Clustering-Based Adaptive Dropout for CNN-Based Classification

Zhiwei Wen, Zhiwei Ke, Weicheng Xie(✉), and Linlin Shen

Computer Vision Institute, School of Computer Science and Software Engineering,
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ),
Guangdong Key Laboratory of Intelligent Information Processing,
Shenzhen University, Shenzhen 518060, China
wcxie@szu.edu.cn

Abstract. Dropout has been widely used to improve the generalization ability of a deep network, while current dropout variants rarely adapt the dropout probabilities of the network hidden units or weights dynamically to their contributions on the network optimization. In this work, a clustering-based dropout based on the network characteristics of features, weights or their derivatives is proposed, where the dropout probabilities for these characteristics are updated self-adaptively according to the corresponding clustering group to differentiate their contributions. Experimental results on the databases of Fashion-MNIST and CIFAR10 and expression databases of FER2013 and CK+ show that the proposed clustering-based dropout achieves better accuracy than the original dropout and various dropout variants, and the most competitive performances compared with state-of-the-art algorithms.

Keywords: Feature and weight clustering · Feature derivative dropout · Self-adaptive dropout probability · Facial expression recognition

1 Introduction

To improve the generalization ability of deep networks, regularizer and batch normalization [1] and sparse deep feature learning [2] were proposed to reduce the possibility of over-fitting. Dropout [3] that randomly drops network hidden units or weights, has been also applied to many object recognition problems [4]. Motivated from the hidden unit dropout, connection (weight) dropout [5] was proposed dropout weight elements randomly. *Khan et al.* [6] proposed to perform dropout for the spectral transformation of a feature map, where three different variants corresponding to the reshaped dimension of the feature map were introduced.

The work was supported by Natural Science Foundation of China under grants no. 61602315, 61672357 and U1713214, the Science and Technology Project of Guangdong Province under grant no. 2018A050501014, the Tencent "Rhinoceros Birds"-Scientific Research Foundation for Young Teachers of Shenzhen University, the School Startup Fund of Shenzhen University under grants no. 2018063.

S. Palaiahnakote et al. (Eds.): ACPR 2019, LNCS 12046, pp. 46–58, 2020.
https://doi.org/10.1007/978-3-030-41404-7_4

However, the hidden units or weights in the traditional dropout are suppressed element by element, which may neglect the structural information implied in the element block. *Tompson et al.* [7] proposed spatial dropout to drop one entire feature map, i.e. the hidden units in one feature map are all dropped or retained simultaneously. Poernomo and Kang [8] divided the features into two groups with equal size according to the magnitudes of hidden unit responses [9], and assigned a dropout probability to each group. Meanwhile, an additional cross-map dropout [8] was proposed, where the elements at the same coordinate on different feature maps are dropped or retained simultaneously. However, two groups are not large enough to differentiate the contributions among different features, more groups should be devised. *Rohit et al.* [10] proposed the guided dropout by dropping nodes according to the strength of each node. *Zhang et al.* [11] proposed the region dropout to use the combination of the salient regions for training. However, the relative positions and sizes of the regions are fixed, which are not flexible enough. *Zhang et al.* [12] proposed grid dropout to reduce the searching space to facilitate the exploration of the global feature. However, the elements in the same grid may be significantly different from each other, the same dropout probability assigned to the entire grid may not work well for the significantly different elements in the same grid.

For the **characteristics (hidden unit, feature or weight)** grouping for dropout, the state-of-the-art dropout variants do not partition these characteristics with enough flexibility and diversity. Actually, for network back propagation, even adjacent elements in feature map and weight matrix contribute largely differently to the network loss. For example, Fig. 1 shows the active regions of the feature maps of an expression image with ResNet18 [13], where different feature maps are categorized into three different levels of importance, i.e. insignificant, fair and significant according to the heat maps response. Intuitively, the magnitude of the characteristic element response should be negatively correlated with the probability of the dropout probability. However, traditional dropout and the state-of-the-art variants can not gather these insignificant feature maps or elements distributed on an entire map for dropout. In this work, network element clustering is introduced in dropout to group the similar elements to share the same dropout probability. Thus, with the proposed clustering, the insignificant elements can be suppressed simultaneously by assigning the corresponding group with a large dropout probability.

