Signal Processing: Image Communication 60 (2018) 193–198

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Convolutional neural network with nonlinear competitive units

Zhang-Ling Chen [a], Jun Wang [b,*], Wen-Juan Li [c], Nan Li [a], Hua-Ming Wu [a], Da-Wei Wang [a]

[a] Center for Applied Mathematics, Tianjin University, Tianjin 300072, P.R. China
[b] School of Mathematics, Tianjin University, Tianjin 300072, P.R. China
[c] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P.R. China

ARTICLE INFO

Keywords:
Nonlinear competitive unit
Feature fusion
Activation function
Face verification
Visual classification

ABSTRACT

Convolutional Neural Network (CNN) has been an important breakthrough in pattern recognition in recent years. Nevertheless, with the increase in complexity, CNN becomes more difficult to train. To alleviate the problem of training difficulties, we propose a novel nonlinear unit, called Nonlinear Competitive Unit (NCU). By comparing the elements from different network layers and selecting the larger signals element-wisely, it can not only strengthen feature propagation but also accelerate the convergence of CNN. This unit can be regarded as a feature fusion method as well as a kind of activation function. We evaluate our NCU-based models for face verification task and visual classification task on four benchmark datasets. The experimental results demonstrate the superior performance of our models over many state-of-the-art methods, which shows the advantage and potential of the NCU in networks.

1. Introduction

Since AlexNet [1] was introduced by Hinton in 2012, convolutional neural network (CNN) has received extensive attention. A number of vision tasks, such as image classification [2,3], face recognition [4,5] and face verification [6], have benefited from the robust and discriminative representation learnt via CNN models [7–9,5,10–12]. Compared with traditional feature extraction methods [13,14], CNN models mainly take advantage of the large-scale training data and the end to end learning framework. In recent years, a large number of skills have been introduced to promote the performance of CNN. They are mainly along two directions: (a) presenting effective training strategies, for example, the well-designed initialization strategies [2] and effective regularization techniques [15]; (b) creating more effective models, such as increasing the depth of the CNNs [6,11,12], expanding the width [16] and the use of nonlinear activation functions [17–19].

Depth and width are two crucial components for the diversification of network architecture. Nevertheless, training deep or wide networks would also suffer several difficulties, such as time consuming and exploding/vanishing gradient or degradation. Actually, it needs lots of effects to learn a deep hierarchical structure effectively and efficiently.

In some works, instead of drawing the representational power from extremely deep or wide architecture, some researchers try their best to explore different connectivity patterns. GoogLeNet [8] concatenates the outputs of several subnetworks with different length, which can be seen as a feature fusion method. Highway Networks [20] provide a means, the shortcut connection along with gating functions [21], to effectively train CNN models with more than 100 layers. Indeed, shortcut connection [22], a feature fusion method, has been studied for a long time in many works [22,23]. Residual Networks (ResNets) further support the point that shortcut connection is a key factor that eases the training of deep CNN models. Instead of excepting each stacked layers directly fit a desired underlying mapping, ResNets explicitly generate the residual mapping.

In this paper, we propose a new nonlinear unit, named Nonlinear Competitive Unit (NCU). It can be regarded as a feature fusion method or an activation function. As a feature fusion method, the NCU aims at competing the intermediate representations of subnetworks, where the fused output serves as the input of the remaining part of network. From the view of activation function, the most popular activation function is ReLU [17], which keeps the identity of positive elements and zeros otherwise. Although it overcomes the problem of vanishing gradient, it might loss some information, especially in the first several layers. In the NCU, the activation threshold is learnt, which means one of the competitive representations can act as the base threshold. Compared with ReLU, the NCU's final result holds the reasonable activation signals but with a certain of sparsity lost. Another similar unit– the maxout [24]

* Corresponding author.
E-mail address: wangjun265220@sohu.com (J. Wang).

https://doi.org/10.1016/j.image.2017.09.011
Received 2 March 2017; Received in revised form 25 August 2017; Accepted 28 September 2017
Available online 21 October 2017
0923-5965/

can also hold a good fitting ability by adaptively adjusting the activation threshold. The difference between the NCU and maxout are:

- maxout feature map is constructed by taking the maximum across k affine feature map with a large number of parameters, while our NCU is parameter-free.
- maxout is used in conjunction with dropout, and the NCU can be used independently.

