Convolutional Neural Networks with Alternately Updated Clique

Yibo Yang[1,2],   Zhisheng Zhong[2],   Tiancheng Shen[1,2],   Zhouchen Lin[2,3,*]

[1]Academy for Advanced Interdisciplinary Studies, Peking University

[2]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University

[3]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University

{ibo,zszhong,tianchengShen,zlin}@pku.edu.cn

Abstract

Improving information flow in deep networks helps to ease the training difficulties and utilize parameters more efficiently. Here we propose a new convolutional neural network architecture with alternately updated clique (CliqueNet). In contrast to prior networks, there are both forward and backward connections between any two layers in the same block. The layers are constructed as a loop and are updated alternately. The CliqueNet has some unique properties. For each layer, it is both the input and output of any other layer in the same block, so that the information flow among layers is maximized. During propagation, the newly updated layers are concatenated to re-update previously updated layer, and parameters are reused for multiple times. This recurrent feedback structure is able to bring higher level visual information back to refine low-level filters and achieve spatial attention. We analyze the features generated at different stages and observe that using refined features leads to a better result. We adopt a multi-scale feature strategy that effectively avoids the progressive growth of parameters. Experiments on image recognition datasets including CIFAR-10, CIFAR-100, SVHN and ImageNet show that our proposed models achieve the state-of-the-art performance with fewer parameters [1].

1. Introduction

In recent years, the structure and topology of deep neural networks have attracted significant research interests, since the convolutional neural network (CNN) based models have achieved huge success in a wide range of tasks of computer vision. A notable trend of those CNN architectures is that the layers are going deeper, from AlexNet [22] with 5 convolutional layers, the VGG network and GoogLeNet with 19 and 22 layers, respectively [31, 35], to recent ResNets [12] whose deepest model has more than one thousand layers. However, inappropriately designed deep networks
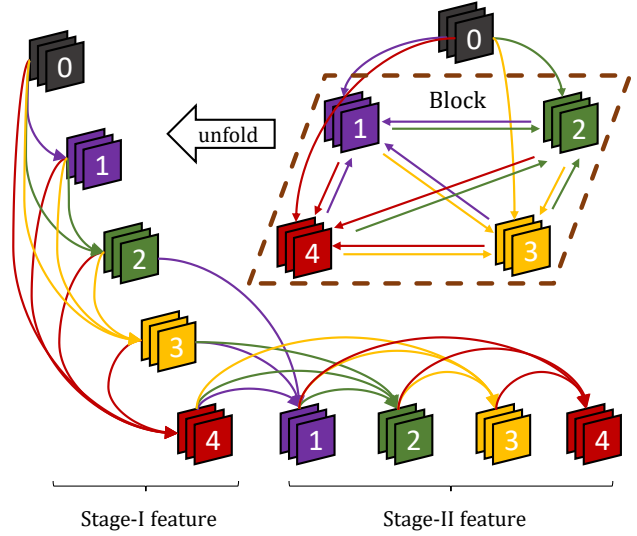

Figure 1. An illustration of a block with 4 layers. Any layer is both the input and output of another one. Node 0 denotes the input layer of this block.

would make it hard for latter layer to access the gradient information from previous layers, which may cause gradient vanishing and parameter redundancy problems [16, 17].

Successfully adopted in ResNet [12] and Highway Network [33], skip connection is an efficient way to make top layers accessible to the information from bottom layers, and ease the network training at the same time, due to its relief of the gradient vanishing problem. The residual block structure in ResNet [12] also inspires a series of ResNet variations, including ResNext [39], WRN [40], PolyNet [43], *etc*. To further activate the gradient and information flow in networks, DenseNet [16] is a newly proposed structure, where any layer in a block is the output of all preceding layers, and the input of all subsequent layers. Recent studies show that the skip connection mechanism can be extrapolated as a recurrent neural network (RNN) or LSTM [13], when weights are shared among different layers [26, 4, 20]. In this way, the deep residual network

---

[1]Code address: http://github.com/iboing/CliqueNet

is treated as a long sequence and hidden units are linked by skip connections. While this recurrent structure benefits feature re-usage and iterative learning, the residual information is restricted among neighboring layers and cannot be considered across multiple layers, because the recurrence only happens once at each single layer.

Attention mechanism is another focus of recent studies on network structure [38, 36, 1, 27] and applications [3, 28, 23, 7]. When people watch a picture or a scene, the information on our target is better captured if we re-look at or re-think the target with additional attention. In cognition theory, the activity of a neuron in visual cortex is influenced by other cortical area's responses transferred through feedback connections [18, 14]. This motivates the introduce of feedback to deep networks [34, 41]. The feedback connections that bring back higher-level semantic information in a top-down manner are able to re-weight the focus, and suppress the non-relevant neuron activations of background and noises.

Inspired by the recurrent structure and attention mechanism, in this study, we propose a new convolutional neural network architecture with alternately updated clique (CliqueNet). In contrast to prior network structures, there are both forward and feedback connections between any two layers in the same block. As illustrated in Figure 1, the layers in Clique Block are constructed as a clique and are updated alternately. Concretely, the several previous layers are concatenated to update the next layer, after which, the newly updated layer is concatenated to re-update the previous layer, so that information flow and feedback mechanism can be maximized. Each layer in a block is both the input and output of another one, which means they are more densely connected than DenseNets [16]. We adopt a multi-scale feature strategy to compose the final representation with the block features in different map sizes.

