PG23
Daejeon, Korea

A Simple Stochastic Regularization Technique for Avoiding Overfitting in Low Resource Image Classification

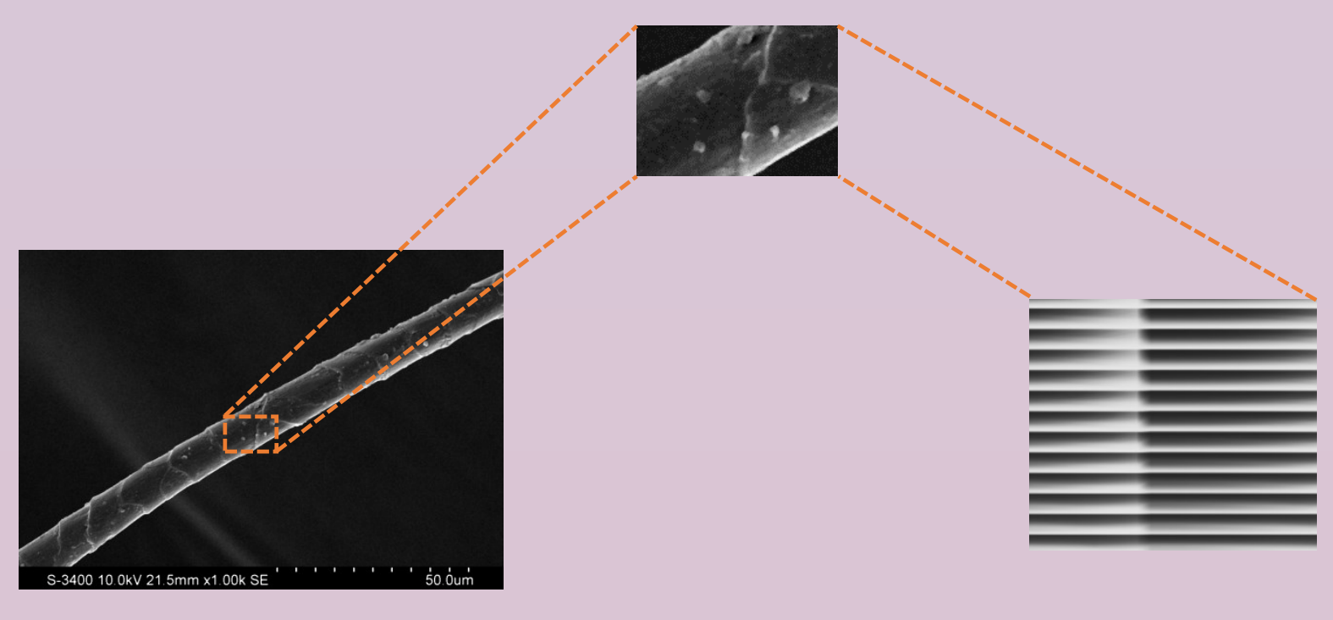
Yatu JI[1]✉, Bailun WANG[1], Qing-dao-er-ji REN[1], Bao SHI[1], Nier WU[1], Min LU[1], Na LIU[1], Xufei ZHAUNG[1], Xuanxuan XU[1], Li WANG[2], Lingjie DAI[2], Miaomiao YAO[2], Xiaomei LI[2]

1. Inner Mongolia University of Technology 2. National Testing Center of Wool and Cashmere Quality

✉ MLjyt@imut.edu.cn

PROBLEM

Low resource image classification faces serious overfitting problems, and the data sparsity problem weakens or even disappears the effectiveness of most regularization methods. Serious overfitting often leads to some features (a) being considered as noise (b), which can be viewed as preventing neurons from co-adapting.


(a) (b)

RELATED WORK

Dropout[1] has been proposed since 2012 to avoid overfitting problems in visual recognition challenges and has gradually been applied to a wider range of research. Subsequently, a series of variants emerged, such as standout[2], which alleviated overfitting problems by adding additional layers to control the proportion of neurons participating in training and reasoning with a certain probability. Dropconnect[3] transforms the regularization method from neuron control to weight control that is more easily mathematically estimated by using Drop weights instead of Drop neurons in Dropout. Similar variants also appear in Maxout[4] and Dropout as weight regulation[5].

These studies focus more on optimizing network structures to avoid the problem of general methods lacking universality under certain special conditions. And this issue has become prominent in low resource image classification.

