Advanced CNN Architectures

Akshay Mishra, Hong Cheng

CNNs are everywhere.

- **Recommendation Systems**
- Drug Discovery
- Physics simulations

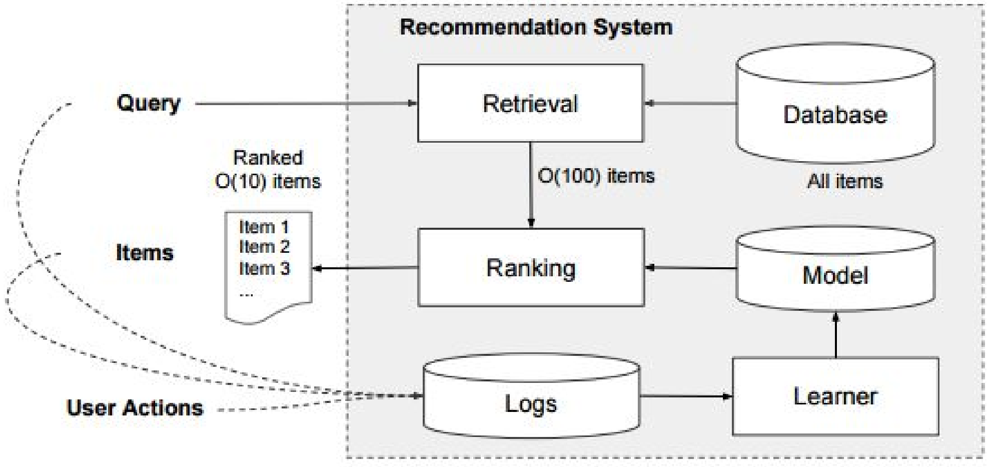


Figure 2: Overview of the recommender system.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah, Wide & Deep Learning for Recommender Systems, arxiv 2016

CNNs are everywhere...

- Recommendation Systems
- **Drug Discovery**
- Physics simulations

Figure 5: Sulfonyl/sulfonamide detection with autonomously trained convolutional filters.

Izhar Wallach, Michael Dzamba, Abraham Heifets, AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery, arxiv 2016

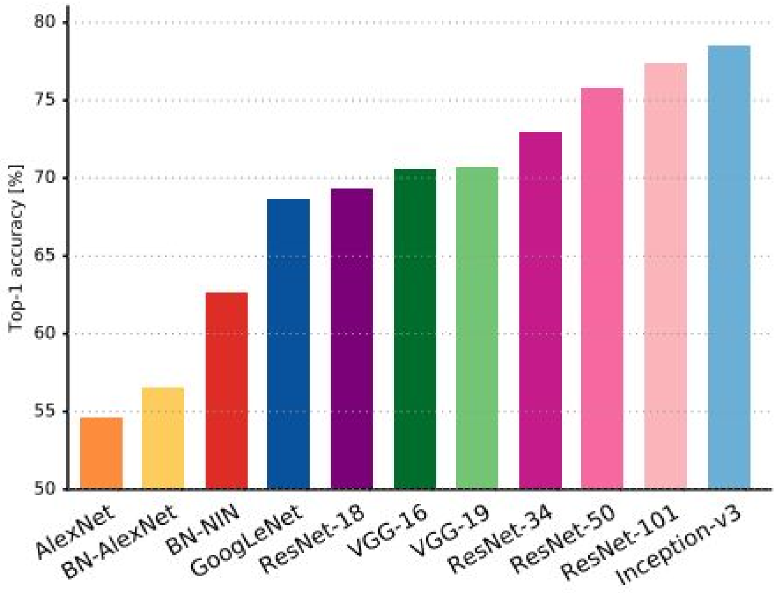
CNNs are everywhere...

- Recommendation Systems
- Drug Discovery
- **Physics simulations**

Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, Ken Perlin, Accelerating Eulerian Fluid Simulation With Convolutional Networks, arxiv 2016

We're focusing on ImageNet

- **Gives us a common task to compare architectures**
- Networks trained on ImageNet are often starting points for other vision tasks
- Architectures that perform well on ImageNet have been successful in other domains



Alfredo Canziani & Eugenio Culurciello, An Analysis of Deep Neural Network Models for Practical Applications, arXiv 2016

We’re focusing on ImageNet

- Gives us a common task to compare architectures
- **Networks trained on ImageNet are often starting points for other vision tasks**
- Architectures that perform well on ImageNet have been successful in other domains

Example applications:

- Object detection
- Action recognition
- Human pose estimation
- Semantic segmentation
- Image captioning

We’re focusing on ImageNet

- Gives us a common task to compare architectures
- Networks trained on ImageNet are often starting points for other vision tasks
- **Architectures that perform well on ImageNet have been successful in other domains**

Novel ResNet Applications:

- Volumetric Brain Segmentation (VoxResNet)
- City-Wide Crowd Flow Prediction: (ST-ResNet)
- Generating Realistic Voices (WaveNet)

Overview

We’ve organized our presentation into three stages:

1. **A more detailed coverage of the building blocks of CNNs**
2. Attempts to explain how and why Residual Networks work
3. Survey extensions to ResNets and other notable architectures

Topics covered:

- Alternative activation functions
- Relationship between fully connected layers and convolutional layers
- Ways to convert fully connected layers to convolutional layers
- Global Average Pooling

Overview

We've organized our presentation into three stages:

1. A more detailed coverage of the building blocks of CNNs
2. **Attempts to explain how and why Residual Networks work**
3. Survey extensions to ResNets and other notable architectures

Topics covered:

- ResNets as implicit ensembles
- ResNets as learning iterative refinements
- Connections to recurrent networks and the brain

Overview

We've organized our presentation into three stages:

1. A more detailed coverage of the building blocks of CNNs
2. Attempts to explain how and why Residual Networks work
3. **Survey extensions to ResNets and other notable architectures**

Motivation:

- Many architectures using residuals
  - WaveNets
  - InceptionResNet
  - XceptionNet
- Tweaks can further improve performance

Overview

We’ve organized our presentation into three stages:

1. A more detailed coverage of the building blocks of CNNs
2. Attempts to explain how and why Residual Networks work
3. **Survey extensions to ResNets and other notable architectures**

Motivation:

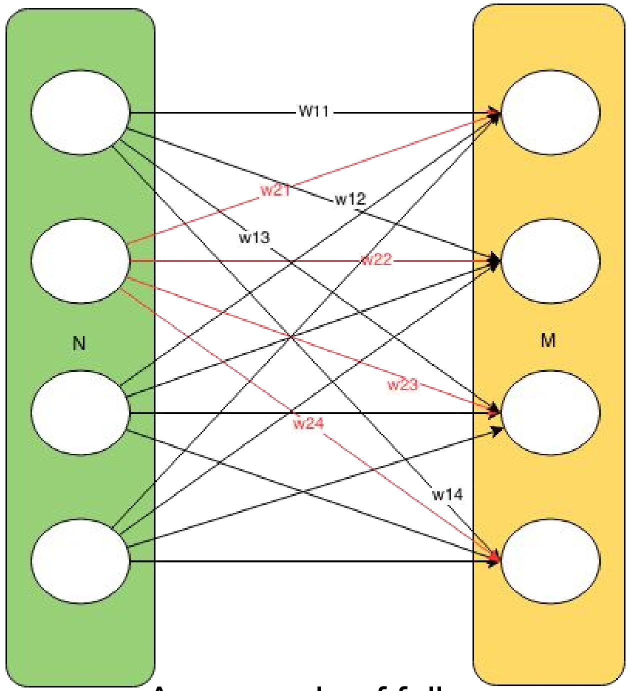
- Get a sense of what people have tried
- Show that residuals aren’t necessary for state-of-the art results
- By doing this towards the end, we can point out interesting connections to ResNets

Stage 1: Revisiting the Basics

- Alternative activation functions
- Relationship between fully connected layers and convolutional layers
- Ways to convert fully connected layers to convolutional layers
- Global Average Pooling

Review: Fully Connected Layers

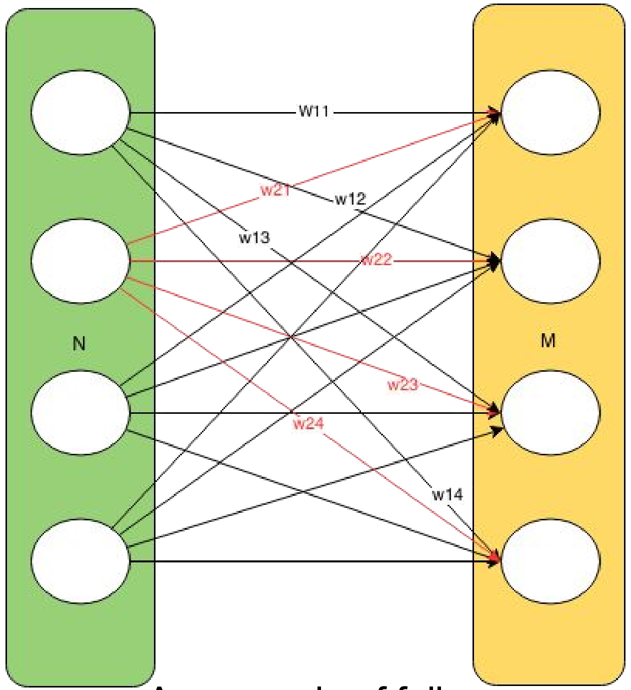
- **Takes N inputs, and outputs M units**
- Each output is a linear combination of inputs
- Usually implemented as multiplication by an (N x M) matrix



An example of fully connected layer.

Review: Fully Connected Layers

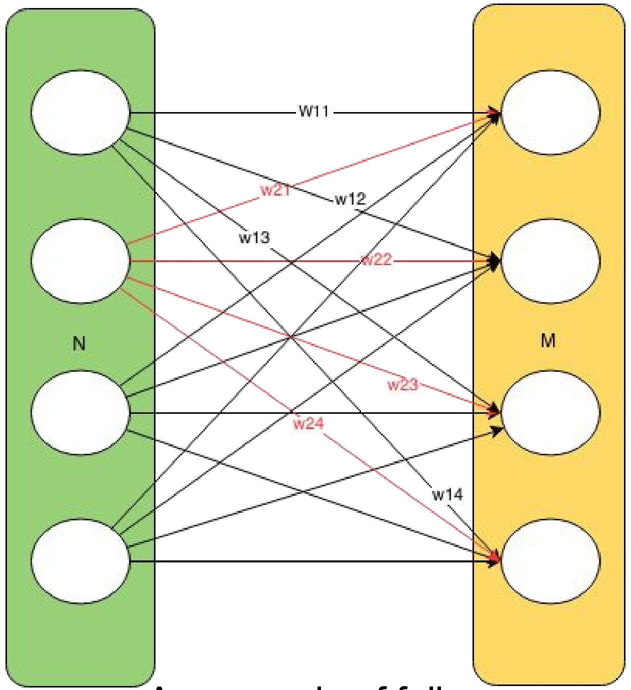
- Takes N inputs, and outputs M units
- **Each output is a linear combination of inputs**
- Usually implemented as multiplication by an (N x M) matrix



An example of fully connected layer.

Review: Fully Connected Layers

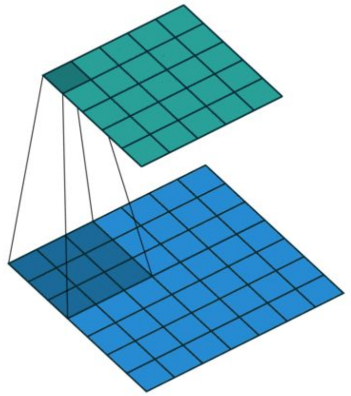
- Takes N inputs, and outputs M units
- Each output is a linear combination of inputs
- **Usually implemented as multiplication by an (N x M) matrix**



An example of fully connected layer.

