Convolutional Neural Networks with Alternately Updated Clique

Yibo Yang; Zhisheng Zhong; Tiancheng Shen; Zhouchen Lin

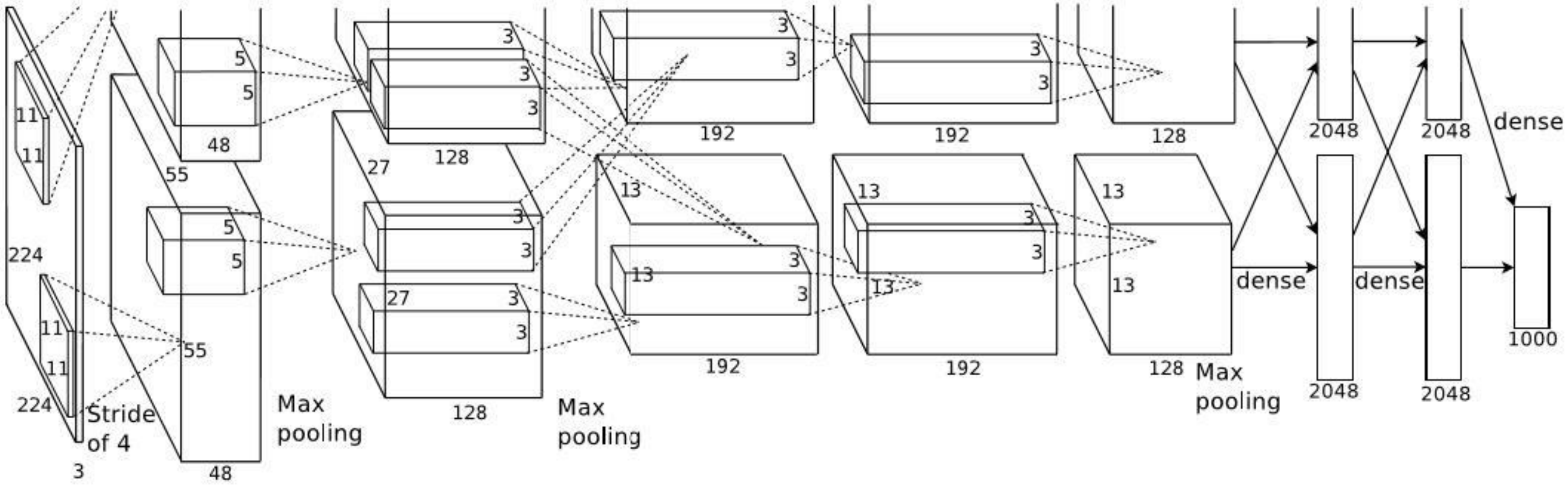
PEKING UNIVERSITY 1898

2018 CVPR
SALT LAKE CITY • JUNE 18-22

Background

Deeper and deeper

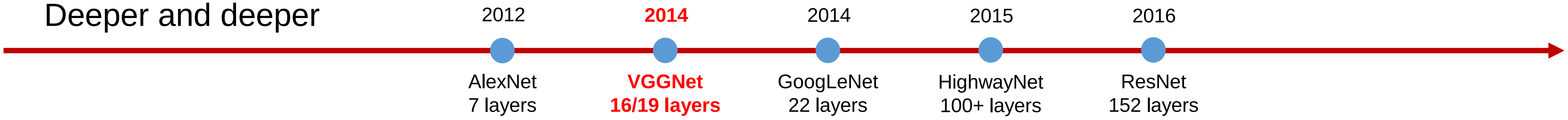
2012	2014	2014	2015	2016
AlexNet **7 layers**	VGGNet 16/19 layers	GoogLeNet 22 layers	HighwayNet 100+ layers	ResNet 152 layers