For the dropout probability setting, the fixed dropout probability throughout the network training may neglect the dynamic influences of different parts for the network optimization. *Wager et al.* [14] treated the dropout training as a form of adaptive regularization with the approximation of second-order derivative. Ba and Frey [15] proposed a self-adaptive dropout by updating a probability mask matrix according to matrix elements' performance. In this work, the dropout probabilities are updated dynamically according to the clustering group of average characteristic response.

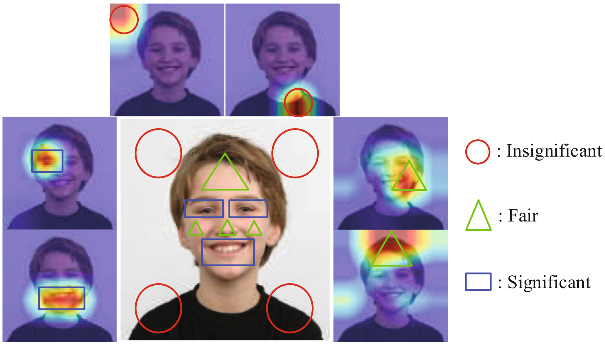


Fig. 1. Six of the 512 feature maps of an example expression in the last convolution layer of the residual network (ResNet18) [13]. According to the effect of the areas of interest on the RaFD database, the feature maps can be divided into different importance levels, i.e. insignificant, fair and significant.

To consider the characteristic for dropout, the fully connected (FC) layer features (i.e. layer input) and weight matrix in a deep network are often used as the discriminative features to determine the recognition performance. Consequently, FC features, the weights, together with their derivatives are used as the characteristics for the clustering.

The main contributions of this work are summarized as follows

- A new dropout based on the clustering of FC features, weights or their derivatives is proposed;
- Self-adaptive renewal of dropout probabilities is proposed based on the response magnitude of each group of feature, weight or derivative clustering;
- Competitive performances are achieved on the databases of Fashion-MNIST and CIFAR10, and expression databases of FER2013 and CK+.

This paper is structured into the following sections. The proposed clustering-based dropout is introduced in Sect. 2. The experimental results and the corresponding illustrations are demonstrated in Sect. 3. Finally, the conclusions and a discussion are presented in Sect. 4.

2 The Proposed Algorithm

In this section, the difference between the proposed dropout and the traditional version [3] is first illustrated, then the framework of the proposed algorithm is introduced. Finally, the related network configuration and loss function are presented.

Figure 2 shows the difference between the traditional dropout and the proposed clustering-based dropout. Compared with the traditional dropout (a) that the FC features are dropped with an uniform dropout probability, the proposed

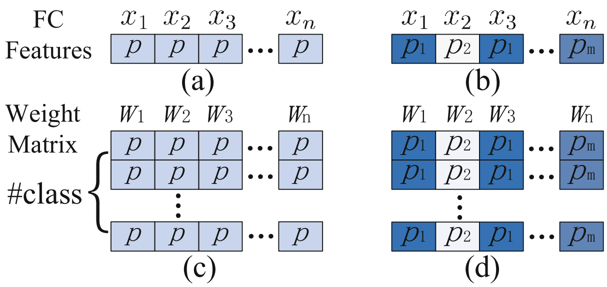


Fig. 2. The traditional dropout [3] ((a), (c)) and the proposed dropout based on clustering ((b), (d)). p, $\{p_1, ..., p_m\}$ are the assigned dropout probabilities. *#class* denotes the number of classes, $x = \{x_1, ..., x_n\}$ denotes the FC input, n is the feature dimension, $W = \{W_1, ..., W_n\}$ denotes the weight matrix.

dropout (b) takes into account the variation among different feature elements. As shown in Fig. 2(d), clustering is performed on the column vectors of a 2D weight matrix, in this way, the elements of each weight vector share the same dropout probability. Based on the network element clustering, different dropout probabilities are assigned to the corresponding groups to differentiate their different contributions.

