Connection Reduction of DenseNet for Image Recognition

Rui-Yang Ju, Jen-Shiun Chiang, Chih-Chia Chen, and Yu-Shian Lin

Department of Electrical and Computer Engineering, Tamkang University

Yingzhuan Rd., Tamsui Dist., New Taipei City, 251301, Taiwan

{jryjry1094791442, jsken.chiang, crystal88irene, abcpp12383}@gmail.com

Abstract—Convolutional Neural Networks increase depth by stacking convolution layers, and deeper network models perform better in image recognition. Empirical research shows that simply stacking convolution layers does not make the network train better, and skip connection (residual learning) can improve network model performance. For the image classification tasks, models with global densely connected architectures perform well in large datasets like ImageNet, but they are not suitable for small datasets such as CIFAR-10 and SVHN. Different from dense connections, we propose two new algorithms to connect layers in this paper. $Baseline$ is a densely connected network, and the networks connected by the two new algorithms are named $ShortNet_1$ and $ShortNet_2$, respectively. The experimental results of image classification on CIFAR-10 and SVHN show that $ShortNet_1$ **has a 5% lower test error rate and 25% faster inference time than** $Baseline$. $ShortNet_2$ **speeds up inference time by 40% with less loss in test accuracy. Code and pretrained models are available at https://github.com/RuiyangJu/ Connection_Reduction/**

Index Terms—**convolution, network architecture, training, skip connection, connection reduction, image classification**

I. INTRODUCTION

Convolutional Neural Networks (CNN) have become the main network architecture in Computer Vision tasks. From the evaluation results of various network models in image classification, the deeper network models always have higher test accuracy. Therefore, researchers began to design the neural network architecture by stacking more convolution layers. However, the researchers found that a bottleneck of increasing depth occurs when simply stacking convolutional layers to a certain extent. ResNet [1] first proposes residual learning to solve this bottleneck, and the network can train deeper models to achieve higher accuracy. On this basis, DenseNet [2] inherits and improves the concept of skip connection, making all layers connect to each other, which obtains state-of-the-art performance in image classification.

Experiments [3] show that not all connections between layers are positive, and the large memory access and slow inference time of the model have become shortcomings of DenseNet. We note that DenseNet cannot be widely used in prediction tasks such as semantic segmentation and object tracking because of the above shortcomings. Therefore we propose two new algorithms for connecting between layers to replace dense connection in this paper.

This research work has two main contributions: The applying conditions of the newly proposed algorithms are not difficult and they can completely replace the dense connection. These two new algorithms can be applied to DenseNet deformation network in the future and have more applications. And this work also proves that not all connections in DenseNet play positive roles for small datasets, and appropriately reducing the connections between layers can improve the efficiency of the network model.

II. RELATED WORKS

In order to solve the gradient vanishing problem, ResNet proposes skip connection, which expresses the output of the convolution layer as a linear superposition of the input and a nonlinear transformation of the input. The key advantage of skip connection is that the distance between feature layers is short during backpropagation.

However, the extreme connection method of DenseNet makes it high computing costs in the application process. Microsoft Research proposed Log-DenseNet [4], which designs a fixed number of skip connections to reduce the total number of connections from L^2 to $Llog_2L$. In object recognition and semantic segmentation tasks, Log-DenseNet outperforms DenseNet, and the speed is improved from L^2 to $Llog_2L$. Liu *et al.* [5] proposed another approach (SparseNet) to sparse DenseNet, reducing the number of connections from L^2 to L while increasing the width of the neural network. The inference rate of SparseNet is 2.6 times faster than DenseNet in image classification. Chao *et al.* [6] analyzed that the memory traffic generated by extracting the feature map of the middle part is a factor affecting the inference speed of the network model, and proposed HarDNet. Therefore, HarDNet changes the global dense connection to the harmonic dense connection, and reduces the loss of accuracy by increasing the output weight, which achieves high efficiency in terms of low memory traffic. On this basis, ThreshNet [7] proposed threshold mechanism, which is used to judge the choice of dense connection and harmonic dense connection, and reset the number of channels in each block. In addition, there are researches [8] [9] to improve the Blocks in DenseNet. The former screen Blocks by maintaining an array recording weight, and the latter completes the network architecture by combining Blocks with two different connection methods.
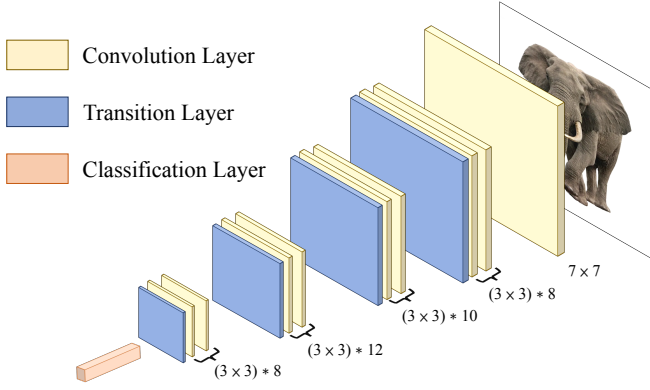
Fig. 1. Neural network with simple stacked convolutional layers.