How fully connected layers fix input size

- **Convolutions can be thought of as sliding fully connected layers**
- When the inputs to a convolutional layer are larger feature maps, outputs are larger feature maps
- Fully connected layers have a fixed number of inputs/outputs, forcing the entire network's input shape to be fixed

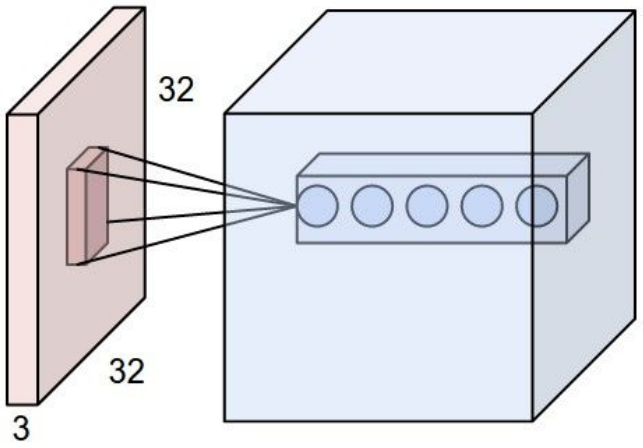


Think of this as fully connected layer that takes n x n inputs sliding across the input

Image source: http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html

How fully connected layers fix input size

- **Convolutions can be thought of as sliding fully connected layers**
- When the inputs to a convolutional layer are larger feature maps, outputs are larger feature maps
- Fully connected layers have a fixed number of inputs/outputs, forcing the entire network's input shape to be fixed

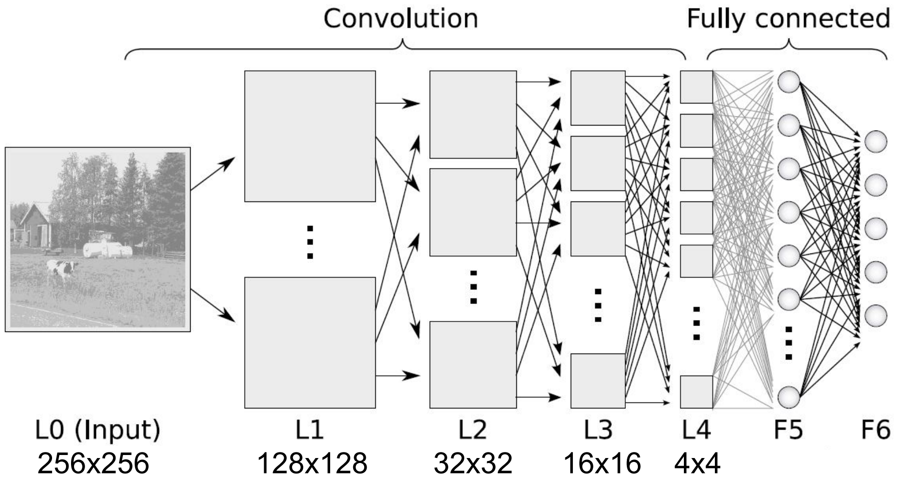


The number of output feature maps corresponds to the number of outputs of this fully connected layer

Image source:
http://cs231n.github.io/convolutional-networks/

How fully connected layers fix input size

- Convolutions can be thought of as sliding fully connected layers
- **When the inputs to a convolutional layer are larger feature maps, outputs are larger feature maps**
- Fully connected layers have a fixed number of inputs/outputs, forcing the entire network's input shape to be fixed

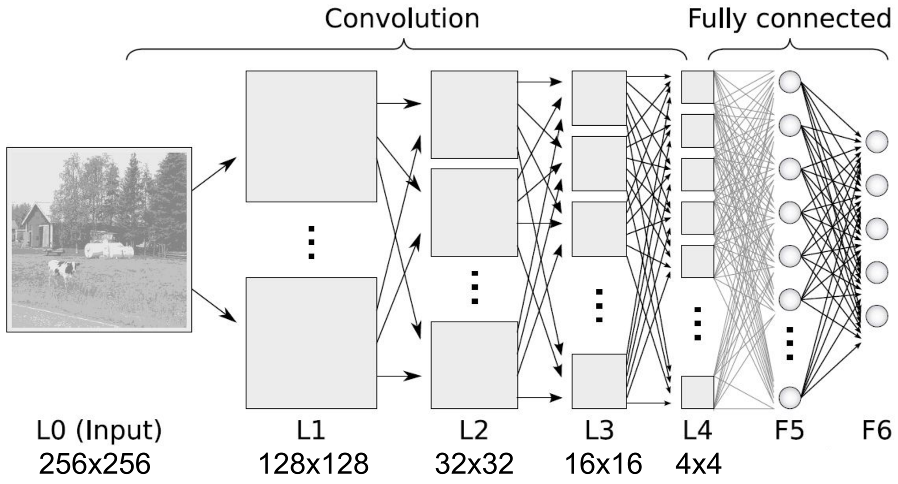


What are the spatial resolutions of feature maps if we input a 512 x 512 images?

Image source: http://www.ais.uni-bonn.de/deep_learning/

How fully connected layers fix input size

- Convolutions can be thought of as sliding fully connected layers
- **When the inputs to a convolutional layer are larger feature maps, outputs are larger feature maps**
- Fully connected layers have a fixed number of inputs/outputs, forcing the entire network's input shape to be fixed

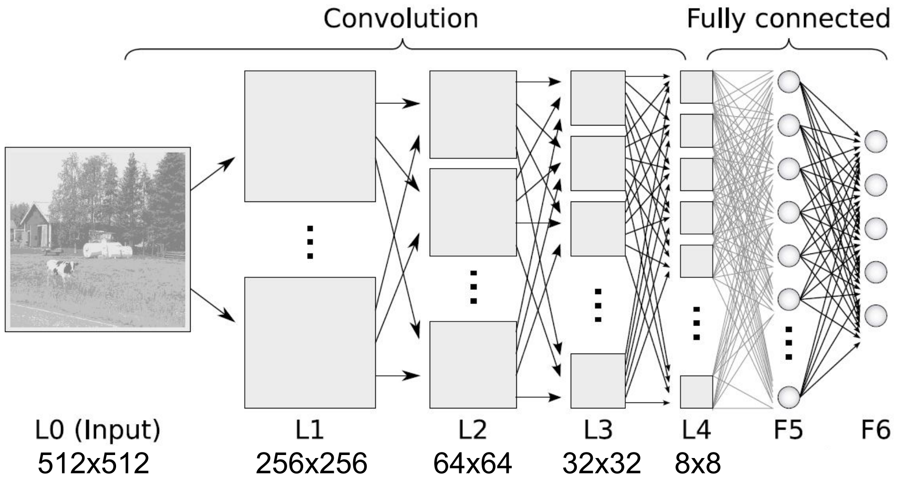


What are the spatial resolutions of feature maps if we input a 512 x 512 images?

Image source: http://www.ais.uni-bonn.de/deep_learning/

How fully connected layers fix input size

- Convolutions can be thought of as sliding fully connected layers
- **When the inputs to a convolutional layer are larger feature maps, outputs are larger feature maps**
- Fully connected layers have a fixed number of inputs/outputs, forcing the entire network's input shape to be fixed

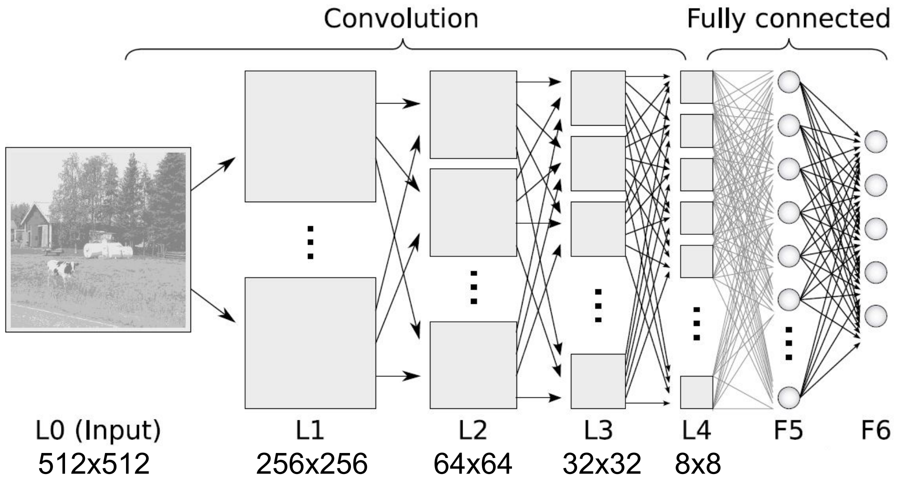


The spatial resolutions are doubled in both dimensions for all feature maps.

Image source: http://www.ais.uni-bonn.de/deep_learning/

How fully connected layers fix input size

- Convolutions can be thought of as sliding fully connected layers
- When the inputs to a convolutional layer are larger feature maps, outputs are larger feature maps
- **Fully connected layers have a fixed number of inputs/outputs, forcing the entire network's input shape to be fixed**

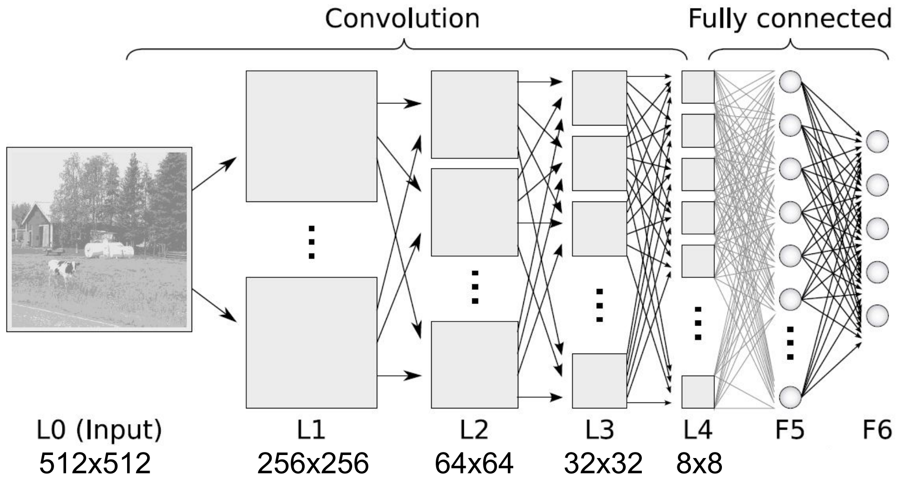


What happens to to the fully connected layers when input dimensions are doubled?

Image source: http://www.ais.uni-bonn.de/deep_learning/

How fully connected layers fix input size

- Convolutions can be thought of as sliding fully connected layers
- When the inputs to a convolutional layer are larger feature maps, outputs are larger feature maps
- **Fully connected layers have a fixed number of inputs/outputs, forcing the entire network's input shape to be fixed**



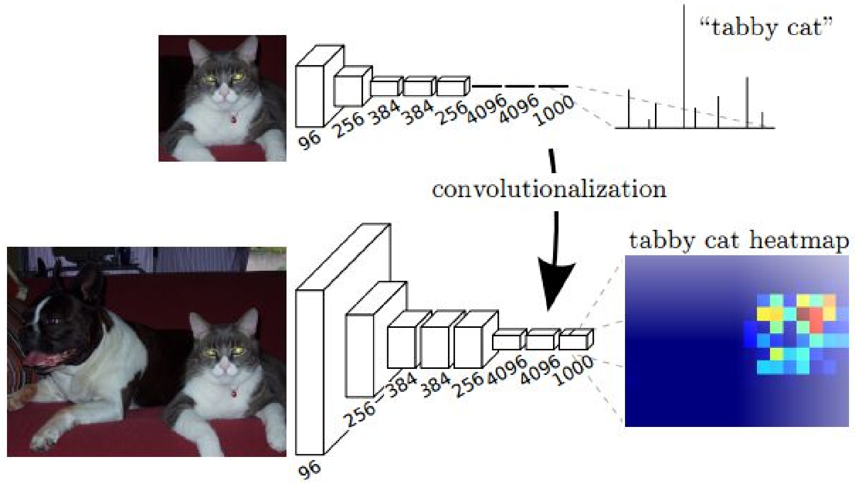
It becomes unclear how the larger feature maps should feed into the fully connected layers.