A. Krizhevsky, et al. Imagenet classification with deep convolutional neural networks. In NIPS, 2012

Background

Deeper and deeper

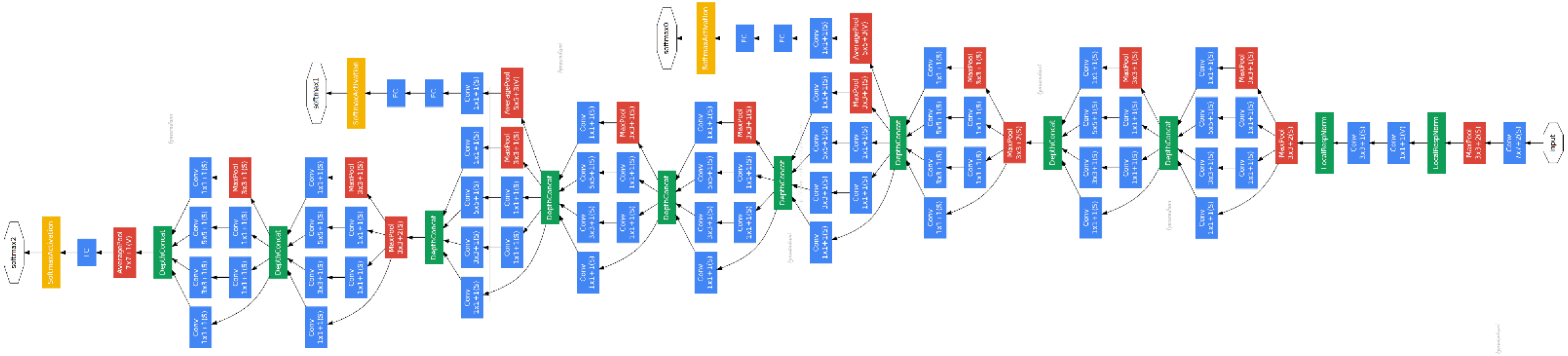


K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

Background

Deeper and deeper

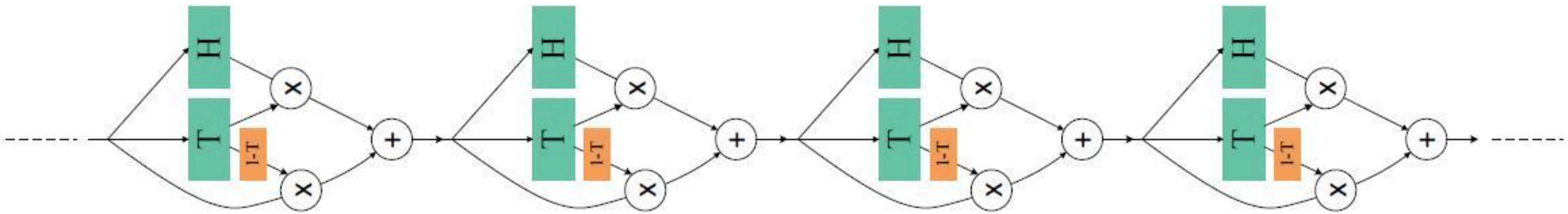
2012	2014	**2014**	2015	2016
AlexNet 7 layers	VGGNet 16/19 layers	**GoogLeNet 22 layers**	HighwayNet 100+ layers	ResNet 152 layers



Background

Deeper and deeper

2012	2014	2014	**2015**	2016
AlexNet 7 layers	VGGNet 16/19 layers	GoogLeNet 22 layers	**HighwayNet 100+ layers**	ResNet 152 layers

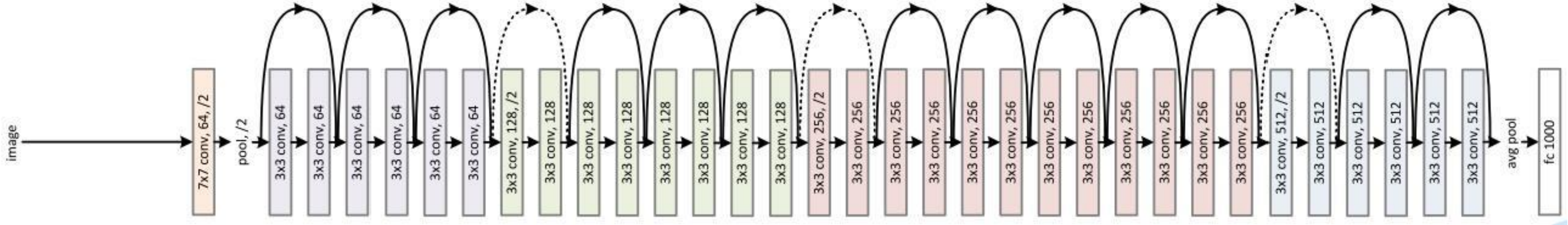


R. K. Srivastava, et al. Training very deep networks. In NIPS, 2015.

Background

Deeper and deeper

2012	2014	2014	2015	**2016**
AlexNet 7 layers	VGGNet 16/19 layers	GoogLeNet 22 layers	HighwayNet 100+ layers	**ResNet 152 layers**