III. RESEARCH METHOD

A. *Baseline*

In order to fairly compare the dense connection and the connections using the two algorithmic approaches, we do not use any tricks for the design of the neural network architecture. The network architecture only consists of convolution layers and transition layers. Fig. 1 represents a neural network with 43 layers, including 39 convolution layers and 4 transition layers. We design two network models with different depths, which are 43 layers and 53 layers, respectively, as shown in TABLE I.

The feature maps of each layer are the same size and can be connected in the channel dimension. The nonlinear combination function $H()$ in the Block of $Baseline$ adopts the structure of $BN + ReLU + 3 \times 3\ Conv$. Each layer of convolution in all blocks outputs k feature maps, which means the number of channels of the obtained feature map is k. k is called growth rate, which is a hyperparameter. Referring to DenseNet [2] setting for growth rate, we set k to 32 in image classification task, which can achieve higher performance.

In addition, because the input of the latter layer would be very large, DenseNet uses the bottleneck layer to reduce the amount of calculation. Bottleneck adopts the architecture of $BN + ReLU + 1 \times 1\ Conv + BN + ReLU + 3 \times 3\ Conv$. The purpose of this paper is to compare the efficiency of different ways of connecting, so the bottleneck is not used to optimize the network model in our experiment.

The connection between the layers in $Baseline$ network adopts the dense connection of DenseNet. Each layer in $Baseline$ network will be connected with all the previous layers on the channel. For the network with n layers, there are $\frac{n(n+1)}{2}$ connections in total, and the formula is as follows:

$$x_n = H_n([x_1, \cdots, x_{n-1}]) \qquad (1)$$

As shown in Fig. 1, we design a transition layer between two adjacent Blocks. The structure of the transition layer is $BN + ReLU + 1 \times 1\ Conv + 2 \times 2\ AvgPool$, and its effect is to reduce the size of the feature map. The transition layer can generate $\theta \times output$ channel features by convolution, where

$\theta \in (0, 1]$ is the compression rate. When $\theta = 1$, the number of features does not change through the transition layer. We use $\theta = 0.5$ for $Baseline$, which means the number of features is reduced by half.

