	77,738%	2,139%
1	4x4	88,013%	87,123%	0,890%
2	4x4	88,572%	88,134%	0,438%
1	3x3x3	88,142%	86,164%	1,978%
2	3x3x3	88,900%	88,247%	0,653%
1	4x4x4	87,074%	86,295%	0,779%
2	4x4x4	89,075%	88,281%	0,794%

Fig. 3 and Fig. 4 indicate changes in training cost, test cost and test AUC values through 200 epochs for 3x3 and 4x4x4 architectures, respectively. In both experiments, the proposed method produced more robust and less oscillating curves for the test AUC values.

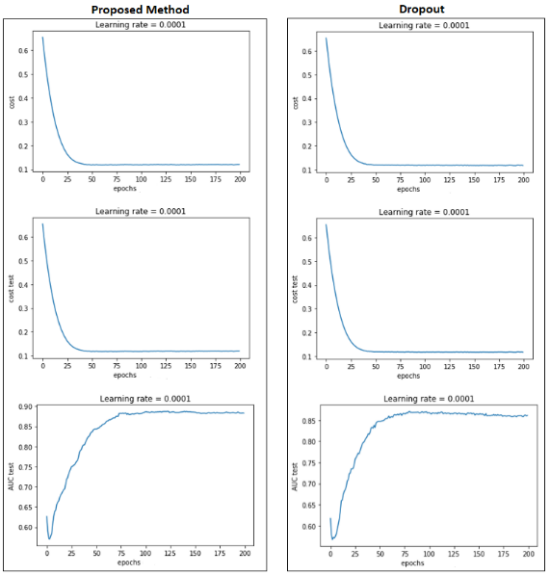

Fig. 3. Proposed Method VS dropout in terms of training cost, test cost and test AUC in 3x3 architecture. AUC values for proposed method and dropout are 88,271% and 86,143%.

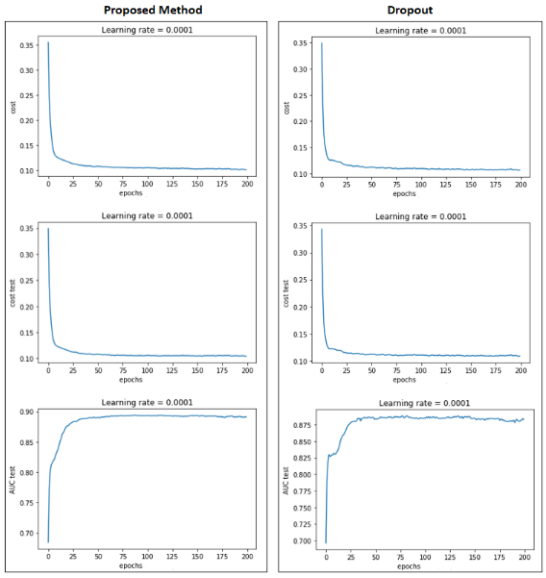

Fig. 4. Proposed Method VS dropout in terms of training cost, test cost and test AUC in 4x4x4 architecture. AUC values for proposed method and dropout are 89,075% and 88,281%.

5 Conclusions

In this paper, we have presented a new cluster-aware regularization method for neural networks. The key idea in this algorithm is to feed a neural network with the information on clusters inherent in the data in a suitable way to improve the regularization task and prediction quality.

We provided experimental results for different network architectures and compared the resulting AUC metrics with those of the dropout method. From these experiments, we can conclude that the proposed method works well in practice and partitioning the hidden nodes of the first hidden layer to different clusters gives better AUC results compared to the dropout method.

For future work, we plan to enhance the proposed method to allow better predictions. Furthermore, experiments with different data sets may provide more insight on the contributions of the proposed algorithm. Finally, analysis of the effects of factors such as the diversity level between clusters, number of clusters, and proximity of clusters (in terms of the number of observations included) on the success of predictions are potential research topics for future work.

References

1. Ericson, K., Pallickara, S.: On the performance of high dimensional data clustering and classification algorithms. Future Generation Computer Systems 29, 1024–1034 (2013)
2. Malecki, K., Watrobski, J.: The Classification of Internet Shop Customers based on the Cluster Analysis and Graph Cellular Automata. In: International Conference on Knowledge Based and Intelligent Information and Engineering Systems, pp. 2280-2289. Procedia Computer Science, Marseille, France (2017)
3. Keng, Kau A., Tang, Y.E.: Ghose S. Typology of online shoppers. Journal of Consumer Marketing 20(2), 139–156 (2003)
4. Rohm, A.J., Swaminathan, V.: A typology of online shoppers based on shopping motivations. Journal of Business Research 57(7), 748– 757 (2004).
5. Jayawardhena C., Tiu, W. L., Dennis, C.: Consumers online: intentions, orientations and segmentation. International Journal of Retail & Distribution Management 35(6), 515–526 (2007).
6. Nielsen, M. A.: Neural Networks and Deep Learning, Determination Press, (2015).
7. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R.: Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. CoRR, abs/1207.0580, 2012.
8. Li, F., Zurada, J. M., Wu, W.: Smooth group L1/2 regularization for input layer of feedforward neural networks. Neurocomputing 314, 109-119 (2018).
9. Korchi, A. E., Ghanou. Y.: DropWeak: A novel regularization method of neural networks. In: The First International Conference On Intelligent Computing in Data Sciences, pp 102-108. Procedia Computer Science, (2018)
10. Iosifidis, A., Tefas, A., Pitas, I.: DropELM: Fast Neural Network Regularization with Dropout and DropConnect. Neurocomputing 162. 57-66 (2015)