OVERVIEW

In this paper, we address this question on the proposed Metcalfe-Drop method to find a flexible parameter sharing method for overfitting. To achieve this goal, the main contributions of this paper are as follows:

⋆ This paper proposes a Drop type method based on Metcalfe'law, *Metcalfe-Drop*, to explore parameter sharing methods in neural networks and reduce overfitting caused by resource scarcity.

⋆ We have constructed multiple strong baseline systems on specific and public datasets, including Resnet-50, VGG-16, Inception V2, and Conv Net, to validate the effectiveness of the Metcalfe-Drop method. The experimental results show that using Metcalfe-Drop to control the training neurons can achieve parameter sharing of neural networks in a new way and solve overfitting problems.

Our code is available at :
https: //gitee.com/giteetu/metcalfe−drop.git

ACKNOWLEDGEMENTS

This study is supported by the National Natural Science Foundation of China (62206138, 62066035), Education Department Science Research Foundation of Inner Mongolia Autonomous Region (JY20220186, NJZZ23081) , Science Research Foundation of Inner Mongolia University of Technology (BS2021079, ZZ202118), Support Program for Young Scientific and Technological Talents in Inner Mongolia Colleges and Universities (NJYT23059), Inner Mongolia Science and Technology Program Project (2020GG0104).

METHODOLOGY

The description of value and number of neurons in Metcalfe's law is intuitive and simple, as shown in formula(1-(i)). When it is associated with neural network training, formula(1) can be seen as a linear representation between the training value V and the neuron n. Setting the value coefficient K as a hyperparameter can make the Metcalfe-Drop(*Figure.1*) method better adapt to a variety of loss functions. For the general low resource classification research, we suggest that it should be set as a constant of no more than 0.5. Optional, k can also be obtained through iterative training of each batch. The constructed value iteration training expects to obtain the determined number of neurons n through training value V. Therefore, under the constraint of n ∈ [1,∞], formula(1-(i)) can be transformed into formula(1-(*ii*)).

$$V = K \times n^2 \quad (i) \quad \rightarrow \quad n = \sqrt{\left|\frac{V}{K}\right|} \quad (ii) \quad n \in [1, \infty) \qquad (1)$$

Then, how to get the training value is the key to inference. Inspired by VI, it is intuitive and reasonable to regard training loss as training value. The effectiveness of such strategy has been fully verified in the research of NLP. It is worth noting that more than one loss function can be applied to this strategy.

In order to avoid a large number of normalization calculations, we take softmax loss as an example to show the transformation process from training loss to training value. Specifically, the prediction probability of the *i*-th sample for class *j* can be expressed as $p_{ij} = \frac{e^{l_{i,j}}}{\sum_{m=j}^{C} e^{l_{i,m}}}$.

Where $l_{i,j}$ represents the logit output by the neural network for the *i*-th sample in the *j*-th class. Logit represents the output of the last full connection layer. C is the total number of categories. Where C_{y_i} is the embedding center of the label corresponding to the *i*-th sample. N is the number of samples for the current training.

$$L_{SE} = \frac{1}{n}\sum_{i=1}^{n}(1 - logp_i, y_i) = \frac{1}{n}\sum_{i=1}^{n}\left(-log\frac{e^{l_{ij}}}{\sum_{j=1}^{C} e^{l_{ij}}}\right) \quad (i)$$

$$(2)$$

$$L_C = \frac{1}{2}\sum_{i=1}^{n}\left\|x_i - C_{y_i}\right\|_2^2 \quad (ii)$$

Therefore, the loss of model training is expressed as $Loss = LSE + LC$. Then, two types of loss values, VSE and VC, can be obtained through value transformation. VSE and VC can be expressed as $VSE = Z_{Score}(LSE)$ and $VC = Z_{Score}(LC)$. Therefore, the training value can be determined by setting a hyper-parameter α expressed flexibly as $V = |\alpha VSE + (1-\alpha)VC|$. The inference process of Metcalfe-Drop can be described as the algorithm shown in *Figure.2*.

So far, we can derive the optimized number of neurons through Metcalfe value, so as to further provide a reliable probability $p^{Metcalfe} \sim \frac{n}{N} = \frac{1}{N}\sqrt{\left|\frac{V}{K}\right|}$ for the sampling basis in Metcalfe-Drop.

RESULTS

We trained Metcalfe-Drop neural networks for classication problems on data sets in two different domains, including extremely low resource dataset of wool and cashmere fiber image(0.8K for training set and 0.2K for test set) and common dataset CIFAR-10/100We found that Metcalfe-Drop improved generalization performance on all data sets compared to neural networks that use other Drop type techniques.