Image source: http://www.ais.uni-bonn.de/deep_learning/

Fully Connected Layers => Convolutions

Based on the relationships between fully connected layers and convolutions we just discussed, can you think of a way to *convert* fully connected layers to convolutions?

- By replacing fully connected layers with convolutions, we will be able to output heatmaps of class probabilities



Jonathan Long, Evan Shelhamer, Trevor Darrell, Fully Convolutional Networks for Semantic Segmentation, arXiv preprint 2016

Convolutionize VGG-Net

- **VGG-Net's final pooling layer outputs 7x7 feature maps**
- **VGG-Net's first fully connected layer ouputs 4096 units**
- What should the spatial size of the convolutional kernel be?
- How many output feature maps should we have?

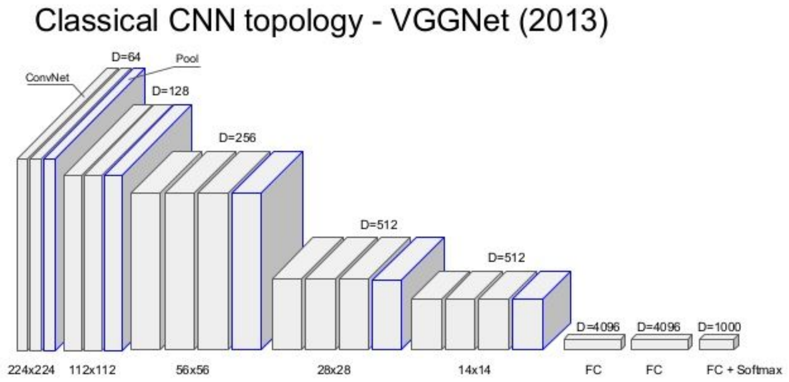


Image source:
http://www.robots.ox.ac.uk/~vgg/research/very_deep/

Convolutionize VGG-Net

- VGG-Net's final pooling layer outputs 7x7 feature maps
- VGG-Net's first fully connected layer ouputs 4096 units
- **What should the spatial size of the convolutional kernel be?**
- How many output feature maps should we have?

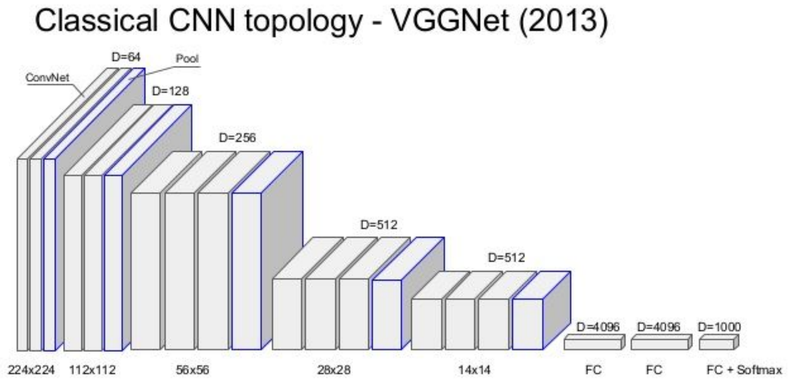
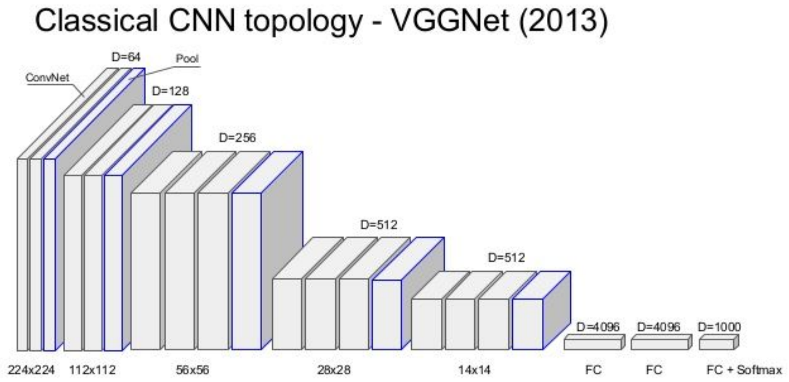


Image source:
http://www.robots.ox.ac.uk/~vgg/research/very_deep/

Convolutionize VGG-Net

- VGG-Net's final pooling layer outputs 7x7 feature maps
- VGG-Net's first fully connected layer ouputs 4096 units
- **What should the spatial size of the convolutional kernel be?**
- How many output feature maps should we have?

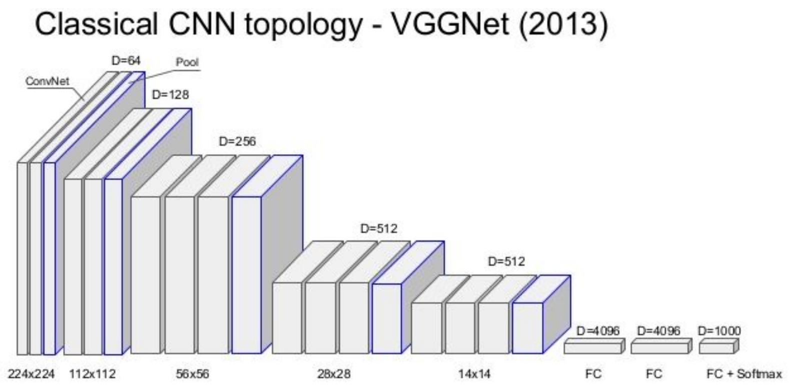


- Convolutional kernel should be 7 x 7 with no padding

Image source:
http://www.robots.ox.ac.uk/~vgg/research/very_deep/

Convolutionize VGG-Net

- VGG-Net's final pooling layer outputs 7x7 feature maps
- VGG-Net's first fully connected layer ouputs 4096 units
- What should the spatial size of the convolutional kernel be?
- **How many output feature maps should we have?**

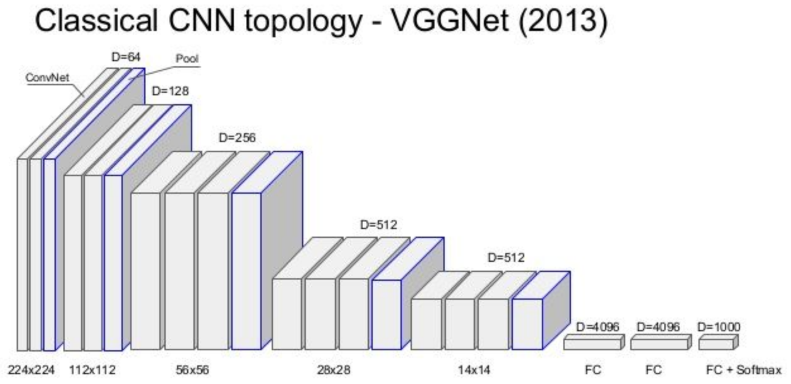


- Convolutional kernel should be 7 x 7 with no padding

Image source: http://www.robots.ox.ac.uk/~vgg/research/very_deep/

Convolutionize VGG-Net

- VGG-Net’s final pooling layer outputs 7x7 feature maps
- VGG-Net’s first fully connected layer ouputs 4096 units
- What should the spatial size of the convolutional kernel be?
- **How many output feature maps should we have?**

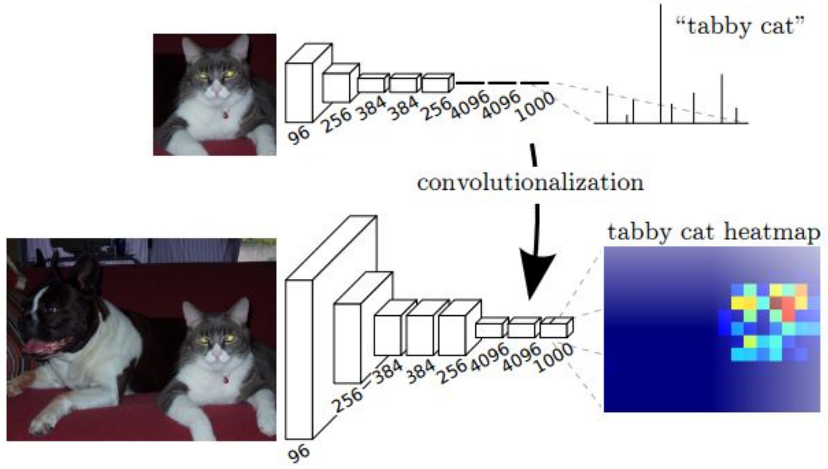


- Convolutional kernel should be 7 x 7 with no padding to correspond to a non-sliding fully connected layer
- There should be 4096 output feature maps to correspond to each of the fully connected layers outputs

Image source: http://www.robots.ox.ac.uk/~vgg/research/very_deep/

Convolutionize VGG-Net

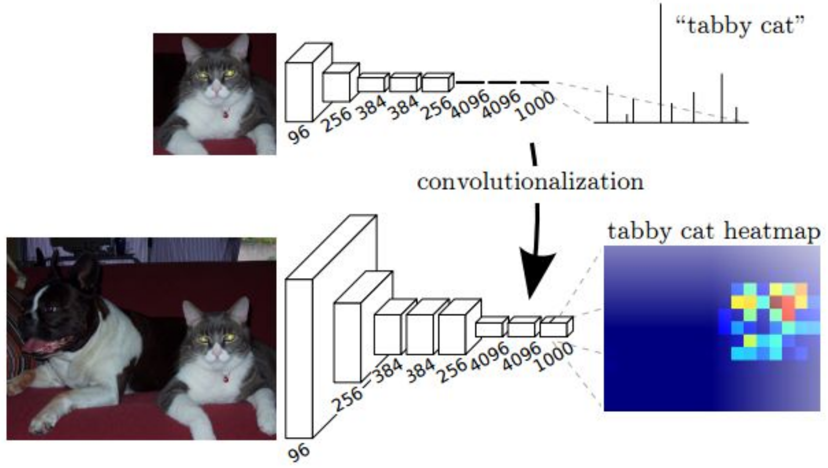
- What just happened?
- The final pooling layer still outputs 7x7 feature maps
- But the first fully connected layer has been replaced by a 7x7 convolution outputting 4096 feature maps
- The spatial resolution of these feature maps is 1x1
- First and hardest step towards convolutionalization is complete!