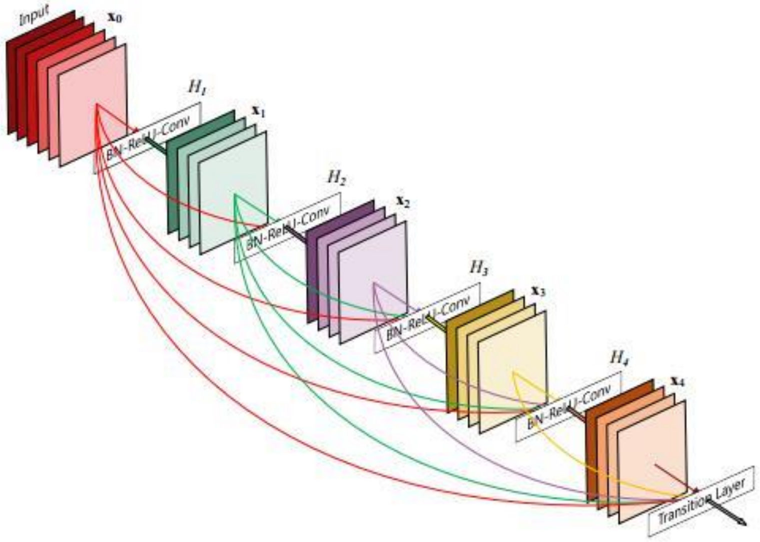


Skip Connections

K. He, et al. Deep residual learning for image recognition. In CVPR, 2016.

Background

DenseNet, 2017



- Propagation

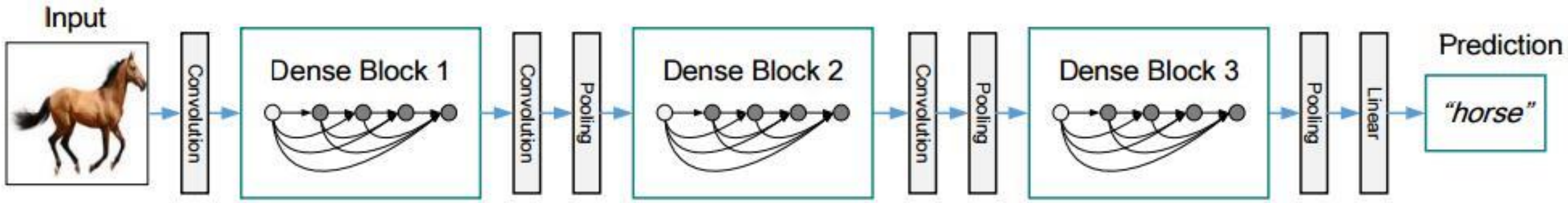
  ResNet: $x_l = H_l(x_{l-1}) + x_{l-1}$

  DenseNet: $x_l = H_l([x_0, x_1, \dots, x_{l-1}])$

- Bottleneck: BN-ReLU-Conv(1x1)-BN-ReLU-Conv(3x3)

- Compression: Reduce the number of feature maps in transition layers.

- Performance: State-of-the-art on CIFAR, SVHN datasets; parameter-efficient than ResNets.



G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In CVPR, 2017.

Motivations

- **How to further maximize information flow?**

Motivations

- **How to further maximize information flow?**
- **Does feedback connection help?**

Motivations

- Attention mechanism
  - Formulating an optimization problem (Cao et al. 2015)
  - Weighting the activations spatially or channel-wisely (Chen et al. 2017, Hu et al. 2017)
  - Feedback connections (Stollenga et al. 2014, Wang et al. 2014, Zamir et al. 2017)

Motivations

- Attention mechanism

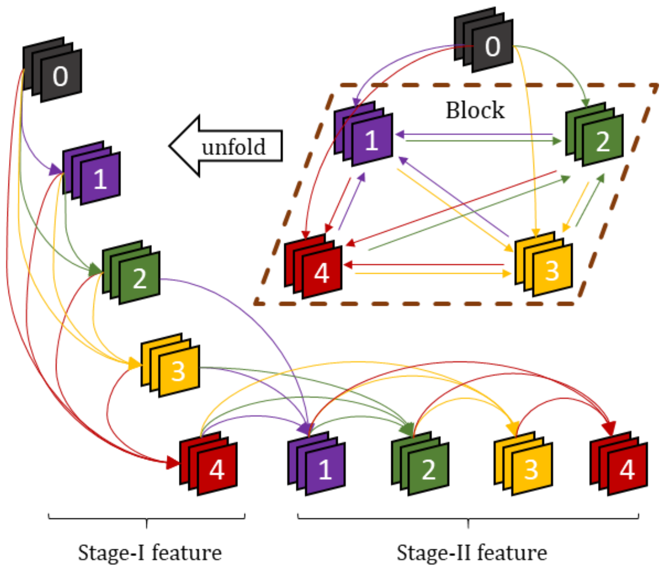
    - Formulating an optimization problem (Cao et al. 2015)
    - Weighting the activations spatially or channel-wisely (Chen et al. 2017, Hu et al. 2017)
    - **Feedback connections** (Stollenga et al. 2014, Wang et al. 2014, Zamir et al. 2017)

- Dense connections & Feedback connections

    - Bi-directionally connected -> Enable feature refinement at each update
    - Densely connected -> Information flow further maximized

Architectures

- Basic block
  - Any two layers have both forward and backward connections, and are both input and output of each other
  - More densely connected than DenseNets



Architectures

- Propagation

- First stage:

  Initialize layers using feedforward dense connections

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

Architectures

- Propagation

- First stage:

  Initialize layers using feedforward dense connections

- Second stage:

  Alternate updating rule

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

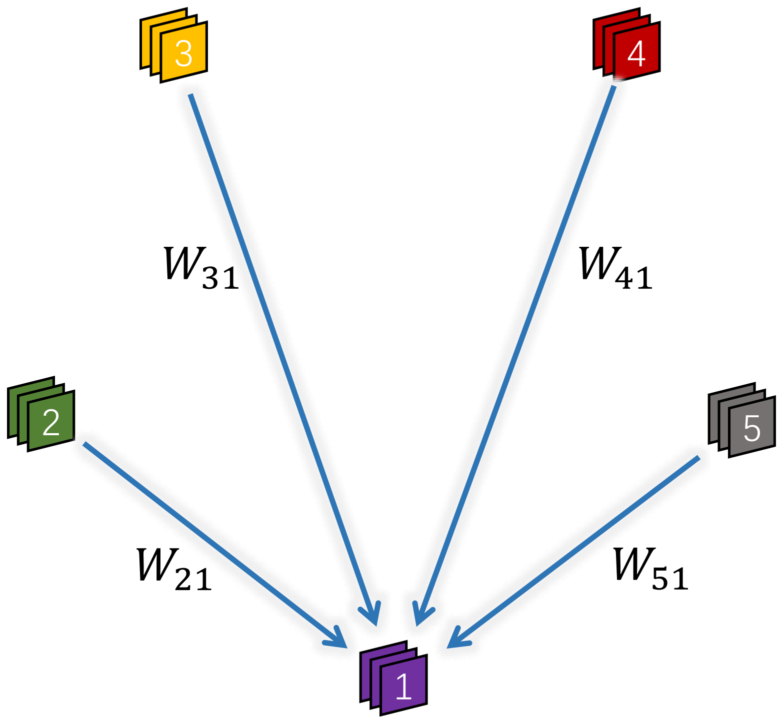
($i \geq 1$, $k \geq 2$, g: non-linear activation)