The framework of the proposed dropout is presented in Fig. 3, where the convolution layers are followed by an average pooling layer and a FC layer, then the dropout is performed on the network characteristics, i.e. features, weights

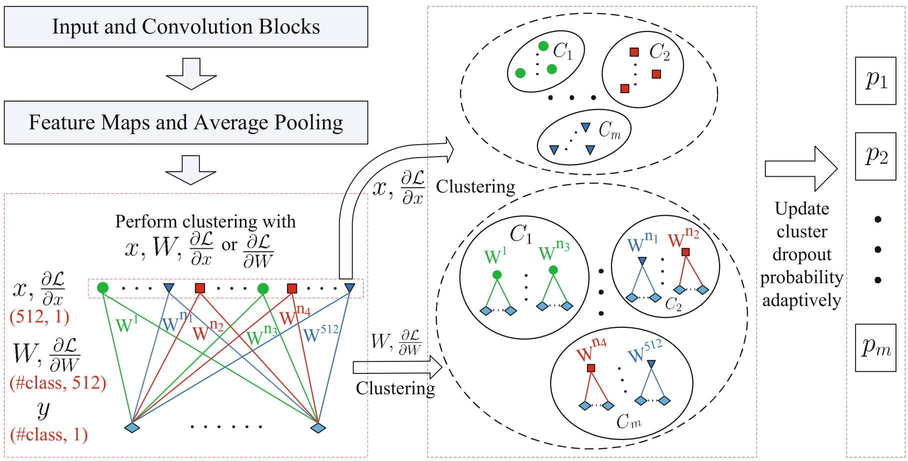

Input and Convolution Blocks
Feature Maps and Average Pooling
Perform clustering with
$x, W, \frac{\partial \mathcal{L}}{\partial x}$ or $\frac{\partial \mathcal{L}}{\partial W}$
$x, \frac{\partial \mathcal{L}}{\partial x}$
(512, 1)
$W, \frac{\partial \mathcal{L}}{\partial W}$
(#class, 512)
y
(#class, 1)
W^1
W^{n_1}
W^{n_2}
W^{n_3}
W^{n_4}
W^{512}
$x, \frac{\partial \mathcal{L}}{\partial x}$
Clustering
$W, \frac{\partial \mathcal{L}}{\partial W}$
Clustering
C_1
C_2
C_m
Update
cluster
dropout
probability
adaptively
p_1
p_2
p_m

Fig. 3. The framework of the proposed clustering-based dropout. Notations x, W, $\mathcal{L}$ and y are the features input, weight matrix, network loss and output of the FC layer, $\mathcal{L}$ is the network loss function, p_i is the dropout probability assigned to the i-th cluster C_i.

or their derivatives related with the FC layer. Finally, the dropout probabilities corresponding to the clusters are updated in a self-adaptive mode.

2.1 Clustering-Based Dropout with Adaptive Probability

For the proposed dropout, four kinds of characteristics are used for the clustering and presented as follows

- Feature vector x, i.e. the FC layer input vector;
- The weight matrix, i.e. W, between the FC layer input and output y;
- The derivatives of the loss with respect to (w.r.t.) the feature x and the weight matrix W, i.e. $\frac{\partial \mathcal{L}}{\partial x}$, $\frac{\partial \mathcal{L}}{\partial W}$;

With the clustering of the FC features, weights or their derivatives, the self-adaptive renewal algorithm of the dropout probabilities is proposed as follows.

$$\begin{cases} \gamma_i = \frac{1}{\#C_i} \sum_{t \in C_i} ||t||_1, \\ \gamma_{min} = \min_i \gamma_i, \gamma_{max} = \max_i \gamma_i, \\ tp_i = p_{min} \frac{\gamma_{max} - \gamma_i}{\gamma_{max} - \gamma_{min}} + p_{max} \frac{\gamma_i - \gamma_{min}}{\gamma_{max} - \gamma_{min}}, \\ p_i = 1 - tp_i. \end{cases} \tag{1}$$

where the user-defined parameters $p_{min} = 0.2$, $p_{max} = 0.8$ are the minimal and maximal dropout probabilities, γ_i is the average of the L_1-norm values of the i-th cluster C_i. p_i is the dropout probability assigned to the i-th cluster C_i. For feature or feature derivative, variable t is a scalar element of the average response of a batch of elements; for weight or weight derivative clustering, t denotes one of their column vector with the dimension of $\#class$.