Jonathan Long, Evan Shelhamer, Trevor Darrell, [Fully Convolutional Networks for Semantic Segmentation](), arXiv preprint 2016

Convolutionize VGG-Net

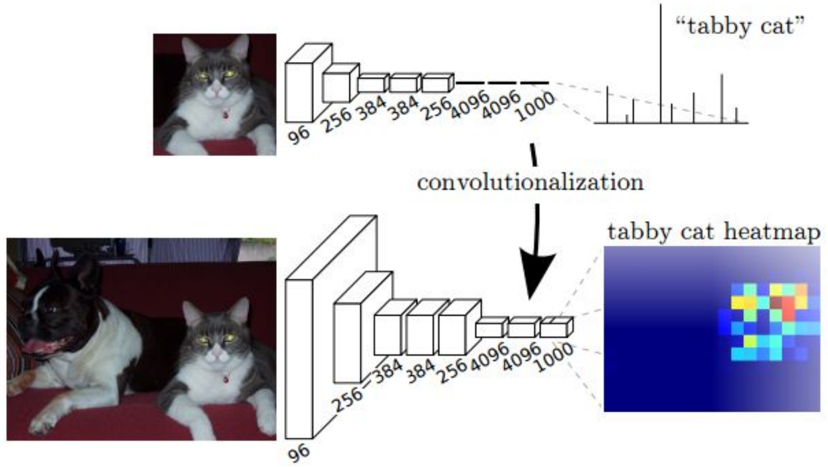
- How do we convolutionalize the second fully connected layer?
- The input to this layer is 1x1 with 4096 feature maps
- What is the spatial resolution of the convolution used?
- How many output feature maps should be used?



Jonathan Long, Evan Shelhamer, Trevor Darrell, Fully Convolutional Networks for Semantic Segmentation, arXiv preprint 2016

Convolutionize VGG-Net

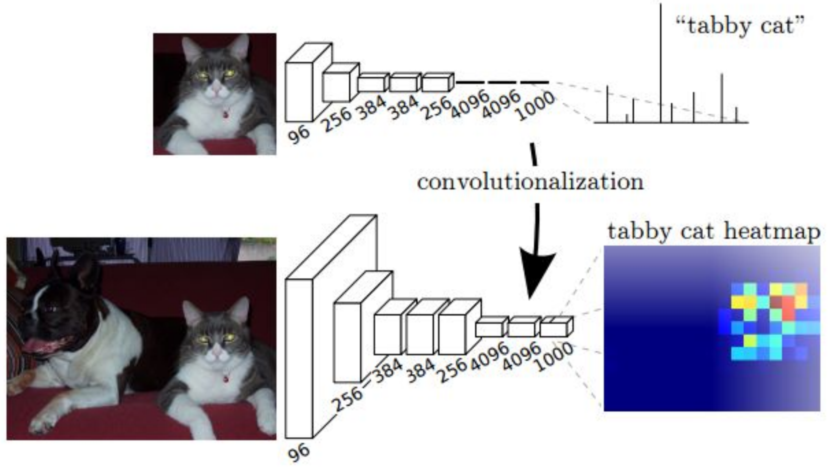
- How do we convolutionalize the second fully connected layer?
- The input to this layer is 1x1 with 4096 feature maps
- What is the spatial resolution of the convolution used?
- How many output feature maps should be used?
- Same idea used for the final fully connected layer



Jonathan Long, Evan Shelhamer, Trevor Darrell, Fully Convolutional Networks for Semantic Segmentation, arXiv preprint 2016

Results

- Now, all the fully connected layers have been replaced with convolutions
- When larger inputs are fed into the network, network outputs grid of values
- The grid can be interpreted as class conditional heatmaps



Jonathan Long, Evan Shelhamer, Trevor Darrell, Fully Convolutional Networks for Semantic Segmentation, arXiv preprint 2016

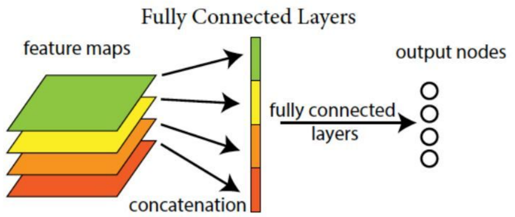
Global Average Pooling

Take the average of each feature map and feed the resulting vector directly into the softmax layer

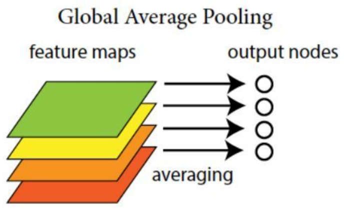
Advantages:

1) More native to the convolutional structure
2) No parameter to optimize. Overfitting is avoided at this layer.
3) More robust to spatial translations of input
4) Allows for flexibility in input size

CNN



NIN

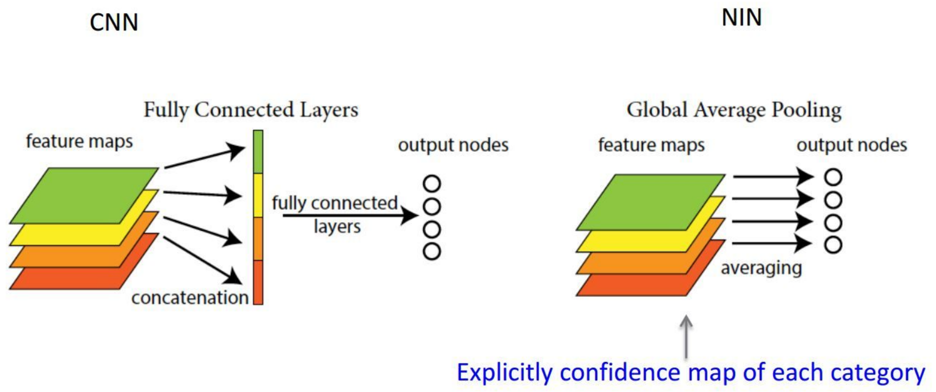


Explicitly confidence map of each category

Min Lin, Qiang Chen, Shuicheng Yan
Network In Network

Global Average Pooling

- In practice, the global average pooling outputs aren't sent directly to softmax
- It's more common to send the filter wise averages to a fully connected layer before softmax
- Used in some top performing architectures including ResNets and InceptionNets



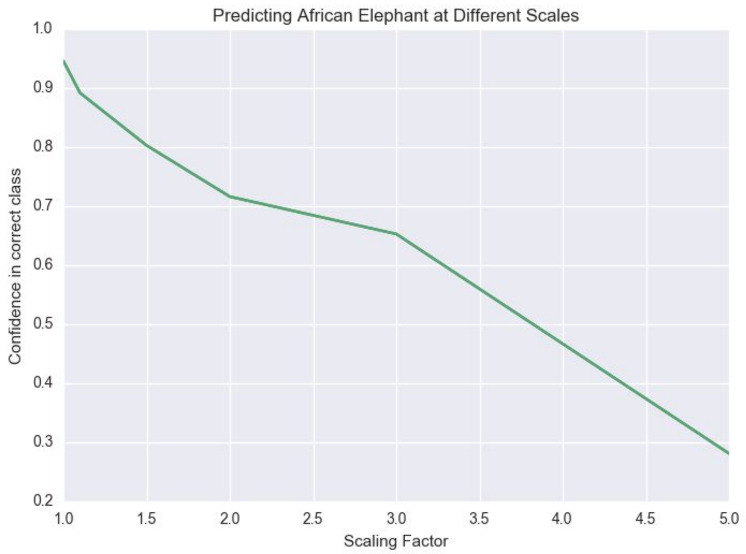
Min Lin, Qiang Chen, Shuicheng Yan Network In Network

Rescaling Demo:

- I fed this picture of an elephant to ResNet-50 at various scales
- ResNet was trained on 224x224 images
- How much bigger can I make the image before the elephant is misclassified?

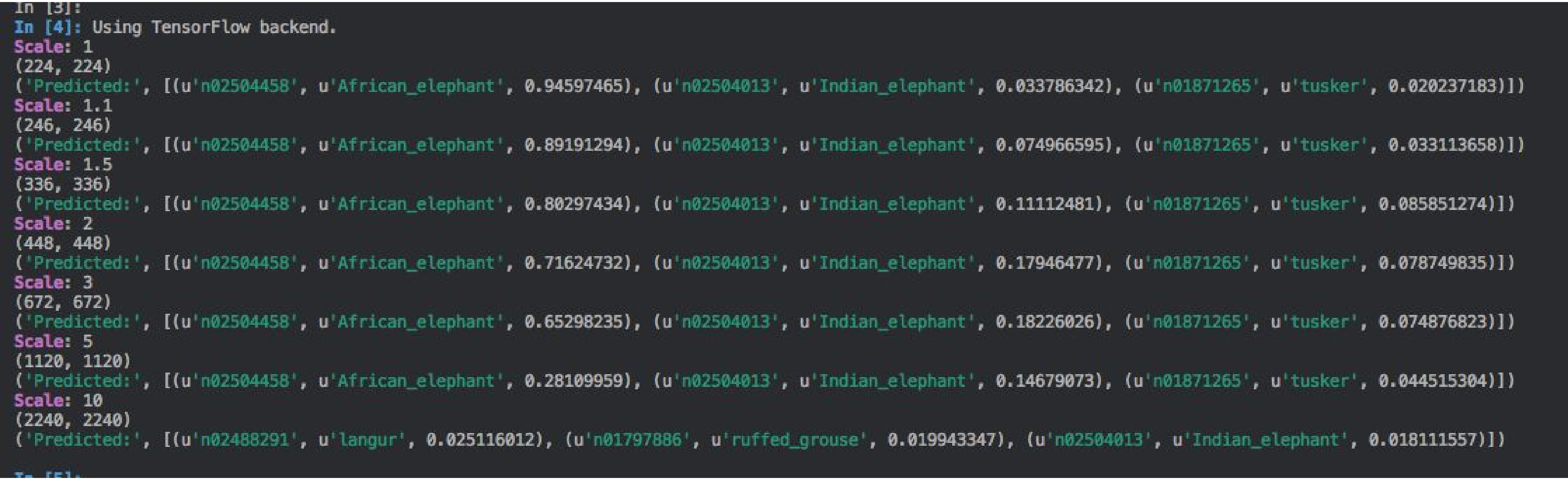
Rescaling Demo:

- I tried rescales of [1.1, 1.5,3,5,10]
- Elephant was correctly classified up till 5x scaling
  - Input size was 1120x1120
- Confidence of classification decays slowly
- At rescale factor of 10, ‘African Elephant’ is no longer in the top 3



Rescaling Demo:

- Raw predictions:



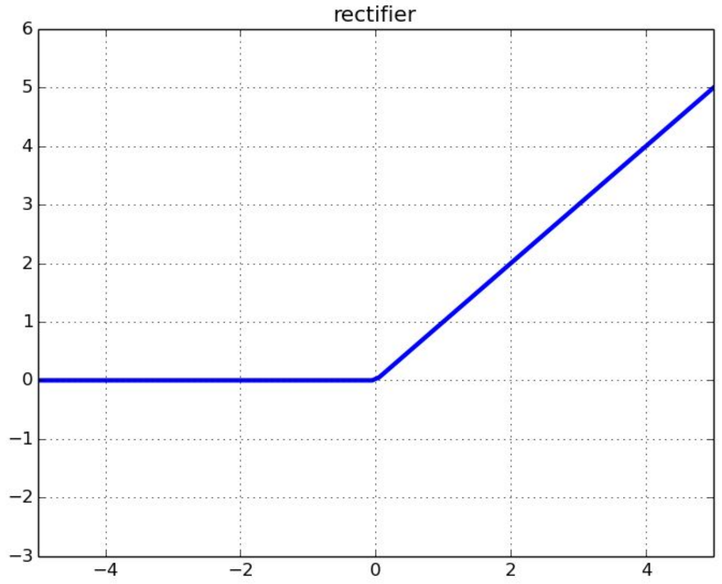
```
In [3]:
In [4]: Using TensorFlow backend.
Scale: 1
(224, 224)
('Predicted:', [(u'n02504458', u'African_elephant', 0.94597465), (u'n02504013', u'Indian_elephant', 0.033786342), (u'n01871265', u'tusker', 0.020237183)])
Scale: 1.1
(246, 246)
('Predicted:', [(u'n02504458', u'African_elephant', 0.89191294), (u'n02504013', u'Indian_elephant', 0.074966595), (u'n01871265', u'tusker', 0.033113658)])
Scale: 1.5
(336, 336)
('Predicted:', [(u'n02504458', u'African_elephant', 0.80297434), (u'n02504013', u'Indian_elephant', 0.11112481), (u'n01871265', u'tusker', 0.085851274)])
Scale: 2
(448, 448)
('Predicted:', [(u'n02504458', u'African_elephant', 0.71624732), (u'n02504013', u'Indian_elephant', 0.17946477), (u'n01871265', u'tusker', 0.078749835)])
Scale: 3
(672, 672)
('Predicted:', [(u'n02504458', u'African_elephant', 0.65298235), (u'n02504013', u'Indian_elephant', 0.18226026), (u'n01871265', u'tusker', 0.074876823)])
Scale: 5
(1120, 1120)
('Predicted:', [(u'n02504458', u'African_elephant', 0.28109959), (u'n02504013', u'Indian_elephant', 0.14679073), (u'n01871265', u'tusker', 0.044515304)])
Scale: 10
(2240, 2240)
('Predicted:', [(u'n02488291', u'langur', 0.025116012), (u'n01797886', u'ruffed_grouse', 0.019943347), (u'n02504013', u'Indian_elephant', 0.018111557)])
```