We constructed several strong baseline models based on a variety of methods. *Figure.5* shows the error rate obtained by different baselines on fiber image and CIFAR-10/100. Note that training a network of 25 million parameters to give good generalization error is very hard with standard regularization methods and early stopping. Metcalfe-Drop prevent overfitting and does not even need early stopping. Due to the sparsity of activation units in low resource classification tasks, we observed that the sparsity of hidden unit activations on a random mini-batch taken from the test set. *Figure.4* compares the sparsity for the Resnet-50 with/without Metcalfe-Drop. As we thought, a good sparse model should only be a few highly activated units for any dataset. Thus, we plot histograms to observe the sparsity performance of Metcalfe-Drop in neural network training. Comparing the results, we can see that fewer hidden units have high activations in *Figure.4*, as seen by the significant mass away from zero that does not use Metcalfe-Drop. As an example, *Figure.3* shows an embedded representation of the generated fiber image after learning through an Metcalfe-Drop network.

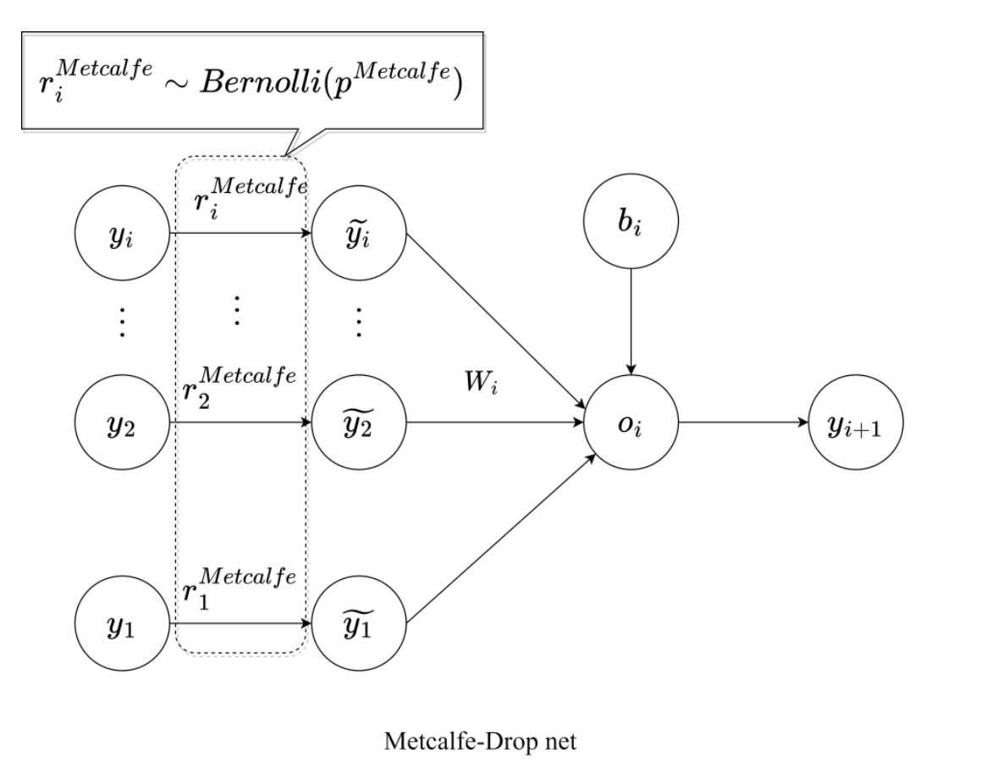


Figure.1 Feedforward calculation process of Metcalf-Drop.

```
Inference with Metcalfe-Drop
Input: example X, parameters θ,
      of Samples O, dimension of softmax d
Output: prediction Z
Extract features: $y \longleftarrow f(X;W_\theta)$
for o = 1:O do %% Draw o samples
   for i = 1:d do %% Loop over units in y
      sample from Metcalfe-Drop Z∼Bernoulli($p^{Metcalfe}$)
      $y \longleftarrow f(Z)$
   end for
end for
Pass result $y$ to next layer
```

Figure.2 Algorithm of inference with Metcalf-Drop.