Based on the updated dropout probabilities, the proposed dropout is performed on the employed characteristic, i.e. FC features vector, weight matrix or their derivatives. More precisely, the dropout on the features or weights is formulated as follows

$$\begin{cases} \text{Feature dropout: } \begin{cases} r_j \sim Bernoulli(p_{x_j}), \\ x \leftarrow x \star r, \end{cases} \\ \text{Weight dropout: } \begin{cases} mask_i \sim Bernoulli(p_{W_i}), \\ W \leftarrow W \star mask, \end{cases} \\ y_i = \frac{e^{W_i^T x + b_i}}{\sum_j e^{W_j^T x + b_j}}. \end{cases} \tag{2}$$

where $\star$ denotes element-wise product, W_i denotes the i-th column of W, p_{x_j} and p_{W_i} are the dropout probabilities assigned to x_j and W_i, respectively. In the network training stage, the connection weights are weighted with the probability of $1 - p_i$.

2.2 Clustering Algorithm and Network Configuration

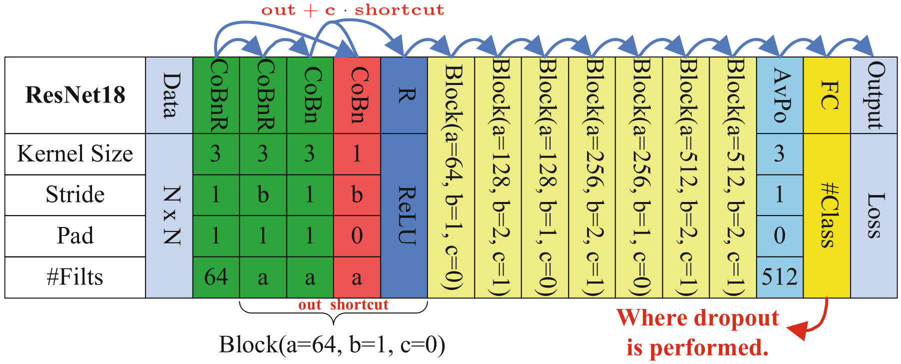
For the network element clustering, k-means algorithm is employed, which is formulated as following equations in an iterative mode

$$\begin{cases} c_i = \frac{1}{\#C_i} \sum_{t \in C_i} t, \\ l_t = argmin_{1 \le i \le m} ||c_i - t||_2^2. \end{cases} \tag{3}$$

where m is the number of clusters, c_i is the center vector of the i-th cluster C_i, and $\#C_i$ denotes the number of samples, l_t is the updated label of the sample t, variable t is defined in Eq. (1).

For the characteristics of the derivatives w.r.t. the features and weights, the similar clustering in Eq. (3) are performed by replacing x or W with $\frac{\partial \mathcal{L}}{\partial x}$ or $\frac{\partial \mathcal{L}}{\partial W}$, then the features or weights based on the results of derivative clustering are used for the dropout in Eq. (2).

The residual network (ResNet18) [13] is used for the training and evaluation. ResNet18 fits the residual mapping $\mathcal{F}$ and then appends it to the identity mapping im to estimate the output $\mathcal{H} = \mathcal{F} + im$, rather than fitting the output $\mathcal{H}$ directly. ResNet18 was reported to be able to decrease the possibility of weight gradient vanishing when the network is very deep. The configuration of the ResNet18 network is presented in Fig. 4.



ResNet18	Data	CoBnR	CoBnR	CoBn	CoBn	R	Block(a=64, b=1, c=0)	Block(a=128, b=2, c=1)	Block(a=128, b=1, c=0)	Block(a=256, b=2, c=1)	Block(a=256, b=1, c=0)	Block(a=512, b=2, c=1)	Block(a=512, b=2, c=1)	AvPo	FC	Output
Kernel Size	N x N	3	3	3	1	ReLU								3	#Class	Loss
Stride		1	b	1	b									1		
Pad		1	1	1	0									0		
#Filts		64	a	a	a									512		

Fig. 4. The network structure of ResNet18. $N \times N$ denotes the image size. *Co*, *Bn*, *R*, *AvPo* and *FC* denote the convolution, batch normalization, ReLU, average pooling and fully connected layers, respectively. *#Filts* and *#class* denote the numbers of feature maps and classes, respectively.