TABLE I
BASELINE NETWORK MODEL ARCHITECTURE

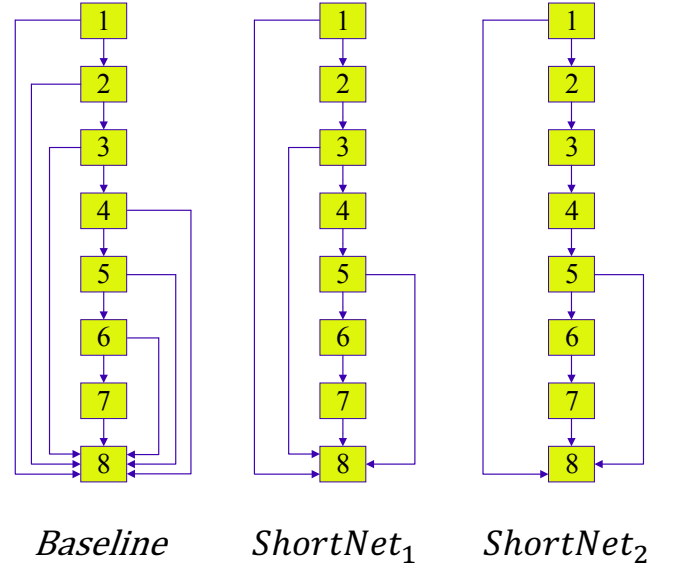| Model	Baseline-43	Baseline-53
Convolution	7×7 Conv	
Block-1	$(3 \times 3$ Conv$) * 8$	$(3 \times 3$ Conv$) * 8$
	1×1 Conv	1×1 Conv
	2×2 AvgPool	2×2 AvgPool
Block-2	$(3 \times 3$ Conv$) * 10$	$(3 \times 3$ Conv$) * 12$
	1×1 Conv	1×1 Conv
	2×2 AvgPool	2×2 AvgPool
Block-3	$(3 \times 3$ Conv$) * 12$	$(3 \times 3$ Conv$) * 16$
	1×1 Conv	1×1 Conv
	2×2 AvgPool	2×2 AvgPool
Block-4	$(3 \times 3$ Conv$) * 8$	$(3 \times 3$ Conv$) * 8$
	1×1 Conv	1×1 Conv
	2×2 AvgPool	2×2 AvgPool
Linear Classification		

Fig. 2. Example of the connections between the $8th$ convolutional layer and the previous layers.

B. $ShortNet_1$

The first new connection method we propose is different from the connection in SparseNet [5] that skips the middle part and only connects the farthest layer and the nearest layer. Instead of deleting connections from a specific concentrated part, we perform connection reductions at intervals of each layer. As shown in Algorithm 1, all layers would connect to the first layer. When the nth layer is an odd-numbered layer, it would connect to all the previous even-numbered layers, when the nth layer is an even-numbered layer, it would connect to all the previous odd-numbered layers. The connection method of this algorithm can be reduced by about half the connections compared to the dense connection used by DenseNet.

Algorithm 1 $Shortnet_1$

Require: Stack 3×3 Convolution Layer in each block
Ensure: $BN + ReLU + 3 \times 3 \ Conv$
  for layer n is odd **do**
    layer n connect to layer 2^0
    for i in 2^1 to 2^5 **do**
      let n connect to layer i ($i \leq n$ and i is even)
    end for
  end for
  for layer n is even **do**
    layer n connect to layer 2^0
    for i in 2^1 to 2^5 **do**
      let n connect to layer i ($i \leq n$ and i is odd)
    end for
  end for

For example, as shown in Fig. 2, when $n = 8$, the $8th$ layer would only connect to the $1st$, $3rd$, $5th$ and $7th$ layers, and the connections to $2nd$, $4th$ and $6th$ layers are deleted. The connection method of Algorithm 1 is shown in the following formula:

$$x_n = \left\{ \begin{array}{l} H_n\left([x_1, x_2, \cdots, x_{2^i}]\right), \ \text{if } n\%2 = 1 \\ H_n\left([x_1, x_3, \cdots, x_{2^i+1}]\right), \ \text{if } n\%2 = 0 \end{array} \right. \quad (2)$$

C. ShortNet$_2$

The second new connection method we propose is more aggressive than the first one, and it deletes more connections. This connection method is similar to the harmonic dense connection used by HarDNet [6], but the algorithm is not the same. As shown in Algorithm 2, when the layer is n, x represents all values of $2^i - 1$. x is required to be less than n, and the nth layer would connect to all $n - x$ layers.