The main contributions of this paper are three fold:

1. A novel nonlinear unit is proposed, named Nonlinear Competitive Unit, which can be regarded as a feature fusion method as well as an activation function.
2. Compared with the benchmark network models, the convergence speed of our model is improved accompanied with higher stability.
3. The experiments validate that NCU-based models can effectively boost the performance in both face verification task and visual classification task.

The remainder of the paper is organized as follows. In Section 2, we first give a brief introduction of residual block, then present a clear definition of the NCU and study a competitive structure, i.e. competitive block. We provide the experimental setups and results in Section 3. Finally, the paper is concluded in Section 4.

2. Nonlinear competitive unit

The intuitive inspiration for our nonlinear competitive unit comes from the residual block of ResNets [12]. In this section, the residual block, the proposed nonlinear competitive unit and the corresponding competitive block will be illustrated.

2.1. Residual block

ResNets improve the classification performance significantly by constructing many stacked residual blocks. Here, set the input of a residual block as $\mathbf{x}$, the basic mapping as $H(\mathbf{x})$, and the output $F(\mathbf{x})$ satisfies the following equation:

$$F(\mathbf{x}) := H(\mathbf{x}) + \mathbf{x}, \tag{1}$$

the detailed operation about kth residual block in ResNet can be written as follows:

$$\mathbf{x}_{k+1} = h(f(\mathbf{x}_k, \mathbf{W}_{ki}) + \mathbf{W}_s\mathbf{x}_k), \qquad i \geq 1, \tag{2}$$

where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are the input and output of the kth residual block, respectively; h is the activation function ReLU, which is of great essential to the successes of CNN models; the function f represents the residual mapping; $\mathbf{W}_{ki}$ is the weight parameter associated with the kth residual block, where parameter i represents the number of weight layer that commonly designed as 2 or 3; $\mathbf{W}_s$ is a linear projection by the shortcut connections to match the dimension.

Especially, when $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ share the same size, the shortcut connection simply performs identity mapping, which adds neither extra parameter nor computational complexity. In [25], it shows that not only the use of identity mapping is sufficient to address the degradation problem but also achieve the fastest error reduction and lowest training loss among all variants, and thus we do not use $\mathbf{W}_s$ when the dimensions of $\mathbf{x}_k$ and f are equal. Without loss of generality, the activation functions Batch Normalization(BN) [19] and ReLU are inserted immediately after the weight layer as post-activation. In contrast to this conventional wisdom, BN and ReLU are put in front of weight layer in [25], called pre-activation.

The architecture of residual block about the number of weight layers and the size of convolutional kernel size have been discussed in [26]. Fig. 1 shows the rough structure of basic block B(3,3) — residual block with two consecutive 3 × 3 convolutions.

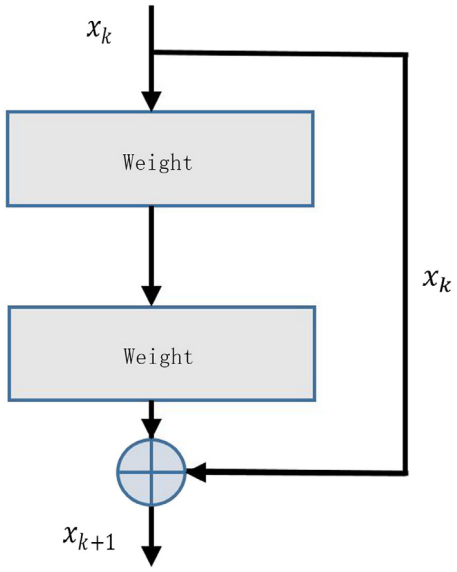


Fig. 1. Residual Block. Longer thick arrow indicates a more direct way for propagating information, which adds neither extra parameter nor computational complexity. Two weight layers represent two convolution operations.