The cross entropy softmax function is used as the network discrimination loss, which is formulated as follows

$$\mathcal{L} = -ln \frac{e^{W_{l_i}^T x} + b_{l_i}}{\sum_j e^{W_j^T x + b_j}}, \tag{4}$$

where l_i is the label of the i-th sample. The derivatives of the loss $\mathcal{L}$ w.r.t. the feature x and weights W, i.e. $\frac{\partial \mathcal{L}}{\partial x}$, $\frac{\partial \mathcal{L}}{\partial W}$ are calculated automatically with network back propagation. For clarity, the proposed dropout is presented in Algorithm 1.

2.3 Implementation Details

We perform the experiments using four-kernel Nvidia TITAN GPU Card and the Pytorch platform. The learning rate is updated with cosine annealing and

Algorithm 1. The proposed dropout.

```
Initialize the network parameters and cluster number $m$.
for $s = 0, \cdots, MaxIter$ do
    Select a combination of the characteristics ($x$, $W$, $\frac{\partial \mathcal{L}}{\partial x}$ and $\frac{\partial \mathcal{L}}{\partial W}$) for clustering with k-means algorithm.
    Update the dropout probability of the features $x$ or weights $W$ in each cluster with equation (1) with the interval of $IntBat$ batches.
    Perform dropout on the features $x$ or weights $W$ with equation (2).
end for
Output the trained network model for testing.
```

SGD optimizer is employed. $IntBat = 1$, $m = 10$, the batch size and learning rate are 64 and 0.01, respectively.

3 Experimental Results

The experiments are performed in the following sequence. First, the employed databases are introduced; Second, various clustering parameter settings and dropout variants are evaluated on four public recognition problems; Lastly, the proposed dropout is compared to the state-of-the-art algorithms.

The Fashion-MNIST (FM.) [16] is a standard dataset of clothing, which consists of 28×28 pixels of grayscale clothing images with 60,000 training and 10,000 testing samples.

The CIFAR10 (CIF.) [17] dataset consists of 50,000 training and 10,000 testing samples, each with 32×32 pixels of RGB color. The task is to classify the images into 10 different objects.

The FER2013 (FER.) [18] database consists of 35887 grayscale face images with size 48×48, which is collected from the internet and used for a challenge. The faces were labeled with one of seven categories, i.e. angry, disgust, fear, happy, sad, surprise and neutral. The training, public test (validation) and final test (testing) sets consist of 28,709, 3,589 and 3,589 examples, respectively.

The CK+ [19] database consists of 593 expression sequences from 123 subjects, where 327 sequences are labeled with one of seven expressions, i.e. six basic and 'contempt' expressions. Five non-neutral images sampled from each expression sequence are used for testing. The person-independent strategy with ten-fold setting is employed for CK+ testing. The example samples of the four databases are presented in Fig. 5.

To evaluate different model settings in the proposed dropout, three independent trails are performed for each parameter or model setting. Table 1 presents the average recognition accuracies and their standard variances using different network characteristics. For the dropout fusing with two characteristics in the last two columns of Table 1, the clustering with each characteristic is weighted by 0.5 for the dropout probability update in Eq. (1).

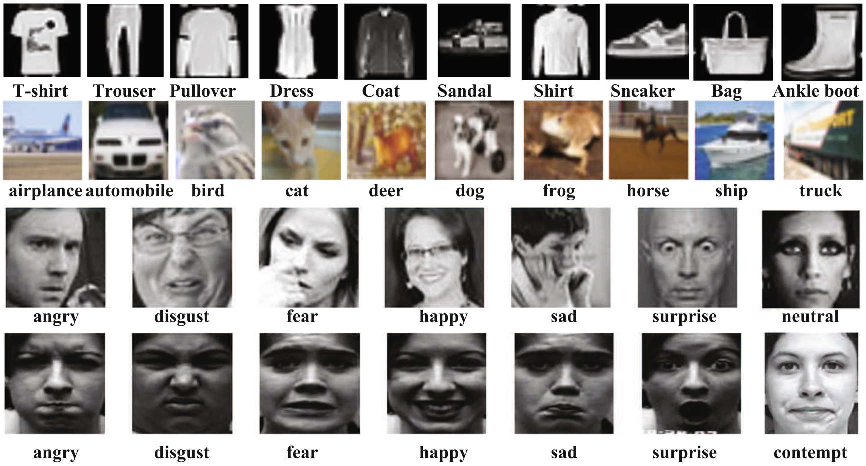


Fig. 5. Example images and the corresponding labels of the Fashion-MNIST, CIFAR10, FER2013 and CK+ databases.