Algorithm 2 $Shortnet_2$

Require: Stack 3×3 Convolution Layer in each block
Ensure: $BN + ReLU + 3 \times 3 \ Conv$
  for layer n is even **do**
    for i in 1 to 5 **do**
      $x = 2^i = 1$
      $y = n - x$ ($x \leq n$)
      layer n connect to layer y
    end for
  end for
  for layer n is odd **do**
    layer n connect to layer y
  end for

For example, as shown in Fig. 2, when $n = 8$, and $x = 1$, 3, 7, then $n - x = 1$, 5, 7. Therefore, the $8th$ layer would only connect to the $1st$, $5th$ and $7th$ layers. The model designed by connecting layers through this algorithm would be much smaller than the densely connected network model. The connection method of Algorithm 2 is shown in the following formula:

$$x_n = \left\{ \begin{array}{l} H_n\left([x_{n-1}]\right), \ \text{if } n\%2 = 1 \\ H_n\left([x_1, \cdots, x_{n-2^i+1}]\right), \ \text{if } n\%2 = 0 \end{array} \right. \quad (3)$$

IV. EXPERIMENT

A. Dataset

The experiments use two small datasets (CIFAR-10 and SVHN) for image classification. CIFAR-10 [10] dataset consists of 60,000 images of 32×32 pixels for 10 categories, of which 50,000 are used for training and 10,000 are used for testing. The Street View House Numbers (SVHN) [11] dataset is real house numbers captured by Google Street View. The entire dataset contains 73,257 training images and 26,032 testing images.

B. Training

For a fair comparison between $Baseline$ and $ShortNet_1$, $ShortNet_2$, we do not perform pre-training, model fine-tuning, and image augmentation. In the process of training the model, we chose a common training hyperparameter, set the learning rate to 0.001, select Adam as the optimizer, and set the training batch size to 100 epochs. All network models run on the pytorch-1.10.0 framework on a single NVIDIA RTX 3050 4GB GPU.

C. Testing

TABLE II shows the theoretical amount of floating point operation (Flops), the total number of network parameters (Params), the sum of memory read and memory write time (MemR+W), and multiplication and accumulations (MAdd) for 32×32 pixels images in dataset CIFAR-10. Compared with the dense connection, the connection method of Algorithm 1 deletes about half the number of connections, so the number of model parameters of $ShortNet_1$ is much smaller than that of $Baseline$. $ShortNet_1 - 53$ reduces the number of parameters from 3.15M ($Baseline$) to 2.16M. In addition, the memory access of $ShortNet_1 - 53$ has also been optimized to be more suitable for platforms with limited memory in real life. From the test results of datasets CIFAR-10 and SVHN, it can be seen that the performance of the $ShortNet_1$ models with different depths is better than those of $Baseline$. For example, the test error rate of $ShortNet_1 - 43$ in CIFAR-10 is 13.59%, which is lower than $Baseline - 43$ of 14.00%, while the inference time dropped from 72.83ms to 61.17ms. The calculation amount of the model can be shown by the inference time of the image. The calculation amount of $ShortNet_1$ is smaller than that of $Baseline$, and the performance of the network model is improved, which shows that the $ShortNet_1$ model is more efficient. In addition, the test results prove that for small datasets such as CIFAR-10 and SVHN, the dense connection is not the most suitable connection method. Reducing partial connections between layers often results in better performing network models. Because for the input image of 32×32 pixels, an appropriate amount of connection plays the role of backpropagation and is enough to complete the feature extraction.

TABLE II
EXPERIMENTAL RESULTS

Model	C10 GPU Time (ms)	C10 Error (%)	SVHN GPU Time (ms)	SVHN Error (%)	Flops (G)	MAdd (G)	Memory (MB)	#Params (M)	MenR+W (MB)
Baseline-43	72.83	14.00	72.64	5.95	509.38	1.02	6.08	2.17	25.93
ShortNet_1-43	61.17	13.59	58.97	5.65	374.00	0.75	4.60	1.59	18.92
ShortNet_2-43	52.48	14.09	50.61	5.48	256.44	0.51	4.00	0.97	13.74
Baseline-53	94.25	13.38	92.11	5.92	783.20	1.56	7.37	3.15	35.46
ShortNet_1-53	71.19	13.36	69.57	5.63	536.76	1.07	5.41	2.16	24.56
ShortNet_2-53	58.14	14.08	55.34	6.59	334.76	0.67	4.37	1.20	16.05

