An efficient fuzzy optimization algorithm based on convolutional neural network

Gaihua Wang^{1,2}, Wenzhou Liu^{2,*}, Mengqi Nie³ and Tao Li²

¹Hubei Collaborative Innovation Centre for High-efficiency Utilization of Solar Energy, Hubei University of Technology, Wuhan, 430068, China

²School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China

³School of Industrial Design, Hubei University of Technology, Wuhan 430068, China

Keywords: Convolutional neural network, Optimization algorithm, Fuzzy membership, Network weights.

Abstract. The paper proposes a method based on dense-sparse-dense optimization algorithm. It uses sparsity to tune network weights. By adding fuzzy membership, the optimization strategy can enhance the feature information with larger weights and weaken the feature information with less weight. Through accurate cutting of network weights, parameters in network are effectively reduced. The experimental results show that the performance of this method is better than the existing method.

1 Introduction

Deep learning is a research field of machine learning, which forms a more abstract high-level representation feature by combining low-level features to discover the characteristics of data [1]. It evolved from multi-layer unsupervised pre-training initialization to convolutional neural networks (CNN). Lenet[2] is the first popular CNN model which is a major breakthrough in deep learning, and consists of two convolutional layers, two pooled layers, and one fully connected layer. In 2012, AlexNet[3], which successfully uses ReLU as the activation function and uses dropout to decrease overfitting, won the Imagenet challenge. The subsequent networks of GoogleNet[4] and VGG[5] have better network performance which are all based on Alexnet. They tried various methods such as the size of different convolution kernels, the depth of the convolutional network, and the processing of image input. VGG is composed of 3*3 convolutional kernel. It proves that the increase of CNN depth could improve the accuracy of the image classification. The training numbers of GoogleNet structured with Inception is not greatly increasing the amount of computation, but the accuracy of image classification in the database of ImageNet is higher than that of AlexNet. The ResNet[6] launched in 2015 solves the problem of the disappearance of gradients. And It maps feature maps from low level to high level network directly by short connection. Saining Xie[7] present a highly modularized network architecture that is constructed by repeating a building block

* Corresponding author: wenzhouromeo@163.com

that aggregates a set of transformations with the same topology. Springenberg et al[8] questioned the down-sampling layer in CNN, and they design a full CNN. From current research circumstance, it has been exploring in two major fields: the depth of CNN and optimizing structure. The appearance of spatial pyramid pooling [9] shows very good result. Its core is to extract features in convolution layers after different pooling size on the featured image, then aggregate to featured vector. XNOR-Net [10] and BinaryNet have also made very interesting advances in smart devices using binarization methods. Other networks were also proposed[11-14]. However, with the increasing network layers, the number of parameters has increased significantly, which has led to an increasing difficulty of training. In 2017, Song Han[15] proposed a dense-sparse-dense(DSD) algorithm, focusing on how to improve the accuracy of the traditional model by selective network weights. Later, Ma et al[16] proposed the sparse to dense algorithm to attain a higher level of robustness and accuracy by introducing additional sparse depth samples. Naveed Akhtar et al[17] proposed a dense collaborative representation with a sparse representation for classification.

Based on above methods, we proposed a Fuzzy-DSD algorithm. This method uses the degree of membership to measure network weights. The train is divided into three phrases: dense-fuzzy sparse-dense. By adding fuzzy phase, the method can enhance accurate cutting of network weights. The experiment shows that the proposed method has better performance than the existing method.

2 The DSD algorithm

The DSD algorithm consists of three steps (Figure 1). Dense, it trains networks to get connection weights. The dots represent Neurons and the connection lines represent the connections between the Neurons. Sparse, it regularizes the network by pruning the unimportant connections with small weights and retraining the network given the sparsity constraint. In the phase, the unimportant connections and weights in the network are pruned into small and medium networks. Dense, it removes the sparsity constraint, re-initialize the pruned parameters from zero and retrain the whole dense network. In the phase, it is to increase the model capacity

Fig. 1. DSD algorithm network structure.