The CliqueNet architecture has some unique properties. An intuition would tell that our proposal is parameter-demanding, because given a block with n layers, DenseNet [16] needs C_n^2 groups of parameters, while ours needs A_n^2 (C and A represents combination operator and permutation operator, respectively). However, the filters in DenseNet increase linearly as the depth rises [4], which may leads to the rapid growth of parameters. In our architecture, only the Stage-II feature in each block is fed into the next block. It turns out that this is a more parameter-efficient way. In addition, traditional neural networks add a new layer with its corresponding parameters. As for CliqueNet, the weights among layers in a block keep recycling during propagation. The layers can be updated alternately for multiple times so that a deeper representation space is attained with the fixed number of parameters.

The CliqueNet architecture also shows a stronger ability for representation learning due to the combination of re-

current structure and feedback mechanism. In each Clique Block, both forward and feedback are densely connected. The information flow is maximized and feature maps are repeatedly refined by attention. The Stage-II feature in each block is accessed to the loss function and fed into the next block, to enable feature re-usage and new features learning, respectively. We observe that our architecture can suppress the activations of background and noises, and prevent overfitting without resorting to data augmentation.

The contributions in this study are listed as follows:

- We propose a new convolutional neural network architecture called CliqueNet, which incorporates both forward and backward connections between any two layers in the same block. The layers constructed as a loop are updated alternately. The CliqueNet that combines both recurrent structure and attention mechanism, is able to maximize information flow and achieve feature refinement. We show that the refined features are more discriminative and lead to a better performance.
- We adopt a multi-scale feature strategy that effectively circumvents the progressive increment of parameters, despite the extra feedback connections.
- We conduct experiments on four benchmark datasets including CIFAR-10, CIFAR-100, SVHN and ImageNet to demonstrate the superiority of our models.

2. Related Work

A number of deep networks with large model capacity have been proposed. For widening the network, the Inception modules in GoogLeNet [35] fuse the features in different map size to construct a multi-scale representation. Multi-column [5] nets and Deeply-Fused Nets [37] also use fusion strategy and have a wide network structure. Wide residual networks [40] increase the width and decrease the depth to improve the performance, while FractalNet [24] deepen and widen at the same time. However, simply widening the network is easy to consume more runtime and memory [43]. For deepening the networks, skip connections or shortcut paths are widely adopted strategies to ease the network training [12, 33]. In [17], it is shown that some of the layers in ResNets are dispensable and cause parameters redundancy. So they randomly drop a subset of layers to ease the training and achieve a better performance. To further increase information flow, DenseNets [16] replace the identity mapping in residual block by concatenating operation, so that new feature learning can be reinforced while keeping old feature re-usage. In line with this view, dual path networks (DPN) [4] are proposed to combine both advantages of residual path and densely connected path.

Both residual path and densely connected path correspond to a recurrent propagation, and their success has been attributed to the recurrent structure and iterative refinement [26, 10, 20]. Studies incorporating recurrent connections
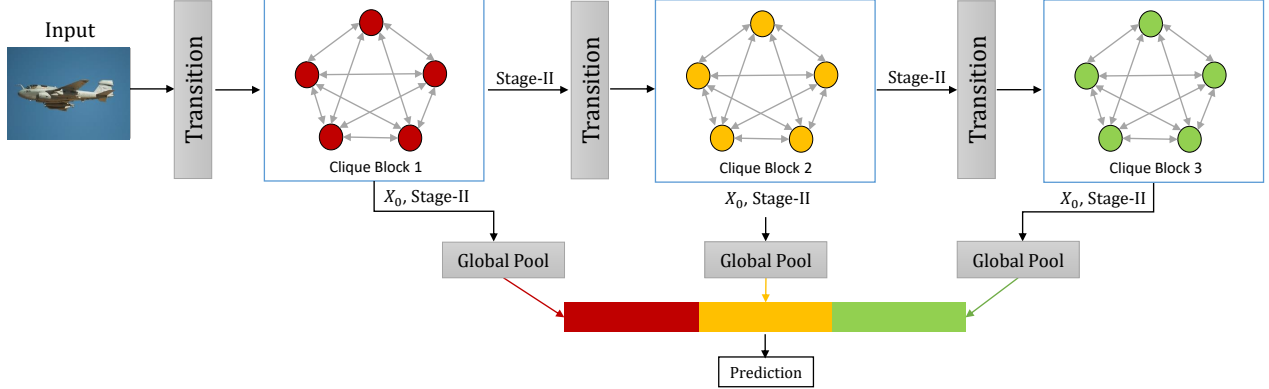
Figure 2. A CliqueNet with three blocks. The input layer together with the Stage-II feature in each block are concatenated to be the block feature, and form part of the final representation after global pooling. The Stage-II feature passes through transition layers, which include a convolution and an average pooling to change map size, and then becomes the input of the next block.

into CNNs also show superiority in object recognition [25], scene parsing [30] and some other tasks. CliqueNet differs from these structures in that the iterative mechanism exists in each step of the propagation, instead of just between neighboring layers or from the top layer to the bottom layer; all layers in a block participate in the recurrent loop so that the filters are communicated sufficiently and the blocks play both roles of information carrier and refiner.