Comparing the 4th and 5th (or 6th and 7th) columns in Table 1, the proposed algorithm with the clustering of FC features generally achieves better performance than that of weight matrix. Comparing the 3rd column with 4th-9th columns, one can observe that the proposed dropout significantly outperforms the original dropout with the significance level of 0.05 on the four databases, especially for FER2013 and CK+ databases, large improvements of 0.35% and

Table 1. Average recognition rates (%, even row) and their standard variances (odd row) of different parameter settings on the four databases. Notations of 'data.', 'Non' and 'Ori.' are the abbreviations of 'database', 'non dropout' and 'original dropout', respectively. For the original dropout, the best performances are achieved when the dropout probability is set as 0.5. † means that the proposed dropout is significantly better than the original dropout with Student's t-Test [20] under the significance level of 0.05.

Dat.	Non	Ori.	x	W	$\frac{\partial L}{\partial x}$	$\frac{\partial L}{\partial W}$	$x+\frac{\partial L}{\partial x}$	$x+W$
FM.	94.08	94.03	$94.27^\dagger$	$\mathbf{94.28}^\dagger$	94.21	94.16	$94.22^\dagger$	$94.25^\dagger$
	0.08	0.03	0.01	0.06	0.12	0.09	0.08	0.05
CIF.	94.5	94.48	$94.72^\dagger$	94.62	$94.65^\dagger$	$94.59^\dagger$	$\mathbf{94.82}^\dagger$	$94.68^\dagger$
	0.09	0.11	0.09	0.09	0.06	0.01	0.1	0.03
FER.	72.68	72.87	73.08	73.08	72.95	$73.19^\dagger$	73.13	$\mathbf{73.22}^\dagger$
	0.13	0.15	0.36	0.05	0.23	0.03	0.3	0.1
CK+	96.64	96.53	$\mathbf{98.37}^\dagger$	$97.76^\dagger$	$98.17^\dagger$	$97.55^\dagger$	$98.16^\dagger$	$98.27^\dagger$
	0.01	0.35	0.14	0.29	0.23	0.25	0.25	0.14

1.84% are achieved. Meanwhile, the fusion of multiple clustering results can balance the performances with single ones on the four databases, which achieved the best performances on the Cifar10 and FER2013 databases.

Table 2. Average recognition rates (%, even row) and their standard variances (%, odd row) of the dropout variants of spatial dropout (S.D.) [7], cross map (C.M.) dropout [8], biased dropout (B.D.) [8], feature x dropout with two-cluster clustering (x 2-c), and our feature dropout with $m = 10$ clusters.

Dat.	S.D.	C.M.	B.D.	x 2-c	x	$x + \frac{\partial L}{\partial x}$	$x + W$
FM.	94.03	94.04	94.12	94.17	**94.27**	94.22	94.25
	0.03	0.09	0.08	0.15	0.01	0.08	0.05
CIF.	94.58	94.63	94.48	94.69	94.72	**94.82**	94.68
	0.07	0.04	0.05	0.02	0.09	0.1	0.03
FER.	72.92	72.95	72.74	72.92	73.08	73.13	**73.22**
	0.19	0.25	0.11	0.14	0.36	0.3	0.1
CK+	97.15	97.35	97.25	97.96	**98.37**	98.16	98.27
	0.64	0.18	0.53	0.38	0.14	0.25	0.14

To compare the proposed dropout with other dropout variants, Table 2 shows the accuracies and their standard variances of five dropout variants. For the dropout probability update with two groups [8], biased dropout (B.D.) and 2-cluster clustering (x 2-c), the characteristic of FC features are employed.

Comparing the 4th and the 5th columns, Table 2 shows that feature clustering outperforms the feature equipartition [8] on the four databases, which illustrates the effectiveness of the clustering employed in the proposed dropout. When the same clustering is employed, 10-cluster setting (6th column) still performs better than 2-clusters (5th column) on the four databases, which reveals that 10-cluster clustering matches the variations of FC features better than 2-cluster clustering.