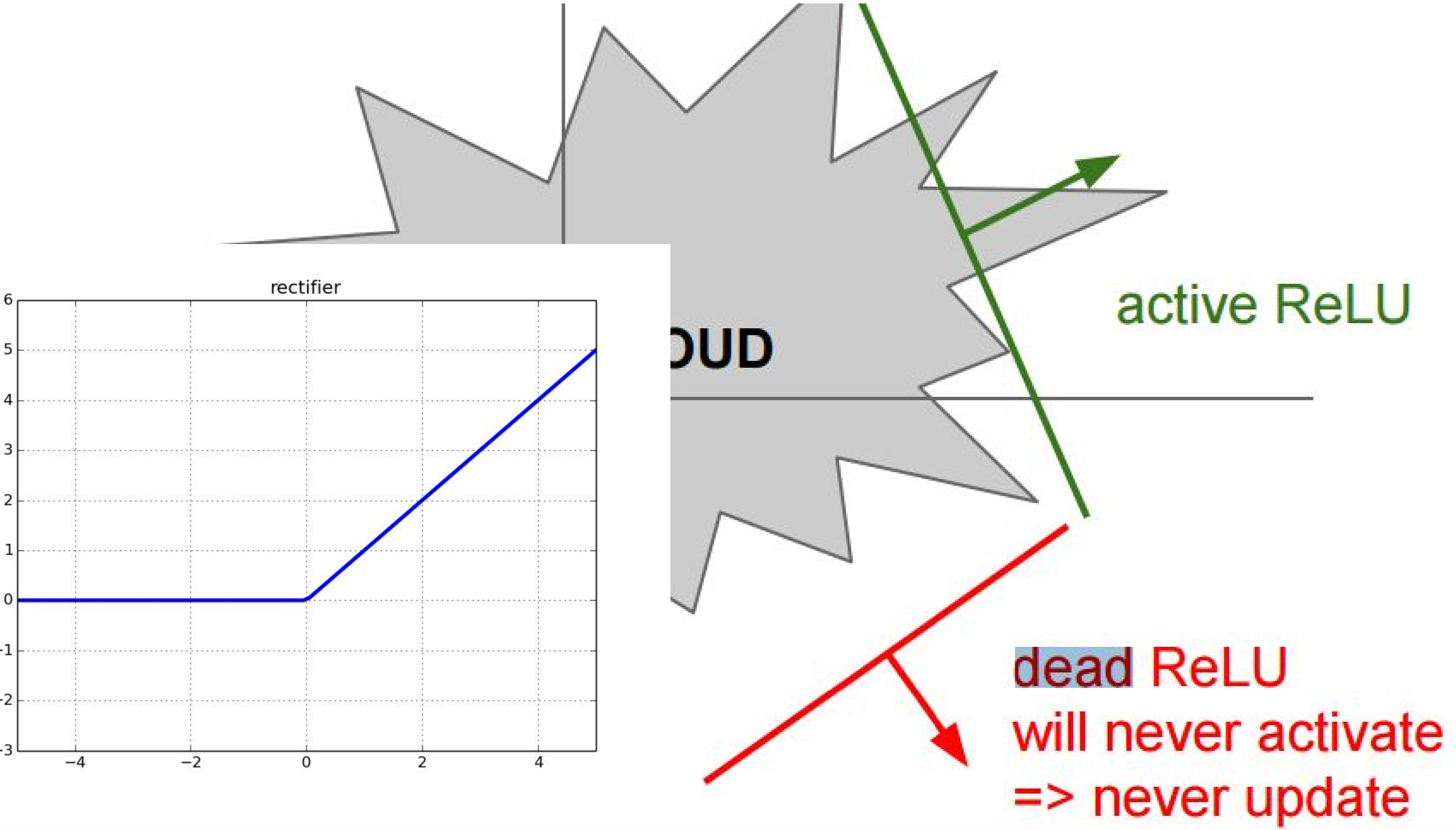
Review: Rectified Linear Units (ReLU)

- ReLU: max(0,x)
- What's the gradient in negative region?

$$RELU(x) = \begin{cases} 0 \ if \ x < 0 \\ x \ if \ x >= 0 \end{cases}$$

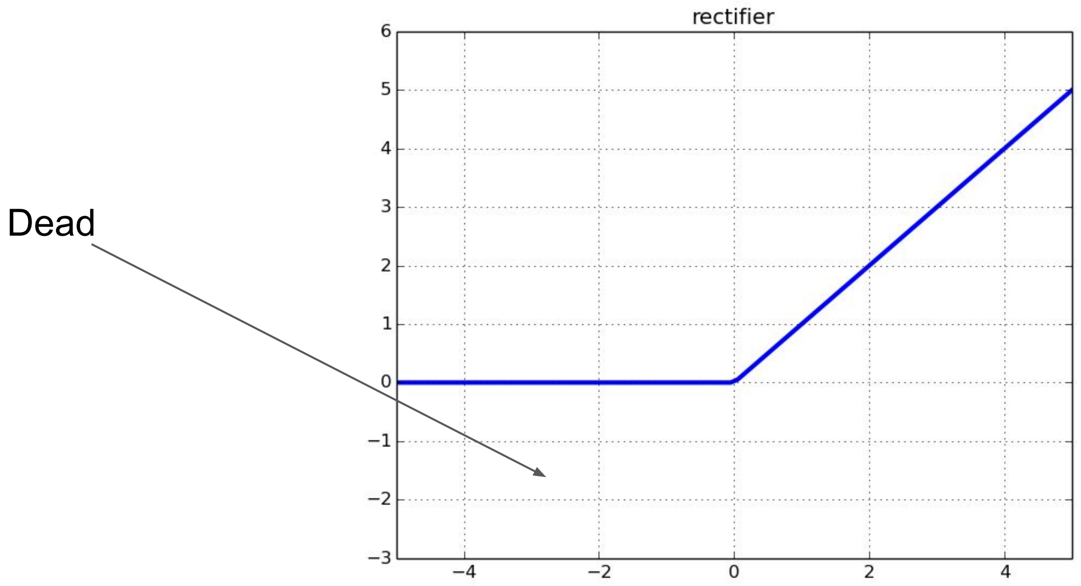
Is there a problem?




rectifier
OUD
active ReLU
dead ReLU
will never activate
=> never update

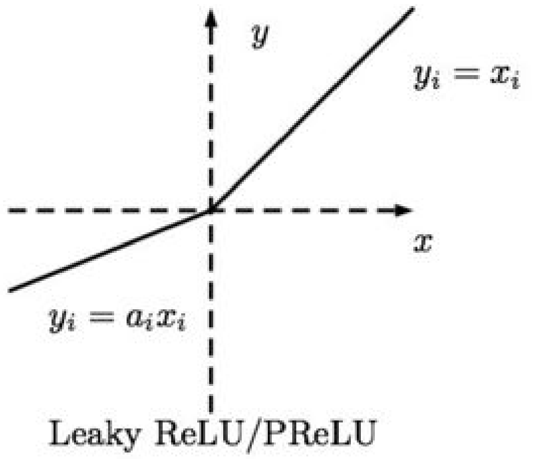
Dying ReLU Problem

- If input to ReLU is negative for the dataset, ReLU dies
- Brief burst of research into addressing dying ReLUs
- General idea is to have **non-zero gradients** even for **negative inputs**



Leaky ReLU & Parameterized ReLU

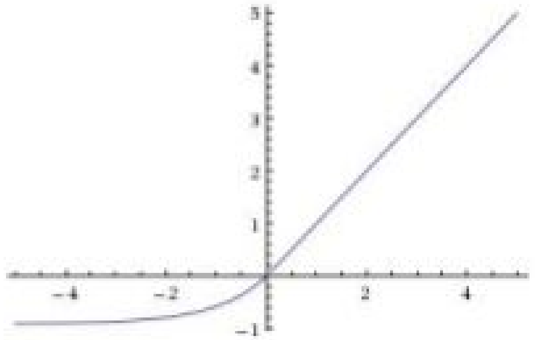
- In Leaky ReLU, **a** is a hyperparameter.
- In Parameterized ReLU, **a** is learned.



Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter Fast and Accurate Deep Network Learning by Exponential Linear Units

Exponential Linear Units (ELUs)

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\,(\exp(x) - 1) & \text{if } x \le 0 \end{cases} \quad , \quad f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ f(x) + \alpha & \text{if } x \le 0 \end{cases}$$

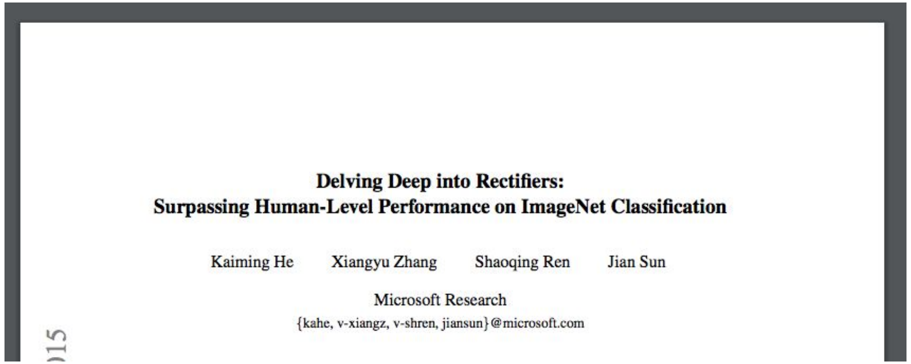


Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter Fast and Accurate Deep Network Learning by Exponential Linear Units

A tale of two papers...

- Top right: paper that introduced PReLUs
- Bottom right: paper that introduced Residual Networks
- What do you notice about these papers?