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

Architectures

- Propagation

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

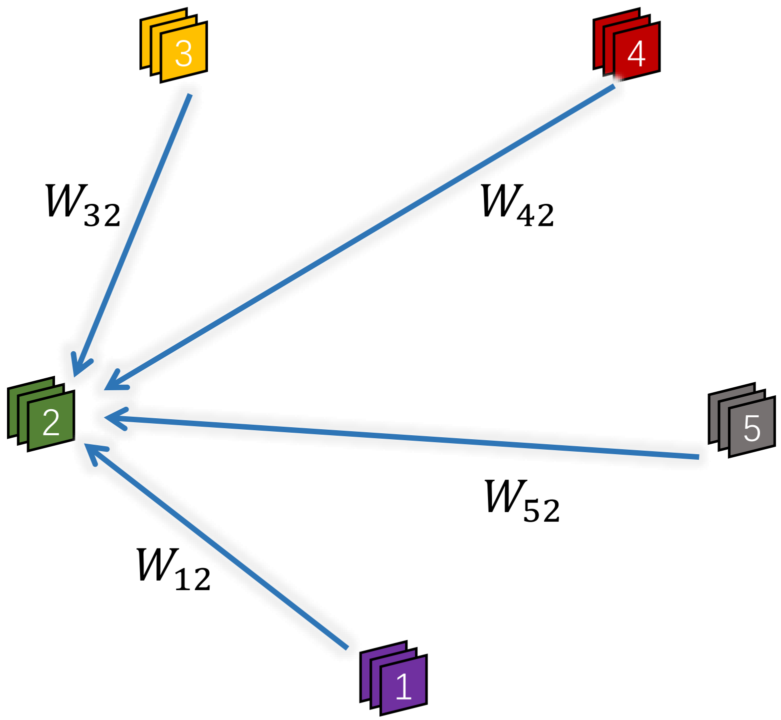


$$X_1^{(2)} = g(W_{21} * X_2^{(1)} + W_{31} * X_3^{(1)} + W_{41} * X_4^{(1)} + W_{51} * X_5^{(1)})$$

Architectures

- Propagation

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



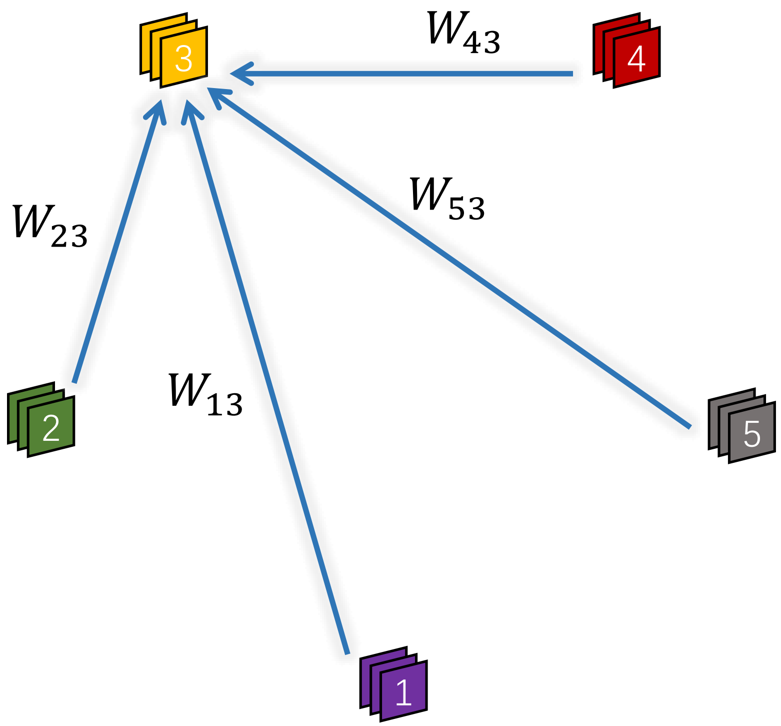
$$X_1^{(2)} = g(W_{21} * X_2^{(1)} + W_{31} * X_3^{(1)} + W_{41} * X_4^{(1)} + W_{51} * X_5^{(1)})$$

