Feature Selection based Neural Network Design for Improved Accuracy

Ashiquzzman Akm*, Dong Su Lee*, Sang Woo Kim*, Lin Van Ma*, Um-Tae Won** and Jin Sul Kim*

Abstract

Deep learning neural networks are a special type of machine learning algorithm that learns the important data representation of features by themselves in a optimization algorithm named backpropagation. As the whole neural network data updation and feature selection was done by themselves, the idea was custom feature section was not studied properly. Decision tree is another machine learning algorithm that learns from the entropy of the datasets itself and seperates the data into desired labels classes. In this paper the main feature selection based neural network wasdesigned and experimented for better performance. In the end the result was slightly improves which suggested feature selectio nbased neural network often give better result in specific dataset.

Key words

Neural Network, Deep Learning, Node Pruning, Optimization, Decision Tree

Ⅰ. Introduction

Decision Tree is a special machine learning algorithm that devices and classify data based on the values itself. A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and separate based on the attributes or decision point that we signifies. Edges represent the separation the to and the leaves represent the actual output or class label. They are used in non-linear decision making with simple linear decision surface. Decision tree gives a proper idea of important attributes ar features of an given dataset that is implemented on the dataset. The important indexed feature might contains more important classification factors based on the tree buildup mechanism. On the other hand, Neural Networks are a specilized learning algorithms that learns by the dataset feature and representation by a mechanism named backpropagation algorithms. neural networks are sometimes users the whole dataset to learn the classification by itself.

In this paper, the idea of neural network training [1] with general method and improved method of with selected features are being tested.

II. Proposed Method And Result

The main dataset that was selected for this study in the NIST [2] digit dataset . The main dataset was given and given the premises that the main dataset has 4200 instances each represents a binary image of a handwritten English numeral digit. The English handwritten Numerical digit class has 10 labels, spanning from 0 to 9. So, basic arithmetic suggests that we have 420 instances or images for a single label data class. The whole images were 18 by 24 dimensions, so the whole image has 18 × 24 =

* School of Electronics and Computer Engineering, Chonnam National Uniersity, Gwangju, South Korea
** Department of Information and Communication Engineering, Chosun University, Gwangju, South Korea

432 data points.

The whole images were given in a single data text file. Then a new python program was written to process this data to get array for the models to learn. I used the Numpy numerical libraries to process this data to input data the decision tree and MLP. The whole dataset were read into the raw array of (100800, 18) numerical value and then transformed into (4200, 24, 18) Multi-dimensional array. Later Scipy page API was used to generate the whole dataset into 70-30 training testing splitting.

In the MLP the neural network was designed very simply. The neural network input has 432 dimension to fit in all the data input as input layer then the hidden first later has 32 nodes that was connected to the 16 nodes in the second layer. The final output layer has 10 node to indicate the output result. The whole network has the hyper tangent (tanh) as the activation function. The “Categorical Cross Entropy” was chosen for the loss/error function and the whole network gradient update was done by the Root Mean squared propagation or “RMSProp” algorithms. The whole network was trained for 50 iterations on batch size 200. The Neural network was performed well on the training. It achieved 98.60% of overall accuracy while the testing validation gave on average 94% accuracy. This problem might occur due to overfitting. However, no such brackets were implemented to reduce over fitting as this was out of the scope for this study.

The Neural network was performed well on the training. It achieved 98.60% of overall accuracy while the testing validation gave on average ~94% accuracy. This problem might occur due to overfitting. However, no such brackets were implemented to reduce over fitting as this was out of the scope for this study.

The Python based library for Machine learning Scipy was implemented to construct the tree based on the feature importance. The all input data dimension were ranked based on the scoring in total dataset classification and data column scored more than 0.003 score were selected as more important feature. Only 212 feature were selected based on the cutoff value.

In the MLP the neural network was designed focusing on main comparison basics. In this second part, the main design of neural network in first experiment were reused for benchmarking purposes. Here neural network input has 212 dimension to fit in all the data input as input layer then the hidden first later has 32 nodes that was connected to the 16 nodes in the second layer. The final output layer has 10 node to indicate the output result. The whole network has the hyper tangent (tanh) as the activation function. The “Categorical Cross Entropy” was chosen for the loss/error function and the whole network gradient update was done by the Root Mean squared propagation or “RMSProp” algorithms. The iteration was 50 with 200 batch size as before.

On these second part the result was as slightly improved. The whole dataset were reduced dimensionally, making the network training time improved in small scale. However, the data accuracy improved due to rearrangement of neural network and training data distribution. The overall accuracy is ~95.86% in overall system

Ⅲ. Conclusion

This paper proposes a novel solution for deep learning neural network optimization. The IoT based application solution sometimes needs to be both memory and computation resource efficient. Deep learning models are often gets very big in model input dimention and precise accurate classification. in this paper, A decision tree based model input feature selection was devised to het better result. based on the dataset and data distribution, it might be profitable to select features based on decision tree first then to apply neural network.

Acknowledgement

This work was supported by Institute for Information &communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2018-0-00691, Development of Autonomous Collaborative Swarm Intelligence Technologies for Disposable IoT Devices). Besides, this research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion). Furthermore, this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST)(Grant No. NRF-2017R1D1A1B03034429)

Reference

[1] Hagan, Martin T., Howard B. Demuth, Mark H. Beale, and Orlando De Jesús. Neural network design. Vol. 20. Boston: Pws Pub., 1996.

[2] Bottou, Léon, Corinna Cortes, John S. Denker, Harris Drucker, Isabelle Guyon, Lawrence D. Jackel, Yann LeCun et al. "Comparison of classifier methods: a case study in handwritten digit recognition." In Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on, vol. 2, pp. 77-82. IEEE, 1994.