2.2. Nonlinear competitive unit

In this part, we propose our Nonlinear Competitive Unit (NCU), which is designed to reformulate the output as a competitive result of the inputs specifically. For two inputs $I_1, I_2 \in R^{p\times q\times n}$, the output $O \in R^{p\times q\times n}$ satisfies the equation:

$$O = I_1 \bigodot I_2, \tag{3}$$

where $\bigodot$ can be described as follows:

$$O^k(i,j) := \max(I_1^k(i,j), I_2^k(i,j)) \tag{4}$$

where $1 \leq k \leq n$, $1 \leq i \leq p$, $1 \leq j \leq q$. According to Eq. (4), we can obtain a representation with the same size of the inputs, while the elements are the larger ones between two inputs. For a specified k, the structure of the NCU can be showed in Fig. 2(a).

As described above, the NCU can be considered as not only a feature fusion method, but also an activation function. And the fusion result or activate value is the competitive winner, i.e. the larger element in the same location.

To verify its advantage in extracting discriminative representation, we implement a specific framework of the NCU called competitive block in the network, as shown in Fig. 2(b). Similar to residual block, the competitive block can be given in a general form:

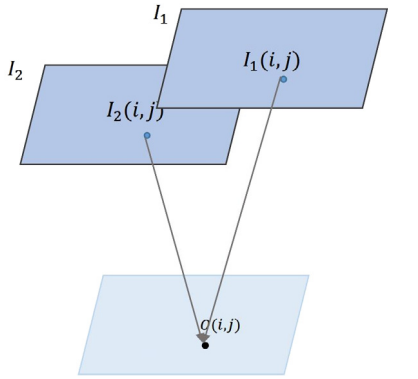
$$\mathbf{x}_{k+1} = h(f(\mathbf{x}_k, \mathbf{W}_{ki}) \bigodot \mathbf{W}_s\mathbf{x}_k), \qquad i \geq 1 \tag{5}$$

where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are the input and the output of kth competitive block.

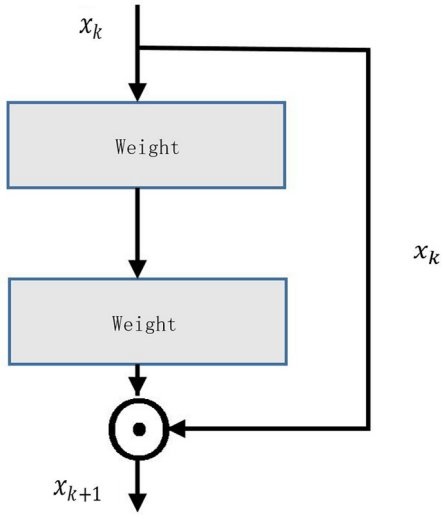
It can be found that the structure of competitive block is similar to residual block except the operation on layer's output. Actually, residual block learns residual mapping while competitive block learns competitive mapping.

3. Experiments

We evaluate the performance of the NCU in two typical vision applications: face verification and visual classification. In face verification, we evaluate our NCU-based models on two widely used datasets, i.e. Labeled Faces in the Wild (LFW) [27] and YouTube Faces (YTF) [28]. In visual classification, we use the standard benchmark datasets: MNIST [29] and CIFAR-10 [30].



(a) NCU.



(b) Competitive block.

Fig. 2. The structure of the NCU and the competitive block. (a) NCU. $I_1(i,j)$ and $I_2(i,j)$ come from different feature layers, respectively. $O(i,j)$ is the corresponding output. (b) Competitive block. The competitive framework of the NCU.