$$X_2^{(2)} = g(W_{12} * X_1^{(2)} + W_{32} * X_3^{(1)} + W_{42} * X_4^{(1)} + W_{52} * X_5^{(1)})$$

Architectures

- Propagation

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



$$X_1^{(2)} = g(W_{21} * X_2^{(1)} + W_{31} * X_3^{(1)} + W_{41} * X_4^{(1)} + W_{51} * X_5^{(1)})$$

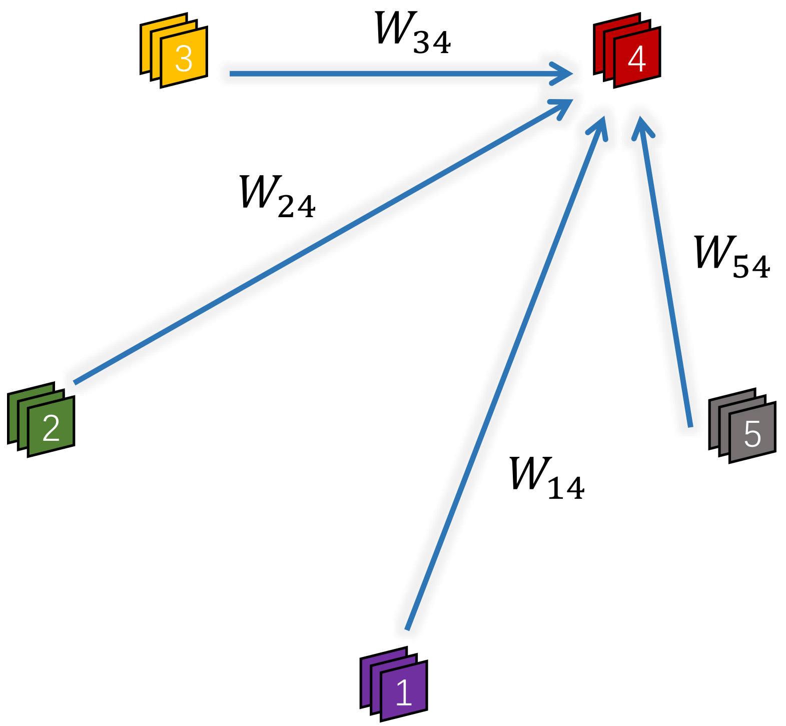
$$X_2^{(2)} = g(W_{12} * X_1^{(2)} + W_{32} * X_3^{(1)} + W_{42} * X_4^{(1)} + W_{52} * X_5^{(1)})$$

$$X_3^{(2)} = g(W_{13} * X_1^{(2)} + W_{23} * X_2^{(2)} + W_{43} * X_4^{(1)} + W_{53} * X_5^{(1)})$$

Architectures

- Propagation

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



$$X_1^{(2)} = g(W_{21} * X_2^{(1)} + W_{31} * X_3^{(1)} + W_{41} * X_4^{(1)} + W_{51} * X_5^{(1)})$$

$$X_2^{(2)} = g(W_{12} * X_1^{(2)} + W_{32} * X_3^{(1)} + W_{42} * X_4^{(1)} + W_{52} * X_5^{(1)})$$

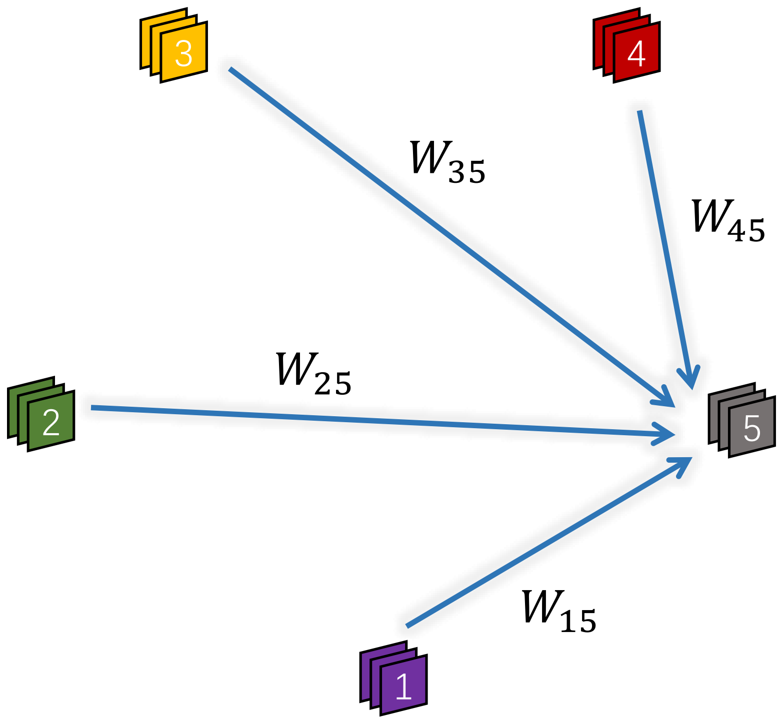
$$X_3^{(2)} = g(W_{13} * X_1^{(2)} + W_{23} * X_2^{(2)} + W_{43} * X_4^{(1)} + W_{53} * X_5^{(1)})$$