Recent studies have embraced the attention mechanism as an effective technique to strengthen some neurons that feature the target, and improve the performance as a result. It is proved fruitful in many applications, including image recognition [36, 7], image captioning [3], image-text matching [28], and saliency detection [23]. In general, visual attention can be achieved by formulating an optimization problem [1], weighting the activations spatially or channel-wisely [3, 15], and introducing feedback connections [38, 34, 41]. In [41], the model makes consecutive decisions for a more accurate prediction via feedback connections. The input of the next decision is based on the output of the last decision. Experiments show that the top-down propagation is capable of refining lower-level features, and improving classification performance [34], especially on datasets with noise and occlusion [38, 27]. But how to make a proper attention mechanism and boost the supervision between layers remains further exploration.

There are also some studies that design attention mechanism tied with recurrent neural networks [27, 23, 7]. A recent report [2] tries to propose a loopy net, but it just repeats the skip connections and does not make layers communicated. CliqueNet enables true cycling because of the alternate propagation. Although alternate updating has been an important method in the optimization theory [8], it has not been introduced into deep learning. At the best of out knowledge, we are the first to use updated layers to re-update previous layers alternately, and these layers construct a loop to cycle for multiple times.

3. CliqueNet Architecture

The CliqueNet architecture has two main ingredients, the block with alternately updated clique (Clique Block) to enable feature refinement, and the multi-scale feature strategy that facilitates parameter efficiency.

3.1. Clique Block

In order to maximize the information flow among layers, we design the Clique Block. Any two layers in the same block are connected bidirectionally except for the input node. Compared with Dense Block [16] where each layer is the output of all previous layers, and the input of all subsequent layers, Clique Block makes each layer both the input and output of any other layers. The propagation of a Clique Block with 5 layers is illustrated in Table 1. At the first stage, the input layer (X_0) initializes all layers in this block by single directional connections. Each updated layer is concatenated to update the next layer. From the second stage, the layers begin updating alternately. All layers except the top layer to be updated are concatenated as the bottom layer, and their corresponding parameters are also concatenated. Accordingly, the ith ($i \geq 1$) layer in the kth ($k \geq 2$) loop can be formulated as:

$$X_i^{(k)} = g\left(\sum_{l<i} W_{li} * X_l^{(k)} + \sum_{m>i} W_{mi} * X_m^{(k-1)}\right) \quad (1)$$

where $*$ denotes the convolution operation with parameters W, and g is the non-linear activation function. Each layer will always receive the feedback information from the layers that are updated more lately. It achieves a spatial attention mechanism due to the top-down refinement brought by each propagation. This recurrent feedback structure ensures that the communication is maximized among all layers in the block.

Bottom Layers	Weights	Top Layer	Feature
X_0	W_{01}	$X_1^{(1)}$	
$\{X_0, X_1^{(1)}\}$	$\{W_{02}, W_{12}\}$	$X_2^{(1)}$	
$\{X_0, X_1^{(1)}, X_2^{(1)}\}$	$\{W_{03}, W_{13}, W_{23}\}$	$X_3^{(1)}$	Stage-I
$\{X_0, X_1^{(1)}, X_2^{(1)}, X_3^{(1)}\}$	$\{W_{04}, W_{14}, W_{24}, W_{34}\}$	$X_4^{(1)}$	
$\{X_0, X_1^{(1)}, X_2^{(1)}, X_3^{(1)}, X_4^{(1)}\}$	$\{W_{05}, W_{15}, W_{25}, W_{35}, W_{45}\}$	$X_5^{(1)}$	
$\{X_2^{(1)}, X_3^{(1)}, X_4^{(1)}, X_5^{(1)}\}$	$\{W_{21}, W_{31}, W_{41}, W_{51}\}$	$X_1^{(2)}$	
$\{X_3^{(1)}, X_4^{(1)}, X_5^{(1)}, X_1^{(2)}\}$	$\{W_{32}, W_{42}, W_{52}, W_{12}\}$	$X_2^{(2)}$	
$\{X_4^{(1)}, X_5^{(1)}, X_1^{(2)}, X_2^{(2)}\}$	$\{W_{43}, W_{53}, W_{13}, W_{23}\}$	$X_3^{(2)}$	Stage-II
$\{X_5^{(1)}, X_1^{(2)}, X_2^{(2)}, X_3^{(2)}\}$	$\{W_{54}, W_{14}, W_{24}, W_{34}\}$	$X_4^{(2)}$	
$\{X_1^{(2)}, X_2^{(2)}, X_3^{(2)}, X_4^{(2)}\}$	$\{W_{15}, W_{25}, W_{35}, W_{45}\}$	$X_5^{(2)}$	