Screenshots of both papers were taken from arXiv

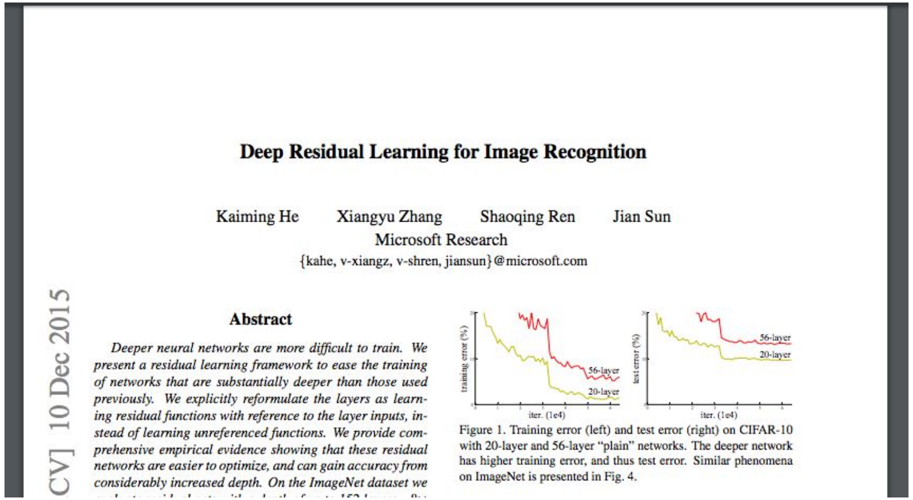


**Delving Deep into Rectifiers:
Surpassing Human-Level Performance on ImageNet Classification**

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research

{kahe, v-xiangz, v-shren, jiansun}@microsoft.com



CV] 10 Dec 2015

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research

{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

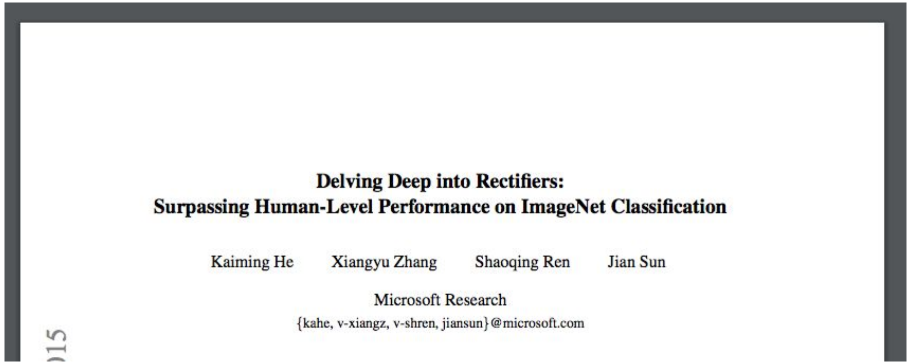
Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Contextualizing

- **Same team that introduced PReLU created ResNet**
- Went back to ReLUs for ResNet
- Focus shifted from activations to overall network design



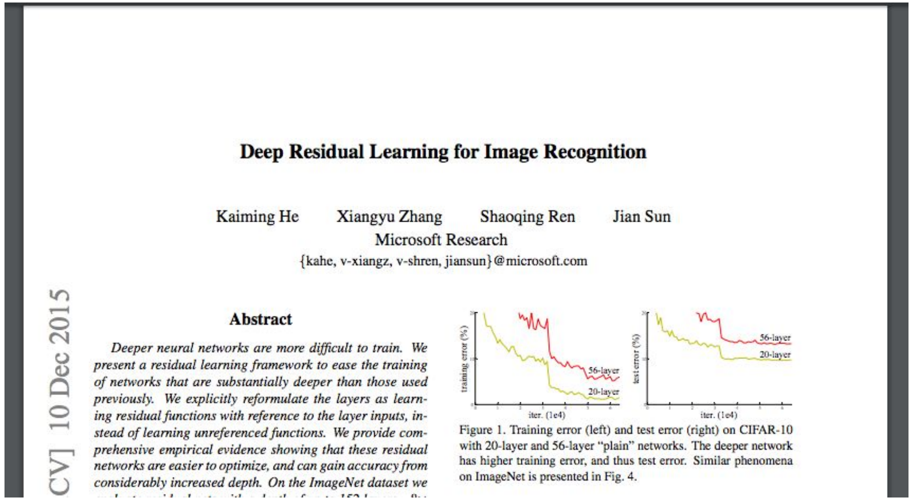
Delving Deep into Rectifiers:
Surpassing Human-Level Performance on ImageNet Classification

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research

{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

015



Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research

{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

CV] 10 Dec 2015

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Screenshots of both papers were taken from arXiv

Contextualizing

- Same team that introduced PReLU created ResNet
- **Went back to ReLUs for ResNet**
- Focus shifted from activations to overall network design

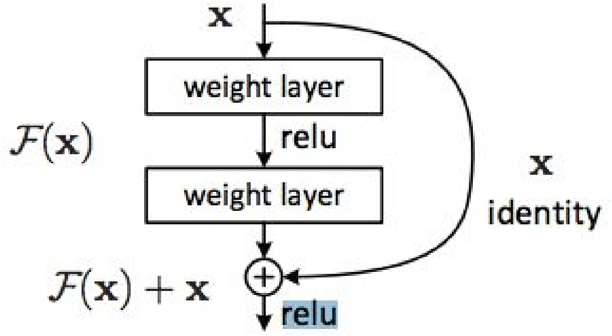

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2. Residual learning: a building block.

Contextualizing

- Same team that introduced PReLU created ResNet
- Went back to ReLUs for ResNet
- **Focus shifted from activations to overall network design**

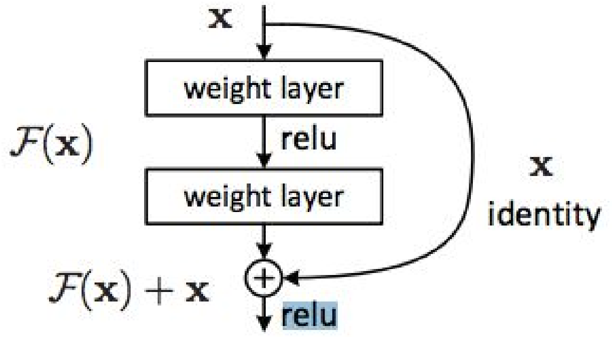
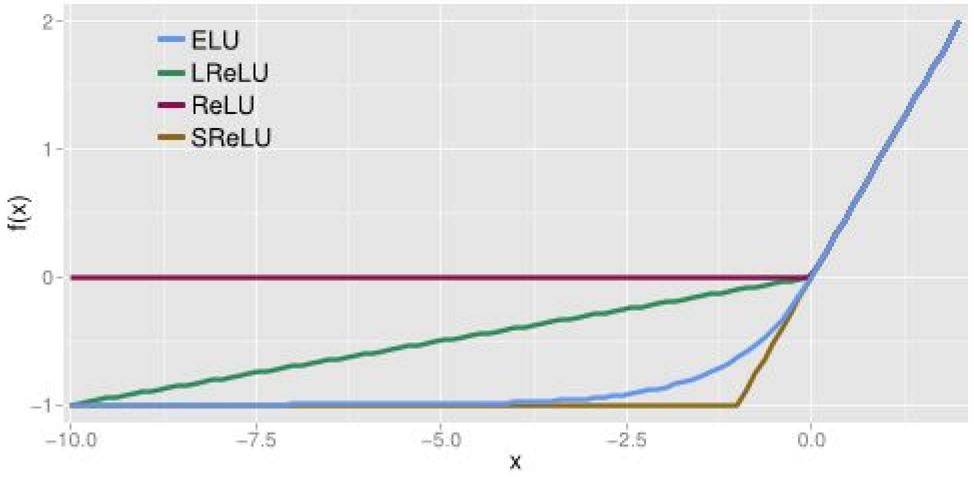


Figure 2. Residual learning: a building block.

In conclusion

- The papers introducing each alternative activations claim they work well
- ReLU still most popular
- All the architectures we are about to discuss used ReLUs (and batch norm)

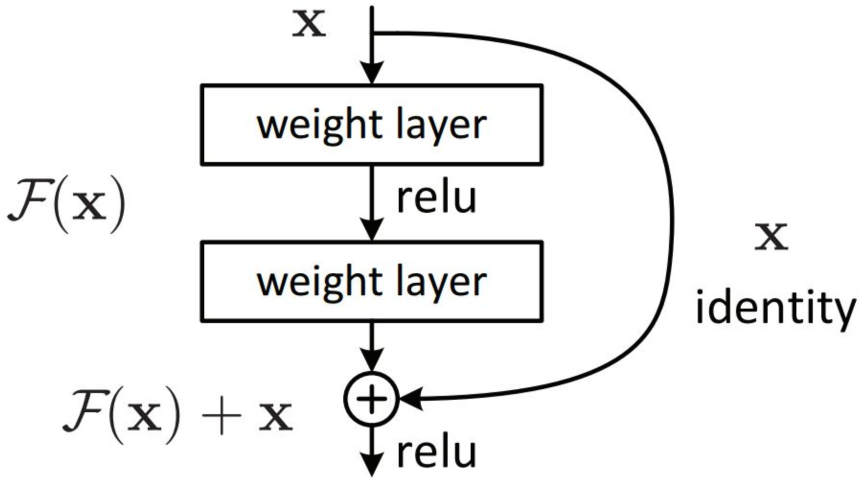


Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter Fast and Accurate Deep Network Learning by Exponential Linear Units

Stage 2:
“Understanding” ResNets

Review

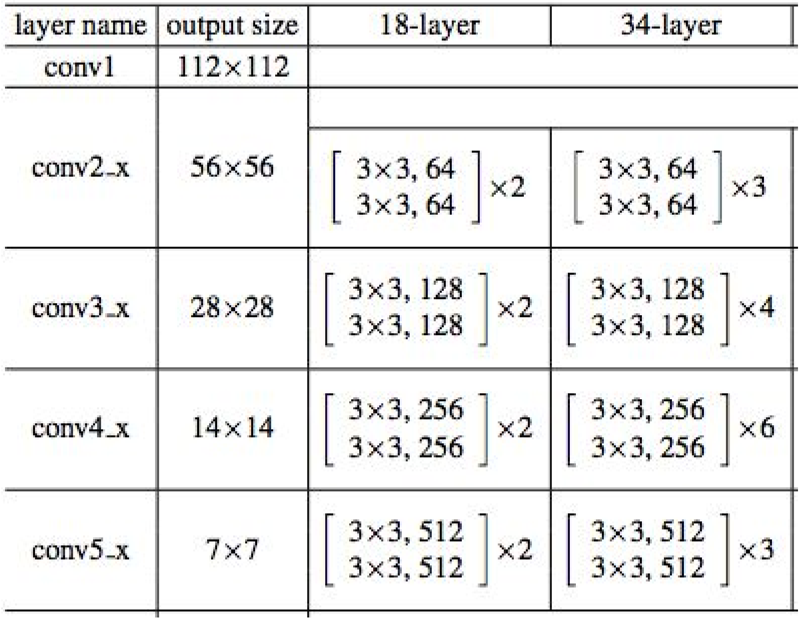
- What is going on inside a Residual Block? (shown to the right)
- Why are there two weight layers?
- What advantage do they have over plain networks?