3.1. Datasets

- **CASIA-WebFace** contains about 500,000 images of 10,575 subjects and all face images are proposed by face detection, face landmarking and alignment.
- **LFW** provides a set of labeled face images spanning the range of conditions typically encountered by people in daily life, and it contains 13,233 web-collected images from 5749 different identities. The dataset has been organized into two "Views" [27], we select View 2 to evaluate the performance of our models, and use the training part of View 1 to learn covariance matrix for PCA.
- **YTF** is a video dataset. The quality of images is much worse than web photos due to motion blur and high compression ratio. In experiments, we follow the unrestricted labeled outside data protocol and report the results on 5000 video pairs.
- **MNIST** has a training set of 60k examples and a test set of 10k examples. The digits have been size-normalized and centered as a 28×28 image.
- **CIFAR-10** consists of 60k 32×32 color images from 10 classes, with 6k images per class. The dataset is divided into five training batches and one test batch, each with 10k images. The test batch contains exactly 1k randomly selected images from each class, and the training batches contain the remaining images in random order.

3.2. Face verification

For face verification task, the protocol we use to build the training dataset (CASIA-WebFace) is:

- all face images are aligned by the proposed algorithm [31] and tailored into 256×256;
- the false positive images (i.e. images not properly clipped) are discarded conservatively;
- in order to data augmentation, each tailored image is flipped horizontally, then the images are randomly cropped from the original image and the flipped one, rather than heavy data augmentation as presented in [32].

In the testing stage, all the images in LFW and YTF datasets are dealt with the same protocol as the training dataset.

3.2.1. Implementation details

The hyperparameters' setup and the optimal configuration are described in this part.

Unless explicitly stated, otherwise models are trained using the stochastic gradient descent algorithm [33] with a mini-batch size of 100. The learning rate is initialized with 0.05 and multiplied by 0.5 after every 20,000 iterations. The momentum and weight decay are designed as 0.9 and 0.0001, respectively. The weights in networks are initialized from zero-mean gaussian distribution while the bias terms are initialized with zero. We implement our system on widely used Caffe infrastructure [34].

The structure of our baseline (BaseNet) is similar with the 18-layer ResNet with pre-activation residual blocks. Specifically, we incorporate an additional fully-connected layer to map a representation space of dimension 1024 before the last softmax regression classifier. Comp-m is the network that replaces the last m residual block of baseline into competitive block, and retains the number of parameters and the layers. The architectures of networks are shown in Table 1. The dimension of input is $224 \times 224 \times 3$. There is an initial convolutional layer followed by a 3×3 max-pooling and 4 blocks conv-k ($k = 2,3,4,5$). The size of filters in each block is 3×3. In face verification stage, the deep representations are taken from the output of the first inner product layer. And we choose cosine distance as the discriminant standard.

3.2.2. Test on LFW and YTF

In this part, the experiments are conducted as following methods:

- **A: DR + Cosine**
- **B: DR + PCA1 + Cosine**
- **C: DR + PCA2 + Cosine.**

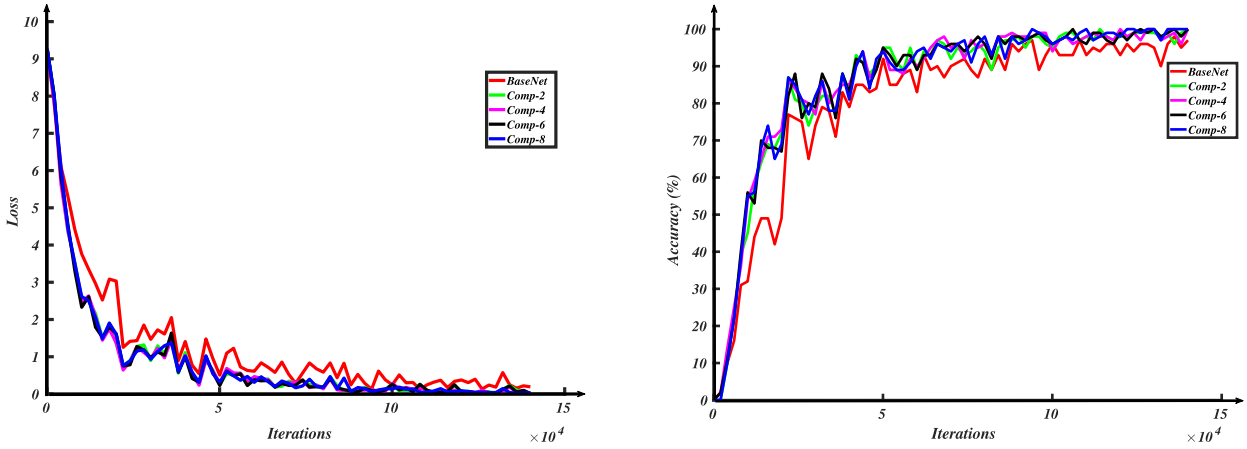
DR is an abbreviation of deep representation extracted by deep models. **PCA1** means covariance matrix directly learned by extracted features. **PCA2** means that covariance matrix is learned by the View1's training part of LFW in the first step, and then the performance is evaluated on View 2. Here we set the ratio of PCA as 0.98.