$\cdots$

Table 1. A diagram of CliqueNet's propagation in a block with 5 layers. W_{ij} is the weights of parameter from X_i to X_j. "{}" denotes the concatenation operator. The Stage-II feature is to be transited as the input layer (X_0) of the next block.
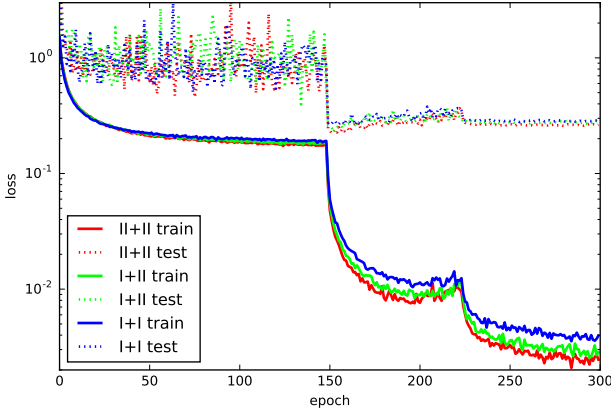

Figure 3. Training and testing curves of different versions of CliqueNets. Learning rate is divided by 10 at epoch 150 and 225.

name	block feature	transit	error(%)
CliqueNet (I+I)	X_0, Stage-I	Stage-I	6.64
CliqueNet (I+II)	X_0, Stage-I	Stage-II	6.1
CliqueNet (II+II)	X_0, Stage-II	Stage-II	5.76

Table 2. Results of different versions of CliqueNets on CIFAR-10.

For the purpose of analyzing the features generated in different stages, we conduct experiments on CIFAR-10 dataset (with no data augmentation) using different versions of CliqueNets. As Table 2 shows, the CliqueNet (I+I) only considers the Stage-I feature. The CliqueNet (I+II) uses the Stage-I feature and input layer as block feature to access loss function, but transits the Stage-II feature into the next block. The CliqueNet (II+II) adopts our strategy mentioned above. They all have 3 blocks with 5 layers in each block. Each layer contains 36 filters. The experimental settings are following [16]. The main results are shown in Figure 3. It is found that the introduce of Stage-II feature indeed leads to a better result by a significant margin. We adopt the CliqueNet (II+II) structure for the following experiments.

3.2. Feature at Different Stages

We analyze the features produced at different stages, and adopt a multi-scale feature strategy to avoid the rapid increment of parameters.

The first stage is used to initialize all layers in the block, and the layers are refined repeatedly since the second stage. Given that the Stage-II feature is refined with attention and assimilates more high level visual information, we make the Stage-II feature together with the input layer in each block concatenated as the block feature, and then accessed to the loss function after global pooling. Only the Stage-II feature is fed into the next block; see Figure 2. In this way, the final representation is characterized by multi-scale feature maps, and the dimensionality in each block will not increase progressively. Because higher stage propagation comes with more computation cost and amplifies the model complexity, we only consider the first two stages.

3.3. Extra Techniques

In addition to the structures mentioned above, we consider some techniques to help strengthen the model and improve the state of the art. In the experimental section, we conduct experiments with and without these additional techniques to show the effectiveness of our model.

Attentional transition. The CliqueNet includes feedback connections to refine lower level activations using higher level visual information. The attention mechanism weight the feature maps spatially to weaken the noises and background. The channel-wise attention, adopted in [3, 36, 15], also benefits recognition problem because it recalibrates

4

different filters to prevent overfitting and inspire new features learning. In CliqueNet, we incorporate channel-wise attention mechanism in transition layers, following the method proposed in [15]. As depicted in Figure 4, the filters are globally averaged after the convolution in transition. They are followed by two fully connected (*FC*) layers. The first *FC* layer has half of the filters and is activated by *Relu* function. The second *FC* layer has the same number of filters and is activated by *Sigmoid* function, so that the activation is scaled into $[0, 1]$ and acts on the input layer by filter-wise multiplication. Different from [15] which sets this module at each residual layer, we only add it to transition layers in order to adjust the filters into the next block.

Bottleneck and compression. Bottleneck is an effective way to decrease the number of parameters and provide further potential to enlarge model capacity. It is conjectured [40] that bottleneck architecture is suitable for deeper networks and large dataset like ImageNet, and recent studies have embraced bottleneck for a better performance [12, 16, 36, 4]. So we introduce bottleneck to our large models. The 3×3 convolution kernels in each block are replaced by 1×1, and produce a middle layer, after which, a 3×3 convolution layer follows to produce the top layer. The middle layer and top layer contain the same number of feature maps. Compression is another tool adopted in [16] to make the model more compact. Instead of compressing the number of filters in transition layers as they do, we only compress the features that are accessed to the loss function, *i.e.* the Stage-II concatenated with its input layer. The models with compression have an extra convolutional layer with 1×1 kernel size before global pooling. It generates half the number of filters to enhance model compactness and keep the dimensionality of the final feature in a proper range.

3.4. Implementation

In our experiments, we test our models on benchmark datasets without the aforementioned extra techniques to show the effectiveness of CliqueNet, and further improve the state-of-the-art performance with them. There are two structure parameters, the sum of layers in all blocks, *T*, and the number of filters per layer, *k*. For our models without bottleneck, convolution layers in each block are with 3×3 kernel size and padded by one pixel to keep the feature maps in the same size. Blocks are linked by transition layers, where a convolution layer with 1×1 kernel size is followed by 2×2 average pooling. All convolutions are performed in a unit composed of three consecutive operations: batch normalization[19], Relu, and the convolution. Stage-II feature with its input layer from all blocks are concatenated after global pooling, and end with a fully-connected layer with softmax.