$$X_4^{(2)} = g(W_{14} * X_1^{(2)} + W_{24} * X_2^{(2)} + W_{34} * X_3^{(2)} + W_{54} * X_5^{(1)})$$

Architectures

- Propagation

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



$$X_1^{(2)} = g(W_{21} * X_2^{(1)} + W_{31} * X_3^{(1)} + W_{41} * X_4^{(1)} + W_{51} * X_5^{(1)})$$

$$X_2^{(2)} = g(W_{12} * X_1^{(2)} + W_{32} * X_3^{(1)} + W_{42} * X_4^{(1)} + W_{52} * X_5^{(1)})$$

$$X_3^{(2)} = g(W_{13} * X_1^{(2)} + W_{23} * X_2^{(2)} + W_{43} * X_4^{(1)} + W_{53} * X_5^{(1)})$$

$$X_4^{(2)} = g(W_{14} * X_1^{(2)} + W_{24} * X_2^{(2)} + W_{34} * X_3^{(2)} + W_{54} * X_5^{(1)})$$

$$X_5^{(2)} = g(W_{15} * X_1^{(2)} + W_{25} * X_2^{(2)} + W_{35} * X_3^{(2)} + W_{45} * X_4^{(2)})$$

Architectures

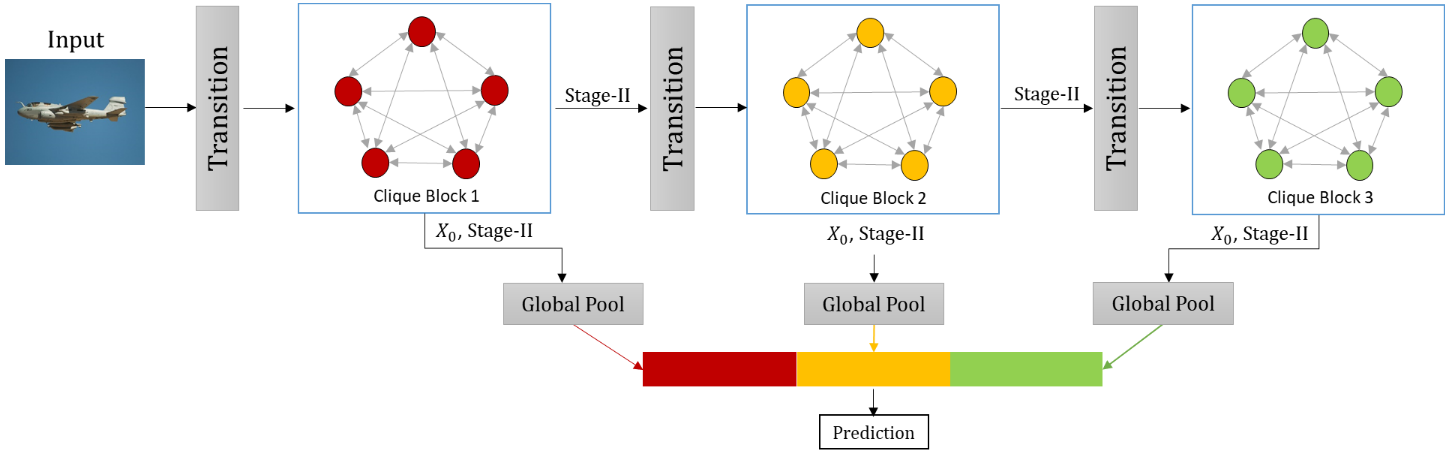
- Propagation

  - Enable top down refinement brought by each propagation in Stage II
  - Parameters keep re-used in different stages during propagation
  - We can have Stage III and even Stage IV feature without any parameter overhead.

Architectures

- Whole structure

  - A multi-scale way to compose the final representation.
  - Only transit Stage-II feature into the next block to avoid progressive increment of parameters and computation.
  - Multi-scale representation is more adopted in object detection or segmentation. Our multi-scale classification network is able to achieve stage-of-the-art image classification performance.



Experiments

- Ablation experiments

- CliqueNet (I+I):

  Only consider Stage-I feature

- CliqueNet (I+II):

  Use Stage-I feature to compose the final representation; transit Stage-II feature into the next block.

- CliqueNet (II+II):

  Use Stage-II feature to compose the final representation; also transit Stage-II feature into the next block.

Experiments

- Ablation experiments

- CliqueNet (I+I):

  Only consider Stage-I feature

- CliqueNet (I+II):

  Use Stage-I feature to compose the final representation; transit Stage-II feature into the next block.

- CliqueNet (II+II):

  Use Stage-II feature to compose the final representation; also transit Stage-II feature into the next block.

Model	Block feature	Transit	CIFAR-10
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

(There are 36 filters in each layer, 5 layers in each block.)

Experiments

- Ablation experiments

- CliqueNet-X

Use Stage-II feature, but only finish the first X steps in the stage II.
When X=0, it reduces to CliqueNet (I+I)
When X=5, it reduces to CliqueNet (II+II)

Model	CIFAR-10	CIFAR-100
CliqueNet (X=0)	5.83	24.79
CliqueNet (X=1)	5.63	24.65
CliqueNet (X=2)	5.54	24.37
CliqueNet (X=3)	5.41	23.75
CliqueNet (X=4)	5.20	24.04
CliqueNet (X=5)	5.12	23.73

(There are 64 filters in each layer, 5 layers in each block.)