Fig. 3 shows convergence curves of BaseNet, Comp-2, Comp-4, Comp-6 and Comp-8 networks. We can find that the networks with competitive block have lower training error and higher accuracy accompanied by a faster and more stable convergence. During the training process, the training speeds of BaseNet, Comp-2, Comp-4, Comp-6 and Comp-8 are recorded, respectively, which is 25.2 s, 17.8 s, 19.2 s, 20.0 s, 21.8 s for one training iteration, indicating that competitive block is easier to be optimized than residual block.

We investigate the performance related to k, which represents the number of competitive block in networks. Here we select the representative structures Comp-k ($k = 2,4,6,8$). More intuitively, the verification accuracies of these models on LFW are illustrated in the form of bar graphs in Fig. 4. The horizontal axis shows different feature extraction methods. The blue bars represent the verification results of the BaseNet.

Table 1
The architecture of networks. $B(3,3)\times 2$ denotes two cascaded residual block, and $\mathbf{C(3,3)\times 2}$ denotes two cascaded competitive block.

Layer name	Output size	BaseNet	Comp-2	Comp-4	Comp-6	Comp-8
conv1	109×109			$7\times 7, 64$, $stride 2$		
max pool	55×55			$3\times 3, stride 2$		
conv-2	55×55	$B(3,3)\times 2, 64$	$B(3,3)\times 2, 64$	$B(3,3)\times 2, 64$	$B(3,3)\times 2, 64$	$\mathbf{C(3,3)\times 2, 64}$
conv-3	28×28	$B(3,3)\times 2, 128$	$B(3,3)\times 2, 128$	$B(3,3)\times 2, 128$	$\mathbf{C(3,3)\times 2, 128}$	$\mathbf{C(3,3)\times 2, 128}$
conv-4	14×14	$B(3,3)\times 2, 256$	$B(3,3)\times 2, 256$	$\mathbf{C(3,3)\times 2, 256}$	$\mathbf{C(3,3)\times 2, 256}$	$\mathbf{C(3,3)\times 2, 256}$
conv-5	7×7	$B(3,3)\times 2, 512$	$\mathbf{C(3,3)\times 2, 512}$	$\mathbf{C(3,3)\times 2, 512}$	$\mathbf{C(3,3)\times 2, 512}$	$\mathbf{C(3,3)\times 2, 512}$
	1×1	$avg-pooling$, $1024-fc$, $softmax$				



(a) Training loss. (b) Training accuracy.

Fig. 3. Training on CASIA-Webface. (a) The training error change along the iterations. (b) The change of training accuracies.