For experiments on CIFAR and SVHN, there are three blocks in total, in which the feature map sizes are 32×32,
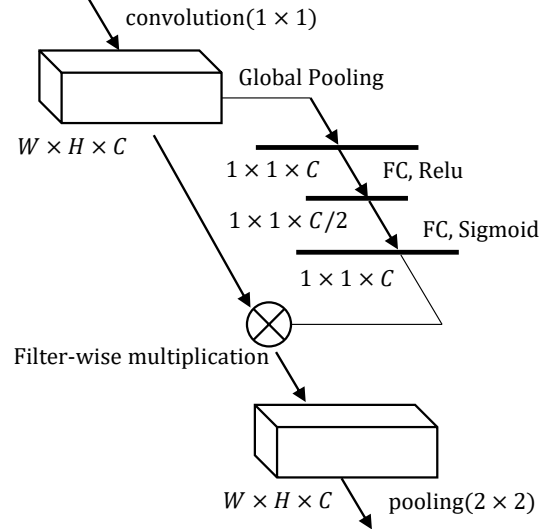

Figure 4. A schema for attentional transition. The transition layer consists of convolution and pooling. The filter-wise multiplication happens after convolution and before down pooling. W, H and C are width, height and channels of feature maps.

Layer	S0	S1	S2	S3
Convolution (112×112)	conv (7×7), 64, stride 2			
Pooling (56×56)	max pool (3×3), stride 2			
Block 1 (56×56)	36×5	36×5	36×5	40×6
Transition: conv (1×1), avg pool (2×2)				
Block 2 (28×28)	64×6	80×6	80×5	80×6
Transition: conv (1×1), avg pool (2×2)				
Block 3 (14×14)	100×6	120×6	150×6	160×6
Transition: conv (1×1), avg pool (2×2)				
Block 4 (7×7)	80×6	100×6	120×6	160×6

Table 3. Structures on ImageNet. The first number in each block is the number of filters per layer, and the second denotes the number of layers in this block.

16×16, and 8×8, respectively. Before entering the first block, the input images pass through a 3×3 convolution with output channels set to be 64 as the input layer (X_0) of the first block. As for ImageNet, we use four blocks with bottleneck and compression, and compare our results with and without attentional transition. The initial transition has 7×7 convolution with stride 2 and 3×3 max pooling with stride 2 on the 224×224 input images. Our four network structures on ImageNet are shown in Table 3.

Model	A	B	C	FLOPs	Params	CIFAR-10	CIFAR-100	SVHN
Recurrent CNN [25]	-	-	-	-	1.86M	8.69	31.75	1.80
Stochastic Depth ResNet [17]	-	-	-	-	1.7M	11.66	37.8	1.75
dasNet [34]	-	-	-	-	-	9.22	33.78	-
FractalNet [24]	-	-	-	-	38.6M	7.33	28.2	1.87
DenseNet ($k = 12, T = 36$) [16]	-	-	-	0.53G	1.0M	7.00	27.55	1.79
DenseNet ($k = 12, T = 96$) [16]	-	-	-	3.54G	7.0M	5.77	23.79	1.67
DenseNet ($k = 24, T = 96$) [16]	-	-	-	13.78G	27.2M	5.83	23.42	1.59
CliqueNet ($k = 36, T = 12$)	-	-	-	0.91G	0.94M	5.93	27.32	1.77
CliqueNet ($k = 64, T = 15$)	-	-	-	4.21G	4.49M	5.12	23.98	1.62
CliqueNet ($k = 80, T = 15$)	-	-	-	6.45G	6.94M	**5.10**	**23.32**	**1.56**
CliqueNet ($k = 80, T = 18$)	-	-	-	9.45G	10.14M	**5.06**	**23.14**	**1.51**
DenseNet ($k = 12, T = 96$) [16]	-	✓	✓	0.58G	0.8M	5.92	24.15	1.76
DenseNet ($k = 24, T = 246$) [16]	-	✓	✓	10.84G	15.3M	5.19	**19.64**	1.74
CliqueNet ($k = 36, T = 12$)	✓	-	-	0.91G	0.98M	5.8	26.41	-
CliqueNet ($k = 36, T = 12$)	-	-	✓	0.98G	1.04M	5.69	26.45	-
CliqueNet ($k = 36, T = 12$)	✓	-	✓	0.98G	1.08M	5.61	25.55	1.69
CliqueNet ($k = 80, T = 15$)	✓	-	✓	6.88G	8M	**5.17**	22.78	**1.53**
CliqueNet ($k = 150, T = 30$)	✓	✓	✓	8.49G	10.02M	**5.06**	**21.83**	**1.64**

Table 4. Error rates (%) on CIFAR-10, CIFAR-100, and SVHN **without** any data augmentation. In CliqueNets and DenseNets, k is the number of filters per layer, and T is the total number of layers in three blocks. "A, B, C" represents attentional transition, bottleneck and compression, respectively. The FLOPs of DenseNets are calculated by ourselves.