$Mask$ denotes sparseness(Equation(1)). When the weight is less than λ , $Mask$ is set to zero. And the weights are remained. When $Mask$ is 1, the weights are removed. it is expressed by Equation(2).

$$Mask = 1(|W^{(t-1)}| > \lambda) \quad (1)$$

$$W^{(t)} = W^{(t)} \cdot Mask \quad (2)$$

3 The proposed method

Aim to inaccuracy of removing unimportant network weights, we add the membership to DSD. Figure 2 is our proposed method which consists of three parts: Dense, Fuzzy phase, and Dense. The first phase is trained to obtain the initial network weights. The second phase, the membership is used to judge the importance of network weights.

$$S = sum(W^{(\varphi)}), \varphi = 1, \dots, N \quad (3)$$

$$\eta_{ij} = \frac{W_{ij}^{\varphi}}{S}, i = 1, \dots, m; j = 1, \dots, n; \quad (4)$$

Given W denotes the weight. N is the number of weights each layer. The S is the sum of network weights. η indicates the membership of whole weight. Then the unimportant connections and weights in the network are weakened.

Fig. 2. Fuzzy-DSD algorithm network structure.

The third stage is to retrain the weights in the network. The structure changes the capacity of the network by pruning and filling the network weights. After the weights are updated by equation(3), they are multiplied by membership η (Equation(4)). And it uses membership to quantify the absolute value of weights to make the network more accurate.

$$W_{ij}^{(t)} = W_{ij}^{(t-1)} - \alpha^{(t)} \nabla f(W_{ij}^{(t-1)}; x^{(t-1)}) \quad (5)$$

$$W_{ij}^{(t)} = W_{ij}^{(t)} \cdot \eta_{ij} \quad (6)$$

α is the learning rate, x is the input. The Workflow of Fuzzy-DSD training is as following.

Algorithm 1: Workflow of Fuzzy-DSD training

Initialization: $W^{(0)} \sim N(0, \Sigma)$

Output: $W^{(t)}$

Initial Dense Phase

While not converged do

$$W^{(t)} = W^{(t-1)} - \alpha^{(t)} \nabla f(W^{(t-1)}; x^{(t-1)});$$

$$t = t + 1;$$

end

Fuzzy Phase

$$S = \text{sum}(W^{(l)}), l = 1, \dots, N.$$

$$\eta_{ij} = \frac{W_{ij}^l}{S}, i = 1, \dots, m; j = 1, \dots, n.$$

While not converged do

$$W_{ij}^{(t)} = W_{ij}^{(t-1)} - \alpha^{(t)} \nabla f(W_{ij}^{(t-1)}; x^{(t-1)});$$

$$W_{ij}^{(t)} = W_{ij}^{(t)} \cdot \eta_{ij};$$

$$t = t + 1;$$

End

Final Dense Phase

While not converged do

$$W^{(t)} = W^{(t-1)} - \alpha^{(t)} \nabla f(W^{(t-1)}; x^{(t-1)});$$

$$t = t + 1;$$

end

go to Sparse Phase for iterative Fuzzy-DSD;

4 Experiments

In this section, experiments on MNIST and CIFAR-10 are conducted to demonstrate the effectiveness of the proposed method. We also compare the proposed method with DSD on different CNN.

4.1 MNIST data set

The MNIST is a classic handwritten digital data set, which has 60,000 training sets and 1000 test sets consist of 28*28 pixel gray image. From Table1, with the increasing iterations, the accuracy of the network models is improved. For LeNet-5 and Alexnet, we add DSD and Fuzzy-DSD. When the iteration is 40, the accuracy of LeNet-5 is 98.43. LeNet-5 with DSD is 98.7. LeNet-5 with Fuzzy-DSD is 99.8. The accuracy of Alexnet is 92.96. Alexnet with DSD is 93.48. Alexnet with Fuzzy-DSD is 94.36. The accuracy of the Fuzzy-DSD is higher than that of others.

Table 1. The test results of the three algorithms on the MNIST dataset.

methods	Iterations								
	1	5	10	15	20	25	30	35	40
LeNet-5	80.80	95.94	97.14	97.8	97.94	98.37	98.26	98.57	98.43
DSD LeNet-5	91.31	96.94	97.95	98.36	98.48	98.63	98.62	98.56	98.7
Fuzzy-DSD LeNet-5	91.71	98.74	98.81	99.15	99.22	99.04	99.06	98.97	99.08
Alexnet	18.35	61.72	77.34	87.11	83.98	89.45	91.41	92.96	92.96
DSD Alexnet	30.31	63.43	78.95	88.26	84.67	90.52	92.71	93.11	93.48
Fuzzy-DSD Alexnet	30.82	66.25	80.46	89.78	86.61	93.03	93.92	93.74	94.36

1. CIFAR-10 data set experiment results.

The CIFAR-10 data set has 60,000 color images which are 32*32 pixels and divided into 10 categories. There are 50,000 for training and 10,000 for testing. For LeNet-5 and Alexnet, we also add DSD and Fuzzy-DSD. From table 2, it shows that the network with the Fuzzy-DSD is better than others. When the iteration is 40, the accuracy of LeNet-5 is 49.7. LeNet-5 with DSD is 51.29. LeNet-5 with Fuzzy-DSD is 54.03. The accuracy of Alexnet is 76.27. Alexnet with DSD is 79.89. Alexnet with Fuzzy-DSD is 82.59.