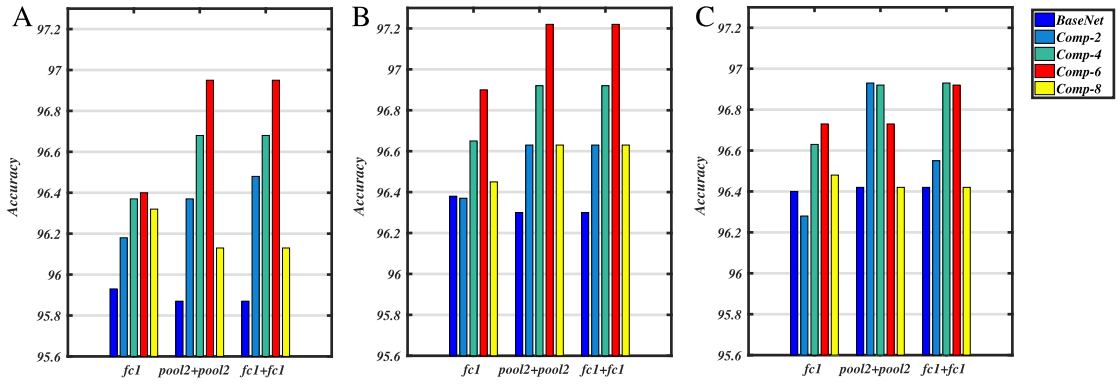


Fig. 4. Face verification accuracies on LFW dataset. The horizontal axis shows different feature extraction methods. Particularly, "fc1" represents the output features of the first inner product layer. "pool2" is the input feature of the layer "fc1". "fc1 + fc1" refers to the sum of the feature "fc1" and its corresponding horizontal mirror element-wisely.

From the verification accuracies on LFW, the following conclusions can be obtained: (a) For each test methods (**A**, **B** or **C**), the NCU-based models always achieve the highest accuracy. For method **B**, Comp-6 achieves an accuracy rate of 97.22%, about 0.9 percent higher than BaseNet. The accuracy of each NCU-based models hold higher test accuracies than BaseNet for method **A**; (b) As noticed, $k = 6$ turns out to be the best performance. It is probably dues to the higher level feature competition in network layer has a better capacity to represent the variations of the complex face images.

In the following experiments, the network models are trained with the joint supervision of softmax and center loss [35]. The experimental results are shown in Table 2. CompNet-4 represents the network that replaces the last 4 residual blocks of ResNet18 (18 layer ResNet with an additional 1024 fully-connected layer) as competitive blocks. CenterNet is the network architecture mentioned in [35], keeping the same training details described in that article. Center-k represents the network that replaces the residual block after the last kth pool layer of CenterNet as competitive block. By training on CASIA-WebFace, the NCU-based models hold higher accuracy in most test standards, especially, Center-3

Table 2
The performance on LFW and YTF with joint supervision of softmax and center loss.

Datasets	LFW			YTF	
Models	Methods				
	A	B	C	A	B
ResNet18	96.97%	97.12%	97.3%	89.04%	91.44%
CompNet-4	97.1%	97.18%	97.32%	89.7%	91.5%
CenterNet	98.28%	98.33%	98.35%	91.22%	93.0%
Center-1	98.22%	98.47%	98.55%	91.40%	92.8%
Center-2	98.38%	98.45%	98.55%	91.38%	93.04%
Center-3	98.30%	98.75%	**98.78%**	**91.56%**	**93.22%**