4. Experiments

We evaluate the CliqueNet on benchmark classification datasets, including CIFAR-10, CIFAR-100, SVHN and ImageNet, and compare our results with the state of the arts.

4.1. Datasets and Training Details

CIFAR. The CIFAR-10 and CIFAR-100 datasets [21] are both 32×32 colored images. CIFAR-10 dataset consists of 60,000 images in 10 classes, with 6,000 images in each class. There are 50,000 images for training and 10,000 images for testing. CIFAR-100 dataset is similar to CIFAR-10 but has 100 classes, each of which contains 600 images. For data normalization, we preprocess the dataset by subtracting the mean and dividing by the standard deviation.

SVHN. The Street View House Number (SVHN) [29] dataset contains 32×32 colored images of house numbers cropped from Google Street View. There are 73,257 images in the training set, 26,032 in the testing set and 531,131 digits for additional training. Following the common practice [40, 17, 24, 16], we use all training samples without augmentation and divide images by 255 for normalization. We report the lowest error rate on the testing set.

ImageNet. We also conduct experiments on ILSVRC 2012 dataset[6], which contains 1.2 million training images, 50,000 validation images, and 100,000 test images with 1,000 classes. Following [12, 16], we adopt the standard data augmentation for the training sets. A 224×224 crop is randomly sampled from the images or its horizontal flip. The images are normalized into $[0, 1]$ using mean values and standard deviations. We report the single-crop error rate on the validation set.

Training Details. For fair comparison, we do not take much hyper-parameter tuning, and most of our training strategies are following [12, 16]. We train our models using stochastic gradient descent (SGD) with 0.9 Nesterov momentum and 10^{-4} weight decay. The parameters are initialized according to [11] and the weights of fully connected layer are using Xavier initialization [9]. For CIFAR and SVHN, we train for 300 epochs and 40 epochs, respectively, with batchsize of 64. The learning rate is set to be 0.1 initially and is divided by 10 at 50% and 75% of the training procedure. Compared with ImageNet, the experiments on CIFAR and SVHN are not resorting to any data augmentation, and we add a dropout layer [32] with drop out rate 0.2 after each convolution layer following[16]. For ImageNet, we train our models for 100 epochs and drop the learning rate by 0.1 at epoch 30, 60, and 90. Because we have only server with 4 GPUs and are constrained by GPU memory, the batchsize is 160 for our models on ImageNet, instead of 256 as most studies did.

4.2. Results on CIFAR and SVHN

Our experimental results on CIFAR and SVHN are shown in Table 4. The first part in the table includes some methods before DenseNets and some other studies that also incorporate feedback connections or attention mechanism. The second and third parts compare the CliqueNets with

Model	Params	top-1	top-5
ResNet-18 [12]	11.7M	30.43	10.76
CliqueNet-S0*	5.7M	27.52	8.98
ResNet-34 [12]	21.8M	26.73	8.74
CliqueNet-S1*	7.96M	26.21	8.3
CliqueNet-S2*	10M	25.85	8.02
DenseNet-121 [16]	7.98M	25.02	7.71
CliqueNet-S2	11M	24.82	7.51
CliqueNet-S3*	13.17M	24.98	7.48
ResNet-50 [12]	25.6M	24.01	7.02
CliqueNet-S3	14.38M	24.01	7.15

Table 5. Single crop error rates (%) on ImageNet. The ∗ indicates the models without attentional transition.

Figure 5. Visualization of the weights in the first block in pretrained DenseNet (left) and CliqueNet (right) by calculating the average absolute value of W_{ij}. Node 0 denotes the input layer of this block.

DenseNets when they both have no extra technique. The last two parts show the situation with extra techniques. The best result and the second best result are marked by red bold and bold, respectively.

Without extra techniques. The first three parts show that, when extra techniques are not considered, CliqueNets outperform most previous methods on CIFAR-10, CIFAR-100, and SVHN with significantly fewer parameters. Because the layers in CliqueNet can be re-updated but contribute features in each cycle, the depth of CliqueNet is much shallower than other models. For our smallest model CliqueNet (36-12), (representing $k = 36$, and $T = 12$), each block contains 4 layers. It has the same number of filters, 144, in each block as DenseNet (12-36), but reduce the error rate from 7% to 5.93% on CIFAR-10 with slightly fewer parameters than its counterpart DenseNet (12-36). Although the ResNet with stochastic depth [17] achieved a slightly better performance with 1.7M parameters on SVHN than CliqueNet (36-12), our model drops the error rate on CIFAR-10 and CIFAR-100 by a large margin. As the model capacity goes larger, we find that the performance of CliqieNets is getting better without overfitting. As for our model CliqueNet (80-15), it has already achieved the state of the art on three datasets, and even outperforms the DenseNets that use extra techniques on CIFAR-10 and SVHN. It has only 6.94M parameters, which are a quarter of DenseNet (24-96) with 27.2M parameters, and a half of DenseNet (24-246) using bottleneck and compression with 15.3M parameters.