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016

Going deeper without residuals

- Consider two non-residual networks
  - We call the 18 layer variant ‘plain-18’
  - We call the 34 layer variant ‘plain-34’
- The ‘plain-18’ network outperformed \`plain-34\` on the validation set
- Why do you think this was the case?



layer name	output size	18-layer	34-layer
conv1	112×112		
conv2_x	56×56		
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun,
Deep Residual Learning for Image Recognition, CVPR 2016

18 vs 34 layer ‘plain’ network

- **Vanishing gradients weren’t the issue**
- Overfitting wasn’t the issue
- Representation power wasn’t the issue

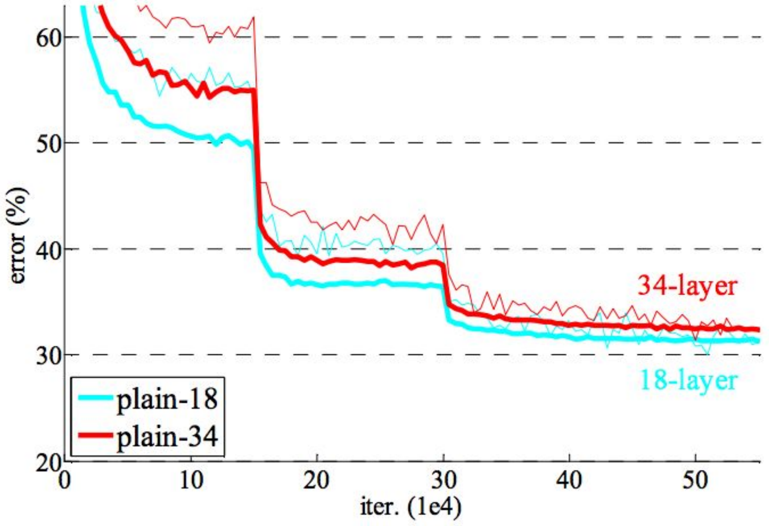
Quote from ResNet paper:

We argue that this optimization difficulty is unlikely to be caused by vanishing gradients. These plain networks are trained with BN, which ensures forward propagated signals to have non-zero variances. We also verify that the backward propagated gradients exhibit healthy norms with BN. So neither forward nor backward signals vanish.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun,
Deep Residual Learning for Image Recognition, CVPR 2016

18 vs 34 layer ‘plain’ network

- Vanishing gradients weren’t the issue
- **Overfitting wasn’t the issue**
- Representation power wasn’t the issue

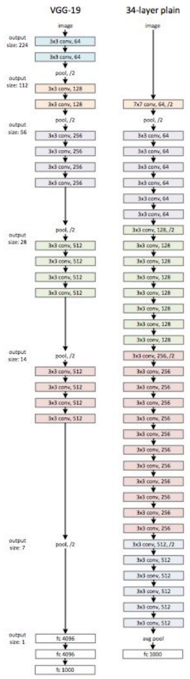


Even the *training* error is higher with the 34 layer network

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016

18 vs 34 layer ‘plain’ network

- Vanishing gradients weren’t the issue
- Overfitting wasn’t the issue
- **Representation power wasn’t the issue**



- The 34 network has more representative power than the 18 layer network
- We can choose padding and a specific convolutional filter to “embed” shallower networks
- With “SAME” padding, what 3x3 convolutional kernel can produce the identity?

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016

18 vs 34 layer ‘plain’ network

- Vanishing gradient wasn’t the issue
- Overfitting wasn’t the issue
- **Representation power wasn’t the issue**

?	?	?
?	?	?
?	?	?

What should the weights be?

- The 34 network has more representative power than the 18 layer network
- We can choose padding and a specific convolutional filter to “embed” shallower networks
- With “SAME” padding, what 3x3 convolutional kernel can produce the identity?

18 vs 34 layer ‘plain’ network

- Vanishing gradient wasn’t the issue
- Overfitting wasn’t the issue
- **Representation power wasn’t the issue**

0	0	0
0	1	0
0	0	0

With ‘SAME’ padding, this will output the same feature map it receives as input

- The 34 network has more representative power than the 18 layer network
- We can choose padding and a specific convolutional filter to “embed” shallower networks
- With “SAME” padding, what 3x3 convolutional kernel can produce the identity?

Optimization issues

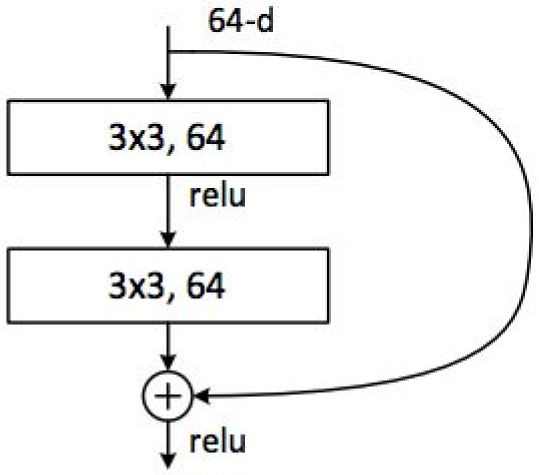
- **Although identity is representable, learning it proves difficult for optimization methods**
- Intution: Tweak the network so it doesn't have to learn identity connections

0	0	0
0	1	0
0	0	0

With 'SAME' padding, this will output the same feature map it receives as input

Optimization issues

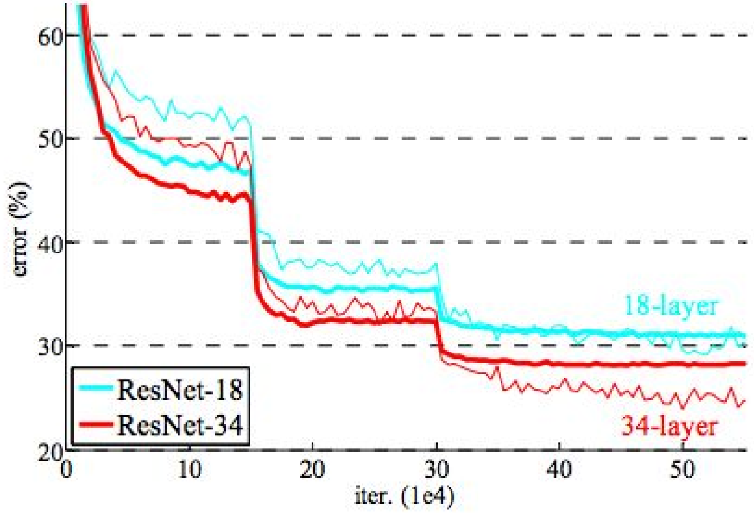
- Although identity is representable, learning it proves difficult for optimization methods
- **Intution: Tweak the network so it doesn't have to learn identity connections**



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016

Optimization issues

- Although identity is representable, learning it proves difficult for optimization methods
- Intution: Tweak the network so it doesn't have to learn identity connections
- **Result: Going deeper makes things better!**



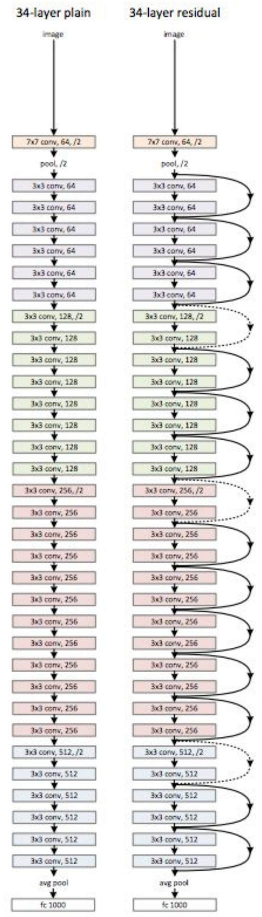
With residuals, the 34-layer network outperforms the 18 layer.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016

Optimization issues

- Although identity is representable, learning it proves difficult for optimization methods
- Intution: Tweak the network so it doesn't have to learn identity connections
- **Result: Going deeper makes things better!**

The architecture of the plain and residual networks were identical except for the skip connections



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016

Interesting Finding

- Less variation in activations for Residual Networks

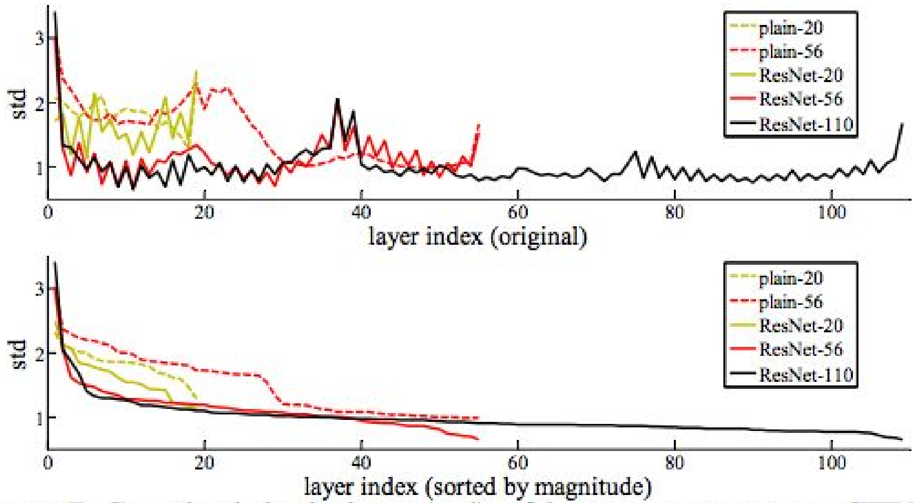
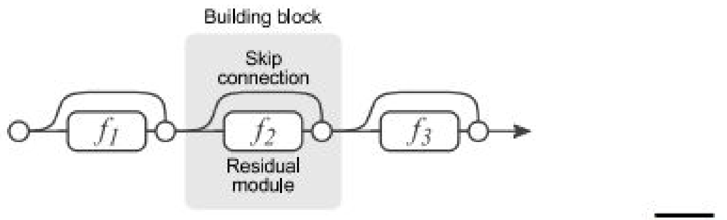


Figure 7. Standard deviations (std) of layer responses on CIFAR-10. The responses are the outputs of each 3×3 layer, after BN and before nonlinearity. **Top**: the layers are shown in their original order. **Bottom**: the responses are ranked in descending order.

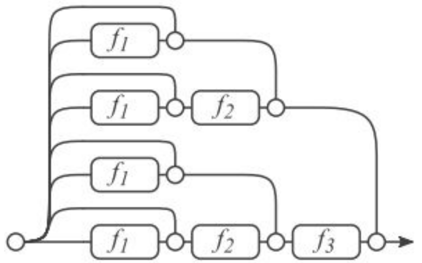
Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun,
Deep Residual Learning for Image Recognition, CVPR 2016

Why do ResNets work? Some ideas:

- **They can be seen as implicitly ensembling shallower networks**
- They are able to learn unrolled iterative refinements
- Can model recurrent computations necessary for recognition



(a) Conventional 3-block residual network

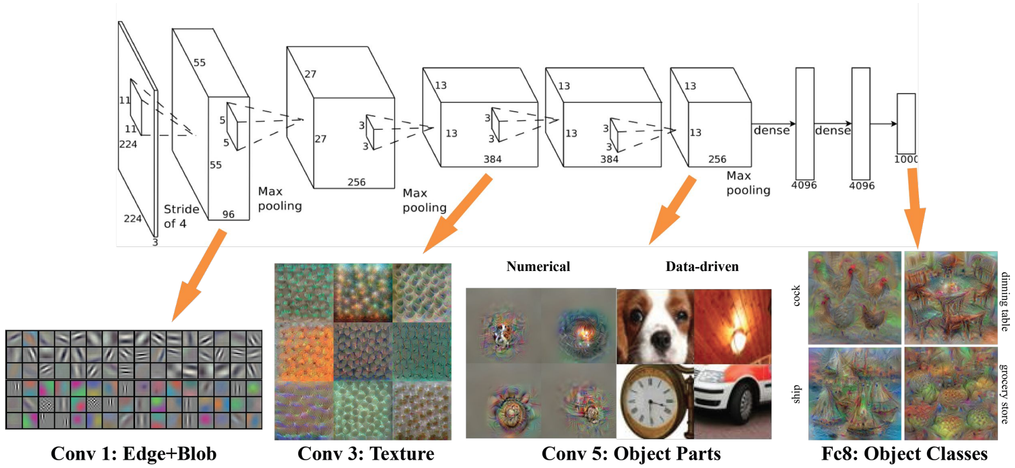


(b) Unraveled view of (a)

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

Why do ResNets work? Some ideas:

- They can be seen as implicitly ensembling shallower networks
- **They are able to learn unrolled iterative estimation**
- Can model recurrent computations necessary for recognition