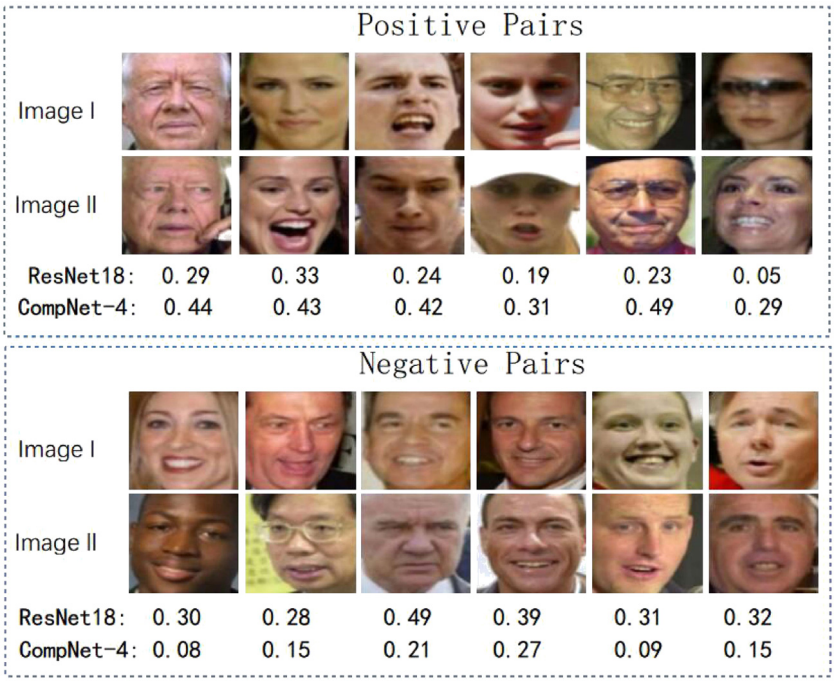


Fig. 5. Face verification on LFW. In positive pairs, the two images (Image I and II) are from the same person, but in negative paris, they are from different persons.

Table 3
The architecture of networks. $(5,20)_{/1,0}$ denotes the convolutional layer with 20 filters of size 5×5, where the stride and padding are 1 and 0, respectively. $max(avg) - 2_{2,0}$ denotes the max(avg)-pooling layers with grid of 2×2, where the stride and padding are 2 and 0, respectively.

Layer	conv	pool	conv	pool	conv	pool	NCU	FC
LeNet	$(5,20)_{/1,0}$	$max-2_{2,0}$	$(5,50)_{1,0}$	$max-2_{/2,0}$	–	–	No	fc-500/fc1-10
LeNet(avg)	$(5,20)_{/1,0}$	$max-2_{2,0}$	$(5,50)_{1,0}$	$avg-2_{/2,0}$	–	–	No	fc-500/fc1-10
LeNet-NCU	$(5,20)_{/1,0}$	$max-2_{2,0}$	$(5,50)_{1,0}$	$max, avg-2_{/2,0}$	–	–	Yes	fc-500/fc1-10
Cifar-Net	$(5,32)_{/2,1}$	$max-3_{2,0}$	$(5,32)_{2,1}$	$avg-3_{/2,0}$	$(5,64)_{/2,1}$	$avg-3_{2,0}$	No	fc-10
Cifar-Net(max)	$(5,32)_{/2,1}$	$max-3_{2,0}$	$(5,32)_{2,1}$	$avg-3_{/2,0}$	$(5,64)_{/2,1}$	$max-3_{2,0}$	No	fc-10
Cifar-Net_NCU	$(5,32)_{/2,1}$	$max-3_{2,0}$	$(5,32)_{2,1}$	$avg-3_{/2,0}$	$(5,64)_{/2,1}$	$avg, max-3_{2,0}$	Yes	fc-10

Table 4
Classification accuracies on MNIST. LeNet(avg) denotes that the last max-pooling in LeNet is replaced by average pooling. LeNet(norm) represents norm and relu are added immediately after the convolutional layer.

Method	Accuracy
LeNet	98.92%
LeNet(avg)	98.79%
LeNet_NCU	99.05%
LeNet(norm)	98.88%
LeNet(avg + norm)	99.13%
LeNet_NCU(norm)	**99.24**%

shares a verification accuracy of 98.78% on LFW and 93.22% on YTF. We owe the superiority of our network to competitive block which extracts the last competitive representations.

In theory, features with better generalization would lead to a less gap between the verification scores in positive pairs and a larger gap between negative pairs. Fig. 5 shows the scores of ResNet18 and CompNet-4 in face verification task. Compared with the ResNet18, CompNet-4 obtains higher scores in positive pairs, and lower scores in negative pairs, which proves the effective of our model in face verification. Indeed, for CompNet-4, 0.28 can be selected as an appropriate threshold, which is greater than the threshold for positive pairs and less than the threshold for negative pairs. However there is no reasonable threshold for ResNet18 that could separate the positive and negative pairs very well.