With extra techniques. The CliqueNets realize spatial attention mechanism due to its recurrent feedback propagation. When armed with channel-wise attention, they achieve an improved performance. This is demonstrated by the CliqueNet (36-12) with attentional transition. It has a better result on CIFAR-10 and CIFAR-100 with slightly more parameters. The compression has the same effect by making the model more compact. It is shown that the attentional transition is compatible with compression. The CliqueNet
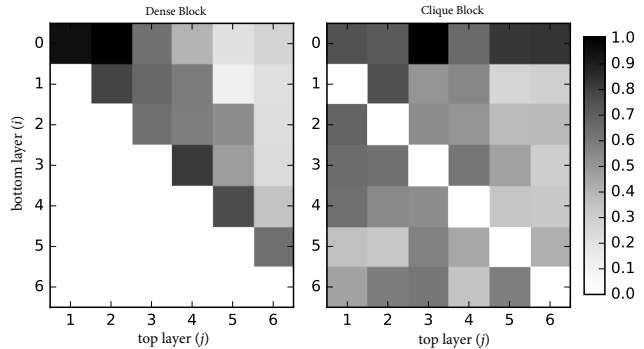
(36-12) with both attentional transition and compression leads to a better result than its original version and its original version with only attentional transition or compression. Compared with its counterpart DenseNet (12-36), it drops an error rate of 1.39% on CIFAR-10, 2% on CIFAR-100, and 0.1% on SVHN, with just 0.08M more parameters. The CliqueNet (80-15) with attentional transition and compression also has an improvement than its original version, and increases the state of the art of SVHN to 1.53% with 8M parameters, while the previously best result 1.59% on SVHN performed by DenseNet (24-96) has three times more parameters. The bottleneck architecture is effective to save parameters, and our largest model CliqueNet (150-15) with bottleneck further improves the performance on CIFAR-10 and CIFAR-100, but increases parameter and computation cost moderately.

4.3. Results on ImageNet

Because we have limited computational resource and can only spread a batch among 4 GPUs, we use a batchsize of 160 on ImageNet, instead of 256 in most studies. Although a smaller batchsize would impair the performance training for the same epochs, the CliqueNets achieve a comparable result on ImageNet with ResNets or DenseNets; see Table 5. This indicates that our proposed models can also be applied on large datasets.

The CliqueNet-S0* and CliqueNet-S1* outperform the ResNet-18 and ResNet-34 with only a half of their parameters. Larger models also achieve on par with the state of the art performed by ResNets and DenseNets. When the attentional transition is considered, the CliqueNet contains both spatial attention and channel-wise attention, and has a better performance accordingly. The CliqueNet-S2 and CliqueNet-S3 both reduce about 1% top-1 error rate compared with their original versions, CliqueNet-S2* and CliqueNet-S3* that do not have attentional transition.

4.4. Further Discussion

In order to better analyze the recurrent feedback mechanism and the multi-scale feature strategy in CliqueNet, we visualize feature maps and parameters based on pre-trained models and provide a further understanding.

Parameter efficiency. Despite the fact that CliqueNet has bipartite connections between any two layers in the same block, which would bring more parameters in the block, we find that the CliqueNet achieves the state of the art on CIFAR and SVHN dataset with considerably fewer parameters than DenseNets. On ImageNet, the CliqueNet using a smaller batchsize also has parameter efficiency compared with ResNets. This is mainly due to the multi-scale feature strategy that only transits the Stage-II feature into the next block, instead of having feature maps stacked towards deeper layers, which may cause progressive increment of parameters. In Figure 5, we visualize the weights among layers within a block of pre-trained CliqueNet and DenseNet. The color pixel of Clique Block covers the whole heat map because of our feedback connections. It is noted that the red dots in Dense Block are concentrated along the diagonal. A similar result is also reported in [16]. The observation reveals that only neighboring layers have strong dependency in DenseNet, while its forward stacking pattern is actually parameter-demanding. This helps to explain the parameter and flop efficiency in CliqueNet where information flow is distributed more evenly in each block.

Feature refinement. In CliqueNet, the layers are updated alternately so that they are supervised by each other. Moreover, in the second stage, feature maps always receive a higher-level information from the filters that are updated more lately. This spatial attention mechanism makes layers refined repeatedly, and is able to repress the noises or background of images and focus more activations on the region that characterize the target object. In order to test the effects, we visualize the feature maps following the methods in [42]. As shown in Figure 6, we choose three input images with complex background from ImageNet validation set, and visualize their feature maps with the highest average activation magnitude in the Stage-I and Stage-II, respectively. It is observed that, compared with the Stage-I, the feature maps in Stage-II diminish the activations of surrounding objects and focus more attention on the target region. This is in line with the conclusion in Table 2 that the Stage-II feature is more discriminative and leads to a better performance.