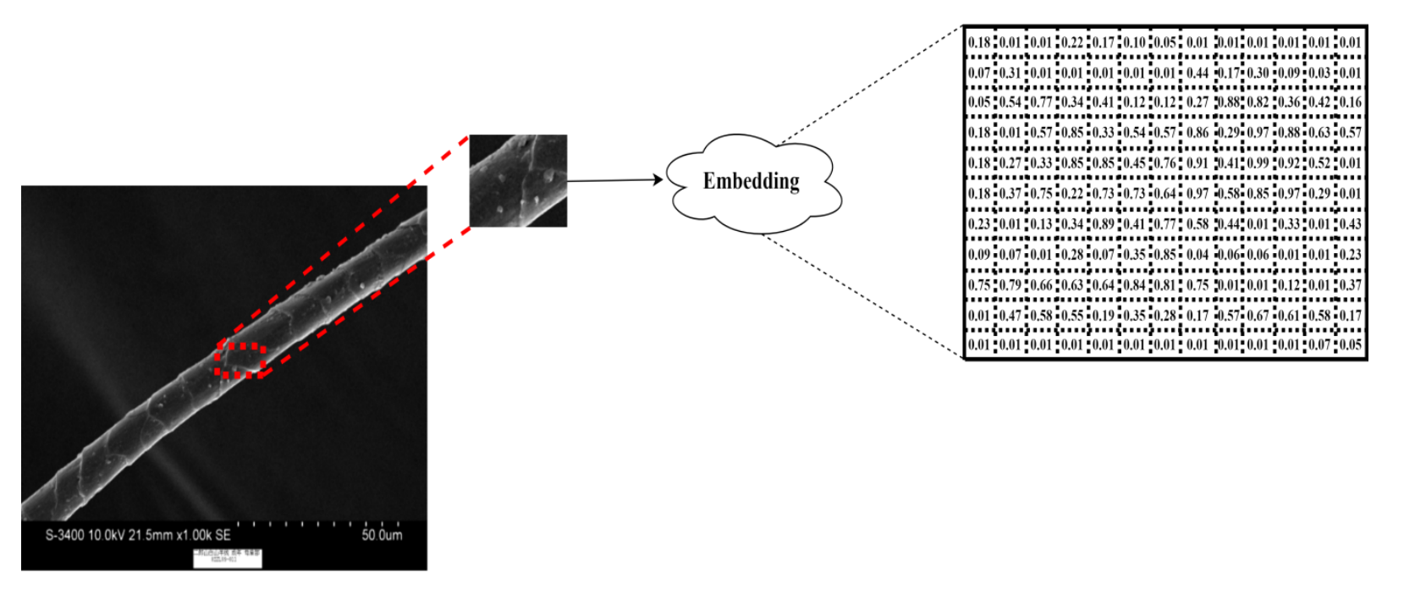


Figure.3 Embedding representation of sparse features in fiber images. We can see that the feature sparsity problem is alleviated after learning from the Metcalfe-Drop network.