Experiments

- Experiments on CIFAR and SVHN

  - Additional techniques

    Bottleneck

    Compression

    Attentional transition (SE module)

J. Hu, et al. Squeeze-and-excitation networks. In CVPR, 2018.

Experiments

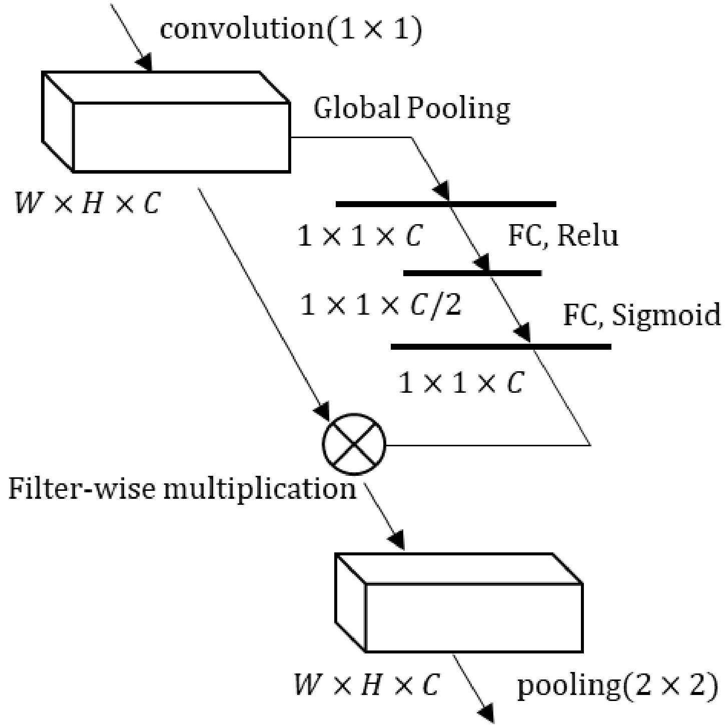
- Experiments on CIFAR and SVHN

  - Additional techniques

    Bottleneck

    Compression

    Attentional transition (SE module)



J. Hu, et al. Squeeze-and-excitation networks. In CVPR, 2018.

Experiments

- Experiments on CIFAR and SVHN

  - Additional techniques

    Bottleneck

    Compression

    Attentional transition (SE module)

  - Implementation details

    300 epochs on CIFAR and 40 epochs on SVHN

    0.1 leaning rate initially, divided by 10 at 50% and 75% of the training procedure

    Batchsize of 64 on CIFAR and SVHN

Experiments

- Experiments on CIFAR and SVHN

Model	A	B	C	FLOPs	Params	CIFAR-10	CIFAR-100	SVHN
Recurrent CNN [26]	-	-	-	-	1.86M	8.69	31.75	1.80
Stochastic Depth ResNet [18]	-	-	-	-	1.7M	11.66	37.8	1.75
dasNet [35]	-	-	-	-	-	9.22	33.78	-
FractalNet [25]	-	-	-	-	38.6M	7.33	28.2	1.87
DenseNet ($k = 12, T = 36$) [17]	-	-	-	0.53G	1.0M	7.00	27.55	1.79
DenseNet ($k = 12, T = 96$) [17]	-	-	-	3.54G	7.0M	5.77	23.79	1.67
DenseNet ($k = 24, T = 96$) [17]	-	-	-	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 ($k = 12, T = 96$) [17]	-	✓	✓	0.58G	0.8M	5.92	24.15	1.76
DenseNet ($k = 24, T = 246$) [17]	-	✓	✓	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**

(Error rates on CIFAR and SVHN without any data augmentation. A, B, and C represents attentional transition, bottleneck, and compression, respectively. k is the number of filters per layer, and T is the total number of layers in three blocks.)

Experiments

- Experiments on CIFAR and SVHN

Model	A	B	C	FLOPs	Params	CIFAR-10	CIFAR-100	SVHN
Recurrent CNN [26]	-	-	-	-	1.86M	8.69	31.75	1.80
Stochastic Depth ResNet [18]	-	-	-	-	1.7M	11.66	37.8	1.75
dasNet [35]	-	-	-	-	-	9.22	33.78	-
FractalNet [25]	-	-	-	-	38.6M	7.33	28.2	1.87
DenseNet ($k = 12, T = 36$) [17]	-	-	-	0.53G	1.0M	7.00	27.55	1.79
DenseNet ($k = 12, T = 96$) [17]	-	-	-	3.54G	7.0M	5.77	23.79	1.67
DenseNet ($k = 24, T = 96$) [17]	-	-	-	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 ($k = 12, T = 96$) [17]	-	✓	✓	0.58G	0.8M	5.92	24.15	1.76
DenseNet ($k = 24, T = 246$) [17]	-	✓	✓	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**

(Error rates on CIFAR and SVHN without any data augmentation. A, B, and C represents attentional transition, bottleneck, and compression, respectively. k is the number of filters per layer, and T is the total number of layers in three blocks.)

Experiments

- Experiments on ImageNet

Model	Params	Top-1	Top-5
ResNet-18	11.7M	30.43	10.76
CliqueNet-S0	5.7M	27.52	8.98
ResNet-34	21.8M	26.73	8.74
CliqueNet-S1	7.96M	26.21	8.3
DenseNet-121	7.98M	25.02	7.71
CliqueNet-S2	11M	24.82	7.51
ResNet-50	25.6M	24.01	7.02
CliqueNet-S3	14.38M	24.01	7.15