To study the variations of the dropout probabilities with the proposed dropout, Fig. 6 presents the numbers of FC features elements in 10 clusters with different iteration epochs. One can observe that the elements with large response values after training account for a small proportion of the number of the entire FC features. More precisely, the feature elements are mostly concentrated in the cluster with small response value. This observation is similar to that of the network compression [21] and L_2-normalization on FC feature for generalization ability improvement. Meanwhile, large difference among the numbers of feature neurons in different clusters is observed, which implies the diverse contributions of different feature groups for network training. By taking into account this difference, the proposed dropout with the feature clustering can better differentiate the feature contribution than the original dropout during network training.

Regarding to the runtime of the proposed algorithm, the time complexity of the k-means algorithm for FC features is $O(n)$ (n is the feature dimension),

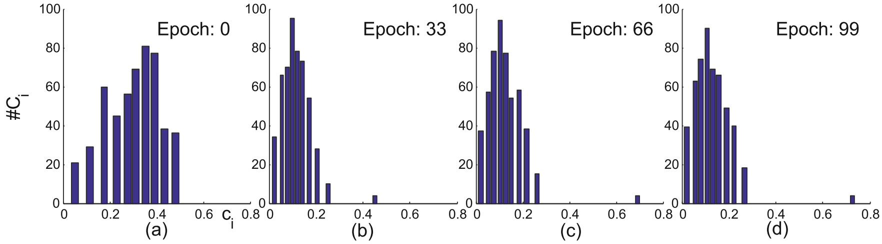


Fig. 6. The number of feature elements of 10 clusters with different iteration epochs on the FER2013 databases, where the FC feature average is used to label the corresponding cluster.

which is almost negligible compared with the network training. For the clustering with weights or weight derivatives, the runtime of the k-means is $O(n \cdot \#class)$. To reduce the runtime cost of the weight clustering in the proposed training, the clustering is performed periodically after a interval of $IntBat$ batches. Meanwhile, the testing performances of the proposed algorithm against the number of interval batches, i.e. $IntBat$ are presented in Fig. 7. Figure 7 shows that a even better performance can be achieved by the proposed algorithm with the fine tuning of the number of interval batches, i.e. $IntBat = 10$. Thus, a slightly large number of interval batches help clustering not only save the runtime cost, but also learn more stable information to contribute to the performance improvement.

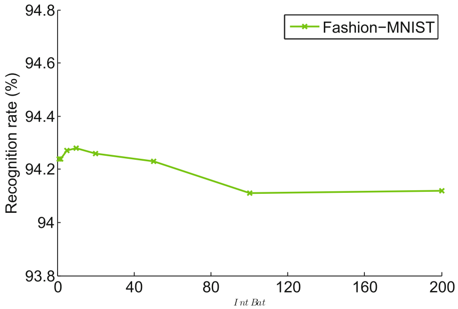


Fig. 7. The performances of the proposed algorithm with different setting of the interval batches ($IntBat$) on the Fashion-MNIST database.

To compare the performance of the proposed algorithm with other state-of-the-art approaches, Tables 3 compares the testing recognition rates of the proposed algorithm with those reported in eight state-of-the-art works on the four databases, where 10-cluster clustering of feature and feature derivative is employed in the proposed dropout. For the CK+ database, the works using the same seven expressions as this paper are included for the comparison.

Table 3. Comparison of accuracies (Acc.) of different algorithms (Algor.) on FM., CIF., FER. and CK+ databases.

FM.		CIF.		FER.		CK+	
Algor.	Acc.	Algor.	Acc.	Algor.	Acc.	Algor.	Acc.
CBD [8]	92.03	CBD [8]	82.45	*Mollahosseini* [22]	66.4	*Jung* [23]	97.25
DS [24]	–	DS [24]	88.1	DS [24]	71.32	*Liu* [25]	97.1
BC [26]	–	BC [26]	91.73	*Wen* [27]	69.96	–	–
GD [10]	–	GD [10]	94.12	–	–	–	–
Ours	**94.27**	Ours	**94.82**	Ours	**73.22**	Ours	**98.37**

Tables 3 show that the proposed algorithm achieves the consistently best performances on the four databases among the state-of-the-art algorithms, where large improvements of 2.24%, 0.7%, 1.9% and 1.12% are achieved by the proposed algorithm on the Fashion-MNIST, CIFAR10, FER2013 and CK+ databases, respectively. The competitive performances verify the effectiveness of the proposed dropout with the clustering of FC feature and weight matrix.