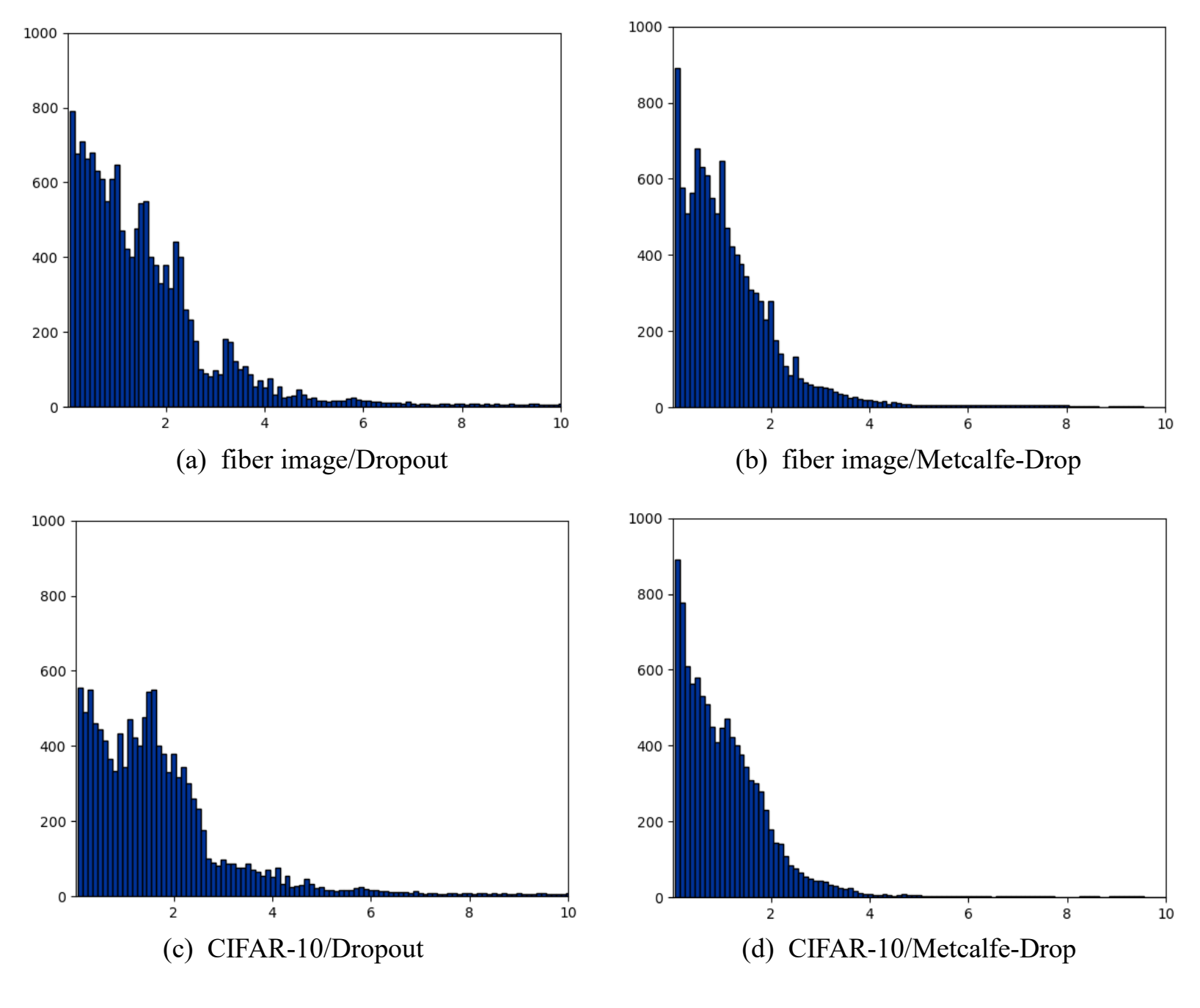


Figure.4 Effect of Metcalfe-Drop on sparsity.

Table 1: Error rates on fiber image dataset and CIFAR-10/100

Method	fiber image	CIFAR-10	CIFAR-100
Resnet-50(*Shen et al., 2022*)	17.71	39.65	52.20
Resnet-50+Dropout	13.60	38.31	50.70
Resnet-50+Dropconnect	13.88	34.43	55.10
Resnet-50+Metcalfe-Drop	12.51	**34.00**	**46.37**
Resnet-50+Metcalfe-Dropconect	**10.75**	34.90	48.18
VGG-16(*Simonyan et al., 2015*)	23.50	41.77	63.55
VGG-16+Dropout	23.15	38.00	**57.61**
VGG-16+Dropconnect	21.20	37.32	58.27
VGG-16+Metcalfe-Drop	20.58	37.20	51.35
VGG-16+Metcalfe-Dropconect	**18.31**	**33.24**	58.10
Conv Net+Dropout (*Snoek et al., 2012*)	17.44	39.90	55.10
Conv Net+Maxout (*Goodfellow et al., 2013*)	13.37	41.17	58.14
Conv Net+Metcalfe-Drop	**11.56**	**37.22**	**54.50**
GoogleLeNet *Szegedy et al., 2015*	14.63	**39.90**	51.02
GoogleLeNet+Metcalfe-Drop	**13.27**	39.95	**43.08**

Figure.5 Effect rates on fiber image dataset and CIFAR-10/100.

REFERENCES

[1] Srivastava N , Hinton G , Krizhevsky A ,et al.Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J].Journal of Machine Learning Research, 2014, 15(1):1929-1958.

[2] L. J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," in Proceedings of the 26th International Conference on Neural Information Processing Systems. NIPS, 2013, 26(1):1173-1186.

[3] Wan L, Zeiler M, Zhang S, et al. Regularization of neural networks using dropconnect[C]. International conference on machine learning. PMLR, 2013: 1058-1066.

[4] Goodfellow I, Warde-Farley D, Mirza M, et al. Maxout networks[C]. International conference on machine learning. PMLR, 2013: 1319-1327.

[5] S.Wager, S.Wang, and P. S. Liang. Dropout training as adaptive regularization[C]. Advances in neural information processing systems, NIPS, 2013:351–359.