3.3. Visual classification

3.3.1. Implementation details

For experiments, we set the batch size as 100. The weight decay is 0.0005 and 0.004, respectively for MNIST and CIFAT-10 datasets. For MNIST, we start with a learning rate of 0.001, the change policy holds "inv". The gamma and power are designed as 0.0001 and 0.075, and eventually terminate training at 20k iterations. For CIFAR-10, the learning rate is fixed at 0.001, and eventually terminate training at 60k.

3.3.2. Test on MNIST and CIFAR-10

The network architectures are shown in Table 3, LeNet(avg) denotes the last max-pooling in LeNet is replaced by avg-pooling, and Cifar-Net(max) denotes the last avg-pooling in Cifar-Net is replaced by max-pooling. In NCU-based networks, avg-pooling and max-pooling are compared to strengthen feature propagation. Table 4 shows the results of the original and our NCU-based models on MNIST. From the results, NCU-based models outperform the original network obviously, which validates the effectiveness of the competition among signals. The results on CIFAR-10 shown in Table 5 further illustrate the generalization of our NCU-based models.

4. Conclusion

In this paper, we proposed a new nonlinear unit named NCU, which could be regarded as a method of feature fusion as well as a crucial activation function. It strengthened the feature propagation by comparing the elements from different network layers and selecting the

Table 5
Classification accuracies on CIFAR-10 datasets.

Method	Accuracy
Cifar-Net	78.70%
Cifar-Net(max)	75.00%
Cifar-Net_NCU	**78.97**%

larger signals element-wisely. Experimental results demonstrated that by strengthening feature propagation in networks, NCU-based models effectively boosted the performance in both face verification task and visual classification task.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[3] L. Wan, M. Zeiler, S. Zhang, Y.L. Cun, R. Fergus, Regularization of neural networks using dropconnect, in: Proceedings of the 30th International Conference on Machine Learning, ICML-13, 2013, pp. 1058–1066.

[4] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, 2014. arXiv preprint arXiv:1411.7923.

[5] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891–1898.

[6] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.

[7] C.C. Pham, J.W. Jeon, Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks, Signal Process., Image Commun. 53 (2017) 110–122.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[9] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[10] R.K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, in: Advances in Neural Information Processing Systems, 2015, pp. 2377–2385.

[11] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. arXiv preprint arXiv:1409.1556.

[12] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[13] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local binary patterns, in: European Conference on Computer Vision, Springer, 2004, pp. 469–481.

[14] P. Viola, M. Jones, et al., Robust real-time object detection, Int. J. Comput. Vis. 57 (2) (2001) 137–154.

[15] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[16] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks,, in: International Conference on Learning Representations (ICLR2014), CBLS, April 2014, 2014.

[17] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning, ICML-10, 2010, pp. 807–814.

[18] R.K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, J. Schmidhuber, Compete to compute, in: Advances In Neural Information Processing Systems, 2013, pp. 2310–2318.

[19] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[20] R.K. Srivastava, K. Greff, J. Schmidhuber, Highway networks, 2015. arXiv preprint arXiv:1505.00387.

[21] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[22] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[23] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, 2007.

[24] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A.C. Courville, Y. Bengio, Maxout networks, in: ICML (3) 28, 2013, pp. 1319–1327.

[25] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, 2016, pp. 630–645.

[26] S. Zagoruyko, N. Komodakis, Wide residual networks, 2016. arXiv preprint arXiv:1605.07146.

[27] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, Tech. rep., Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[28] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 529–534.

[29] Y. LeCun, C. Cortes, C.J. Burges, The MNIST database of handwritten digits, 1998.

[30] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, 2009.

[31] S. Wu, M. Kan, Z. He, S. Shan, X. Chen, Funnel-structured cascade for multi-view face detection with alignment-awareness, Neurocomputing 221 (2017) 138–145.

[32] B. Graham, Fractional max-pooling, 2014. arXiv preprint arXiv:1412.6071.

[33] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[34] Y. Jia, An open source convolutional architecture for fast feature em- bedding, 2013.

[35] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 499–515.