Compared with $ShortNet_1$, $ShortNet_2$ uses a more aggressive connection method algorithm, and deletes more connections, so the network model is smaller. In terms of the parameters of the model, the number of the parameters of $Baseline - 53$ is 3.15M, while the number of the parameters of $ShortNet_2$ is only 1.20M, which is reduced by more than 60% and greatly reduces the computational cost of the model. Compared with $Baseline$, the inference time of $ShortNet_2$ is much shorter. For a neural network with the depth of 43, the inference time of $ShortNet_2$ is 30% faster than that of $Baseline$, and for the depth of 53, $ShortNet_2$ is 40% faster. In the case of greatly shortened inference time, the test accuracy of the model does not reduce too much. The error rate of $ShortNet_2 - 43$ in SVHN is 5.48% that is lower than that of $Baseline$ 5.95%, and the error rate in CIFAR-10 is basically the same as $Baseline$. Because the increase of the depth of the model leads to the increase of the number of layers and the deletion of more connections, it is acceptable that the error rate of $ShortNet_2 - 53$ is slightly higher than that of $Baseline - 53$.

DenseNet is limited by its high computational cost, and its application in different scenarios often cannot be the preferred neural network. The two new algorithms proposed in this article replace the dense connection between layers, and the improved DenseNet can have more applications in the future, such as TOOD [12] and YOLOF [13] architectures for object detection scenes, SOLO [14] and DetectoRS [15] architectures for instance segmentation scenes, Panoptic FPN [16] and Mask2Former [17] architectures for panoptic segmentation scenes. The improved DenseNet model is smaller and more suitable for application in Jetson Nano, mobile phones, FPGA and other platforms [18].

V. CONCLUSION

This paper proposes two algorithms for the connection between layers. Replacing the dense connections of DenseNet with these two algorithms can reduce the amount of model parameters and reduce the computational cost. In the case of evaluating network models in small datasets, reducing the connections between layers can effectively improve model performance and greatly speed up inference while ensuring accuracy. The new algorithms promote the improvement of DenseNet, making the improved DenseNet more suitable for application in real life.

REFERENCES

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

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[3] L. Zhu, R. Deng, M. Maire, Z. Deng, G. Mori, and P. Tan, "Sparsely aggregated convolutional networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 186–201.

[4] H. Hu, D. Dey, A. Del Giorno, M. Hebert, and J. A. Bagnell, "Log-densenet: How to sparsify a densenet," *arXiv preprint arXiv:1711.00002*, 2017.

[5] W. Liu and K. Zeng, "Sparsenet: A sparse densenet for image classification," *arXiv preprint arXiv:1804.05340*, 2018.

[6] P. Chao, C.-Y. Kao, Y.-S. Ruan, C.-H. Huang, and Y.-L. Lin, "Hardnet: A low memory traffic network," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3552–3561.

[7] R.-Y. Ju, T.-Y. Lin, J.-H. Jian, J.-S. Chiang, and W.-B. Yang, "Threshnet: An efficient densenet using threshold mechanism to reduce connections," *arXiv preprint arXiv:2201.03013*, 2022.

[8] H. Wang, J. Yang, W. Kou, X. Chen, J. Li, and Y. Li, "Sparsenet: Deep convolutional network with sparse connections between blocks," in *ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application*. VDE, 2021, pp. 1–5.

[9] R.-Y. Ju, T.-Y. Lin, and J.-S. Chiang, "New pruning method based on densenet network for image classification," in *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE, 2021, pp. 107–111.

[10] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[11] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[12] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "Tood: Task-aligned one-stage object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2021, pp. 3490–3499.

[13] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 039–13 048.

[14] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "Solo: Segmenting objects by locations," in *European Conference on Computer Vision*. Springer, 2020, pp. 649–665.

[15] S. Qiao, L.-C. Chen, and A. Yuille, "Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 213–10 224.

[16] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6399–6408.

[17] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.

[18] R.-Y. Ju, T.-Y. Lin, J.-H. Jian, and J.-S. Chiang, "Triplenet: A low computing power platform of low-parameter network," *arXiv preprint arXiv:2204.00943*, 2022.