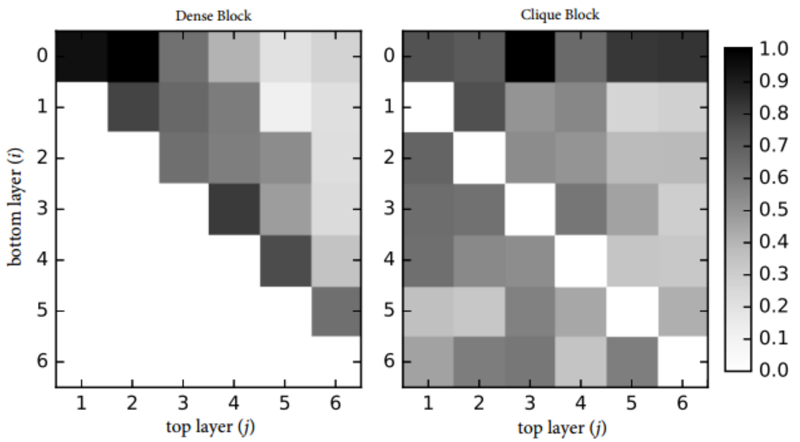
Layer	S0	S1	S2	S3
Convolution (112×112)	conv (7×7), 64, stride 2			
Pooling (56×56)	max pool (3×3), stride 2			
Block 1 (56×56)	36×5	36×5	36×5	40×6
Transition: conv (1×1), avg pool (2×2)				
Block 2 (28×28)	64×6	80×6	80×5	80×6
Transition: conv (1×1), avg pool (2×2)				
Block 3 (14×14)	100×6	120×6	150×6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

(The first number in each block is the number of filters per layer, and the second denotes the number of layers in this block)

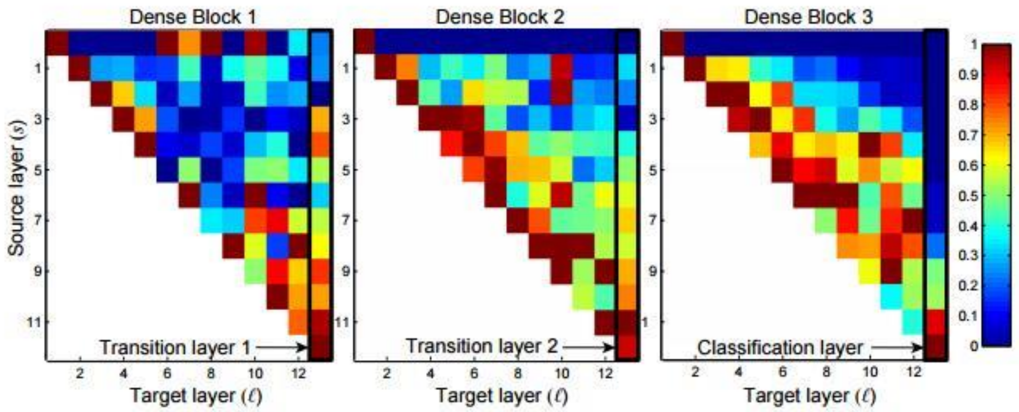
Discussions

- Parameter efficiency

  - The strong weights in DenseNets are along the diagonal.
  - The strong weights in CliqueNets are distributed more evenly.
  - Multi-scale structure enables our parameter efficiency.



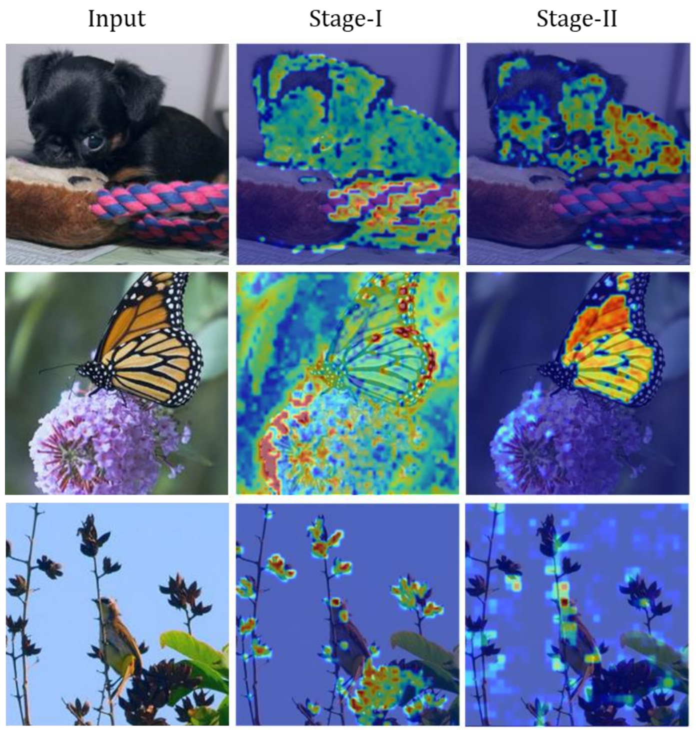
(Our visualization)



(Visualization in DenseNet paper)

Discussions

- Parameter efficiency
  - The strong weights in DenseNets are along the diagonal.
  - The strong weights in CliqueNets are distributed more evenly.
  - Multi-scale structure enables our parameter efficiency.
- Feature refinement
  - Spatial attention is achieved to focus the activations into the target region.
  - Feedback connections and alternate updating enable feature refinement.



Thanks for your attention!

Discussions

- Parameter efficiency
  - The strong weights in DenseNets are along the diagonal.
  - The strong weights in CliqueNets are distributed more evenly.
  - Multi-scale structure enables our parameter efficiency.
- Feature refinement
  - Spatial attention is achieved to focus the activations into the target region.
  - Feedback connections and alternate updating enable feature refinement.