Table 2. Test results of two algorithms on CIFAR-10 dataset.

methods	iterations								
	1	5	10	15	20	25	30	35	40
LeNet-5	15.07	29.12	33.5	39.91	42.23	45.67	45.74	46.89	49.7
DSD LeNet-5	25.46	34.06	44.31	46.42	48.97	51.71	52.14	50.52	51.29
Fuzzy-DSD LeNet-5	26.71	36.93	44.95	46.74	49.94	51.38	51.88	53.13	54.03
Alexnet	52.67	55.42	58.7	61.22	63.78	66.67	71.07	74.88	76.27
DSD Alexnet	63.84	65.52	67.06	67.51	68.25	72.83	74.29	77.91	79.89
Fuzzy-DSD Alexnet	64.51	65.98	70.46	68.85	68.92	72.47	74.12	79.68	82.59

From the table1-2, we can see that, with the increasing iterations, the error of the Fuzzy-DSD has a decreasing trend. And the use of fuzzy DSD can make networks faster convergence. From Table1, it is found that LeNet-5 achieves better classification than

AlexNet. For AlexNet, It may be related to the excessive training parameters. When the training samples and times are less, the network cannot be fully trained.

5 Conclusion

Based on DSD algorithm, an optimization method is proposed by using membership of the weights. The Fuzzy-DSD algorithm can not only preserve the image feature better, but also suppress the unimportant network weights. The experimental results also show that the performance of fuzzy DSD algorithm is better than others.

This work is supported by the National Nature Science Foundation of China under Grant No. 61601176.

References

1. G. E. Hinton, R.R.S., Reducing the Dimensionality of Data with Neural Networks. *Science*, 2006. 313(5786): p. 504-507.
2. Lecun, Y., et al., Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 86(11): p. 2278-2324.
3. Krizhevsky, A., I. Sutskever and G.E. Hinton. ImageNet classification with deep convolutional neural networks. in *International Conference on Neural Information Processing Systems*. 2012. pages 1106 – 1114.
4. Szegedy, C., et al., Going deeper with convolutions. 2014(9): p. 1-9.
5. Simonyan, K. and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science*, 2014: p. 1409.1556.
6. He, K., et al., Deep Residual Learning for Image Recognition. 2015(12).
7. Xie, S., et al., Aggregated Residual Transformations for Deep Neural Networks. *arXiv:1611.05431v2*, 2017. 2(4).
8. J, L.S.S.C., Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, 2006: p. 2006 , 2 :2169-2178.
9. He, K., et al., Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans Pattern Anal Mach Intell*, 2015. 37(9): p. 1904-16.
10. Rastegari, M., et al., XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. 2016: p. 525-542.
11. Sabour, S., N. Frosst and G.E. Hinton, Dynamic Routing Between Capsules. 2017(11).
12. Zhang, C., J. Cheng and Q. Tian, Image-level classification by hierarchical structure learning with visual and semantic similarities. *Information Sciences*, 2018. 422(9): p. 271-281.
13. Zheng, P., et al., A set-level joint sparse representation for image set classification. *Information Sciences*, 2018. 448-449(2): p. 75-90.
14. Mai, S.D. and L.T. Ngo, Multiple kernel approach to semi-supervised fuzzy clustering algorithm for land-cover classification. *Engineering Applications of Artificial Intelligence*, 2018. 68: p. 205-213.
15. Song Han, H.M.E.G., DSD: DENSE-SPARSE-DENSE TRAINING FOR DEEP NEURAL NETWORKS. 2017(2).

16. Ma, F. and S. Karaman, Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. 2017.
17. Akhtar, N., F. Shafait and A. Mian, Efficient classification with sparsity augmented collaborative representation. Pattern Recognition, 2017. 65: p. 136-145.