4 Conclusion

To take into account that the elements in the fully connected (FC) features, weights, derivatives of features and weights contribute differently to the network optimization, a clustering-based dropout with self-adaptive dropout probability is proposed. The proposed dropout is further embedded into the FC layer of ResNet18 for four public databases, i.e. Fashion-MNIST, CIFAR10, FER2013 and CK+, the experimental results verify the competitiveness of the proposed dropout compared with other dropout variants and the related state-of-the-art algorithms.

Although competitive results are achieved by the proposed clustering-based dropout, there remains room for further improvement. First, the influences of introduced hyper-parameters on the network learning, such as the number of clusters, should be further explored. Second, the theoretical analysis of the clustering-based dropout with different model selections should be deeply studied. Lastly, the proposed dropout should be applied in more models and tasks.

References

1. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 730–734 (2015)
2. Xie, W., Jia, X., Shen, L., Yang, M.: Sparse deep feature learning for facial expression recognition. Pattern Recogn. (PR) **96**, 106966 (2019)

3. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. (JMLR) **15**, 1929–1958 (2014)
4. Wang, H., Wang, L.: Learning robust representations using recurrent neural networks for skeleton based action classification and detection. In: International Conference on Multimedia Expo Workshops (ICMEW), pp. 591–596, July 2017
5. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using DropConnect. In: Proceedings of International Conference on Machine Learning (ICML), vol. 28, pp. 1058–1066, June 2013
6. Khan, S., Hayat, M., Porikli, F.: Regularization of deep neural networks with spectral dropout. Neural Netw. **110**, 82–90 (2019)
7. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 648–656 (2014)
8. Poernomo, A., Kang, D.K.: Biased dropout and crossmap dropout: learning towards effective dropout regularization in convolutional neural network. Neural Netw. **104**, 60–67 (2018)
9. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems (NIPS), pp. 1135–1143 (2015)
10. Rohit, K., Richa, S., Mayank, V.: Guided dropout. In: AAAI Conference on Artificial Intelligence (AAAI) (2019)
11. Zhang, X., Yang, Y., Feng, J.: ML-LocNet: improving object localization with multi-view learning network. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 248–263. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_15
12. Zhang, C., Zhu, C., Xiao, J., Xu, X., Liu, Y.: Image ordinal classification and understanding: grid dropout with masking label. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6, July 2018
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
14. Wager, S., Wang, S., Liang, P.S.: Dropout training as adaptive regularization. In: Advances in Neural Information Processing Systems (NIPS), pp. 351–359 (2013)
15. Ba, J., Frey, B.: Adaptive dropout for training deep neural networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 3084–3092 (2013)
16. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
17. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)
18. Goodfellow, I.J., et al.: Challenges in representation learning: a report on three machine learning contests. In: Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M. (eds.) ICONIP 2013. LNCS, vol. 8228, pp. 117–124. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42051-1_16
19. Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG), pp. 46–53 (2000)
20. De Winter, J.C.: Using the student's t-test with extremely small sample sizes. Pract. Assess. Res. Eval. **18**(10), 1–12 (2013)

21. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv preprint arXiv:1510.00149 (2015)
22. Mollahosseini, A., Chan, D., Mahoor, M.H.: Going deeper in facial expression recognition using deep neural networks. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–10 (2016)
23. Jung, H., Lee, S., Yim, J., Park, S., Kim, J.: Joint fine-tuning in deep neural networks for facial expression recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2983–2991 (2015)
24. Tang, Y.: Deep learning using support vector machines. In: Proceedings of International Conference on Machine Learning (ICML) (2013)
25. Liu, X., Vijaya Kumar, B., You, J., Jia, P.: Adaptive deep metric learning for identity-aware facial expression recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 20–29 (2017)
26. Courbariaux, M., Bengio, Y., David, J.P.: BinaryConnect: training deep neural networks with binary weights during propagations. In: Advances in Neural Information Processing Systems (NIPS), pp. 3123–3131 (2015)
27. Wen, G., Hou, Z., Li, H., Li, D., Jiang, L., Xun, E.: Ensemble of deep neural networks with probability-based fusion for facial expression recognition. Cogn. Comput. **9**(5), 597–610 (2017)