5. Conclusion

In this study, we introduce a new convolutional neural network architecture where the layers in a block are constructed as a clique and are updated alternately in a loop manner. Any layer is both the input and output of another
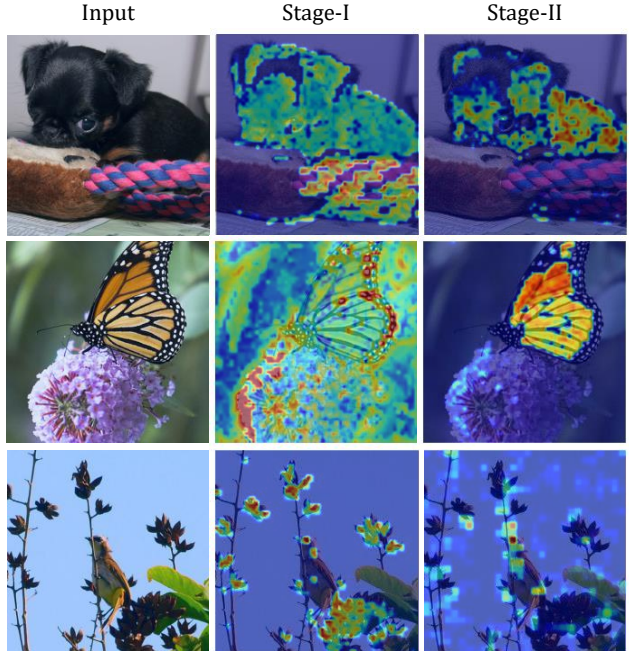

| Input | Stage-I | Stage-II |

Figure 6. Feature maps of Stage-I and Stage-II with the highest average activation in a pre-trained model. The activations of background or surrounding objects are repressed in Stage-II.

one in the same block so that the information flow is maximized. The parameters are circulated in the course of propagation and are able to produce multiple stage features. We analyze the feature in different stages and observe that the introduce of the Stage-II feature helps to suppress noises and leads to a better performance. The multi-scale feature strategy effectively circumvents the progressive increment of parameters. Experiments show that our proposed architectures are able to achieve the state of the arts with fewer parameters, especially on CIFAR and SVHN without resorting to data augmentation.

Different from prior networks, the CliqueNet utilizes a fixed number of parameters to attain a deeper representation space and incorporates the recurrent feedback to achieve attention mechanism. This topology provides the potential of developing models for other computer vision tasks in future work, such as semantic segmentation, salient object detection, image captioning, *etc.*

Acknowledgements

References

[1] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV*, pages 2956–2964, 2015. 2, 3

[2] I. Caswell, C. Shen, and L. Wang. Loopy neural nets: Imitating feedback loops in the human brain. *Tech.Report*. 3

[3] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, July 2017. 2, 3, 4

[4] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. Dual path networks. In *NIPS*, 2017. 1, 2, 5

[5] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649. IEEE, 2012. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009. 6

[7] J. Fu, H. Zheng, and T. Mei. Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, 2017. 2, 3

[8] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976. 3

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 6

[10] K. Greff, R. K. Srivastava, and J. Schmidhuber. Highway and residual networks learn unrolled iterative estimation. In *ICLR*, 2017. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 6

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 5, 6, 7

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[14] J. B. Hopfinger, M. H. Buonocore, and G. R. Mangun. The neural mechanisms of top-down attentional control. *Nature neuroscience*, 3(3):284–291, 2000. 2

[15] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017. 3, 4, 5

[16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, July 2017. 1, 2, 3, 4, 5, 6, 7, 8

[17] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661. Springer, 2016. 1, 2, 6, 7

[18] J. Hupé, A. James, B. Payne, S. Lomber, P. Girard, and J. Bullier. Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons. *Nature*, 394(6695):784–787, 1998. 2

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 5

[20] S. Jastrzebski, D. Arpit, N. Ballas, V. Verma, T. Che, and Y. Bengio. Residual connections encourage iterative inference. *arXiv preprint arXiv:1710.04773*, 2017. 1, 2

[21] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 6

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[23] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *CVPR*, pages 3668–3677, 2016. 2, 3

[24] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *ICLR*, 2017. 2, 6

[25] M. Liang and X. Hu. Recurrent convolutional neural network for object recognition. In *CVPR*, pages 3367–3375, 2015. 2, 6

[26] Q. Liao and T. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *arXiv preprint arXiv:1604.03640*, 2016. 1, 2

[27] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *NIPS*, pages 2204–2212, 2014. 2, 3

[28] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *CVPR*, July 2017. 2, 3

[29] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011. 6

[30] P. H. O. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, pages I–82, 2014. 3

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[32] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014. 6

[33] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In *NIPS*, pages 2377–2385, 2015. 1, 2

[34] M. F. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber. Deep networks with internal selective attention through feedback connections. In *NIPS*, pages 3545–3553, 2014. 2, 3, 6

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1, 2

[36] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *CVPR*, July 2017. 2, 3, 4, 5

[37] J. Wang, Z. Wei, T. Zhang, and W. Zeng. Deeply-fused nets. *arXiv preprint arXiv:1605.07716*, 2016. 2

9

[38] Q. Wang, J. Zhang, S. Song, and Z. Zhang. Attentional neural network: Feature selection using cognitive feedback. In *NIPS*, pages 2033–2041, 2014. 2, 3

[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, July 2017. 1

[40] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016. 1, 2, 5, 6

[41] A. R. Zamir, T.-L. Wu, L. Sun, W. B. Shen, B. E. Shi, J. Malik, and S. Savarese. Feedback networks. In *CVPR*, July 2017. 2, 3

[42] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014. 8

[43] X. Zhang, Z. Li, C. Change Loy, and D. Lin. Polynet: A pursuit of structural diversity in very deep networks. In *CVPR*, July 2017. 1, 2