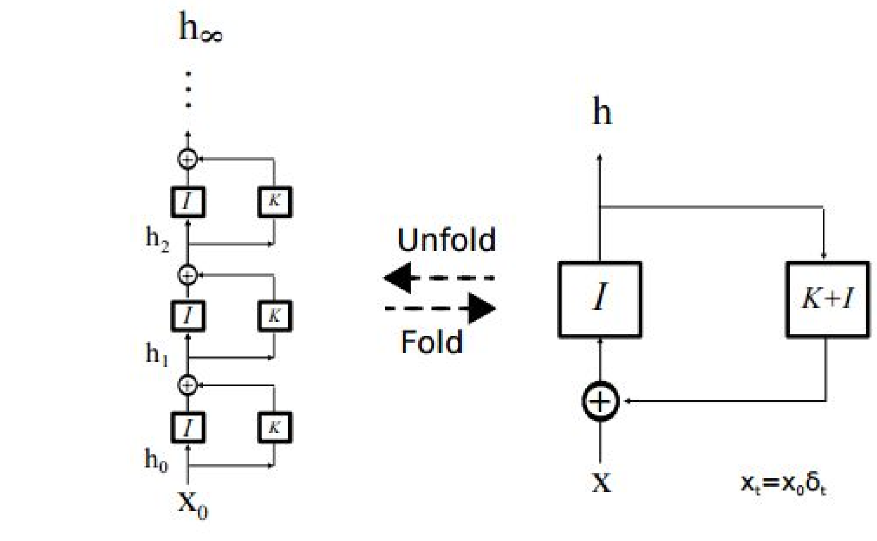


Challenges the "representation view"

Image source: http://vision03.csail.mit.edu/cnn_art/

Why do ResNets work? Some ideas:

- They can be seen as implicitly ensembling shallower networks
- They are able to learn unrolled iterative estimation
- **Can model recurrent computations useful for recognition**



(A) ResNet with shared weights (B) ResNet in recurrent form

Qianli Liao, Tomaso Poggio,Bridging the Gaps Between Residual Learning, Recurrent Neural Networks and Visual Cortex,

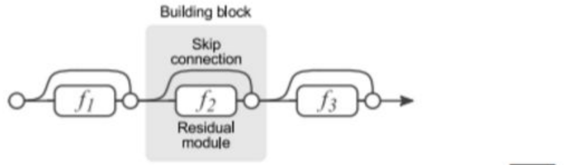
ResNets as Ensembles

- **Can think of ResNets as ensembling subsets of residual modules**
- With L residual modules there are 2^L possible subsets of modules
- If one of modules is removed, there are still 2^{L-1} possible subsets of modules

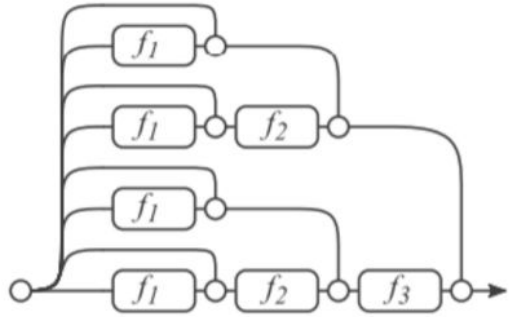
- For each residual module, we can choose whether we include it
- There are 2 options per module (include/exclude) for L modules
- Total of 2^L modules in the implicit ensemble

ResNets as Ensembles

- Can think of ResNets as ensembling subsets of residual modules
- **With L residual modules there are 2^L possible subsets of modules**
- If one of modules is removed, there are still 2^{L-1} possible subsets of modules



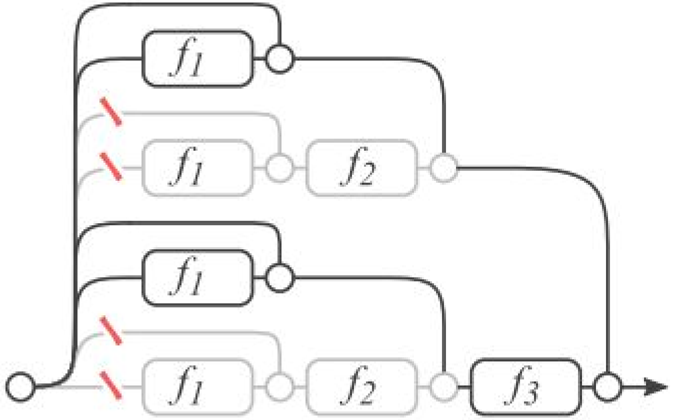
(a) Conventional 3-block residual network



(b) Unraveled view of (a)

ResNets as Ensembles

- Can think of ResNets as ensembling subsets of residual modules
- With L residual modules there are 2^L possible subsets of modules
- **If one of modules is removed, there are still 2^{L-1} possible subsets of modules**

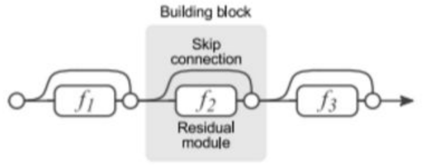


(a) Deleting f_2 from unraveled view

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

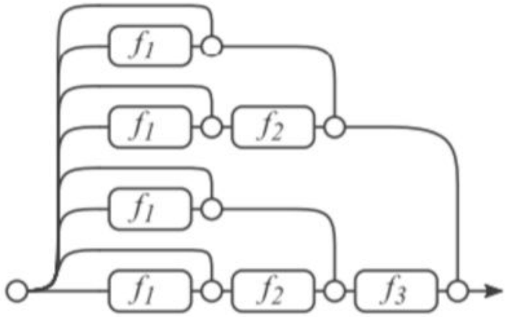
ResNets as Ensembles

- **Wanted to test this explanation**
- Tried dropping layers
- Tried reordering layers
- Found effective paths during training are relatively shallow



(a) Conventional 3-block residual network

=



(b) Unraveled view of (a)

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

ResNets as Ensembles

- Wanted to test this explanation
- **Tried dropping layers**
- Tried reordering layers
- Found effective paths during training are relatively shallow

Dropping layers on VGG-Net is disastorous...

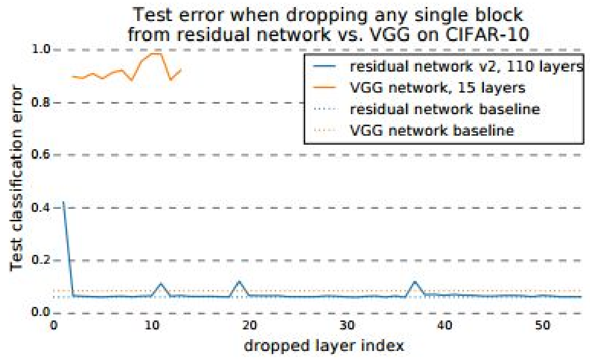


Figure 3: Deleting individual layers from VGG and a residual network on CIFAR-10. VGG performance drops to random chance when any one of its layers is deleted, but deleting individual modules from residual networks has a minimal impact on performance. Removing downsampling modules has a slightly higher impact.

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

ResNets as Ensembles

- Wanted to test this explanation
- **Tried dropping layers**
- Tried reordering layers
- Found effective paths during training are relatively shallow

Dropping layers on ResNet is no big deal

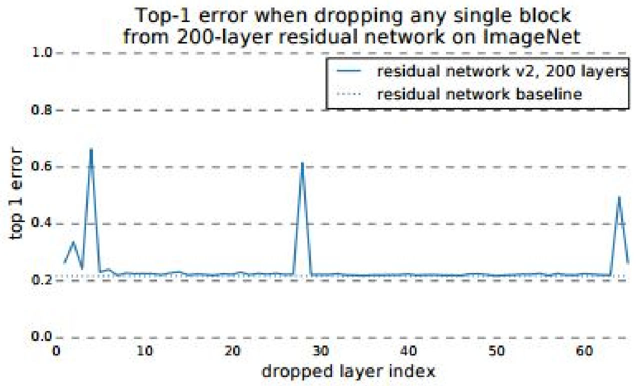
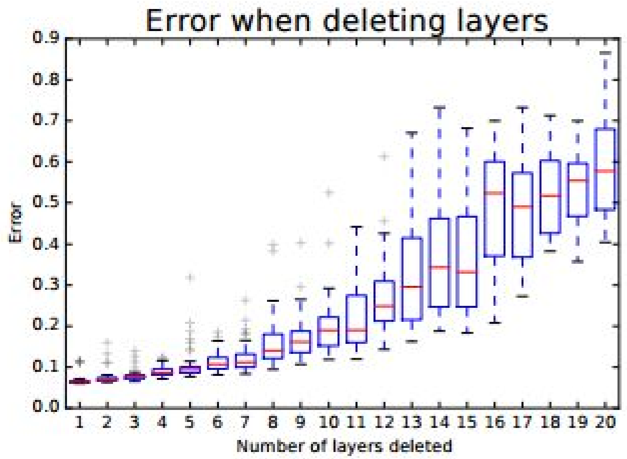


Figure 4: Results when dropping individual blocks from residual networks trained on ImageNet are similar to CIFAR results. However, downsampling layers tend to have more impact on ImageNet.

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

ResNets as Ensembles

- Wanted to test this explanation
- **Tried dropping layers**
- Tried reordering layers
- Found effective paths during training are relatively shallow

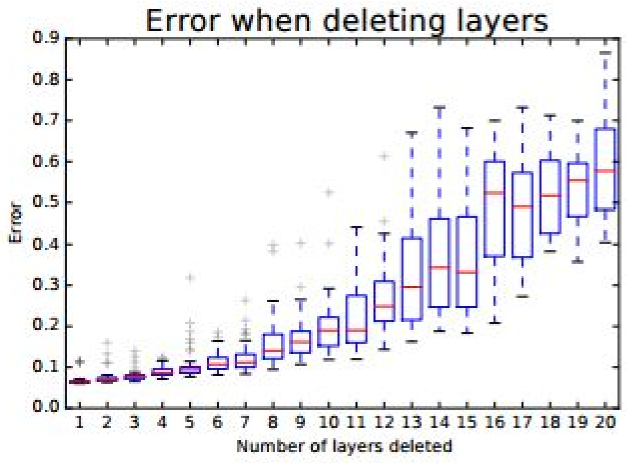


Performance degrades smoothly as layers are removed

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

ResNets as Ensembles

- Wanted to test this explanation
- **Tried dropping layers**
- Tried reordering layers
- Found effective paths during training are relatively shallow

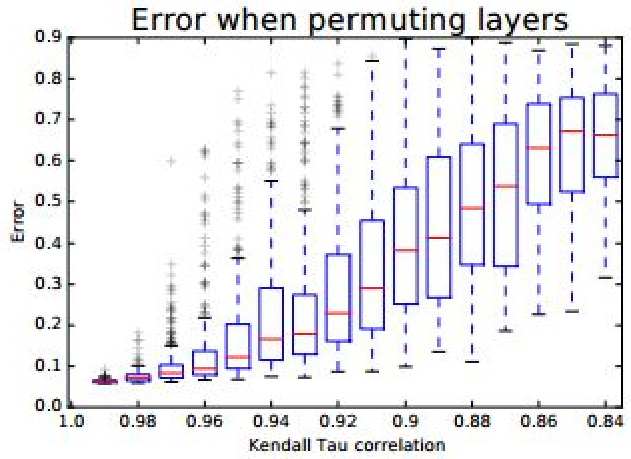


Though the total network has 54 modules; more than 95% of paths go through 19 to 35 modules

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

ResNets as Ensembles

- Wanted to test this explanation
- Tried dropping layers
- **Tried reordering layers**
- Found effective paths during training are relatively shallow



The Kendall Tau correlation coefficient measures the degree of reordering

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

ResNets as Ensembles

- Wanted to test this explanation
- Tried dropping layers
- Tried reordering layers
- **Found effective paths during training are relatively shallow**

[W]e show most gradient during training comes from paths that are even shorter, i.e., 10-34 layers deep.

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

Summary

- ResNets seem to work because they facilitate the training of deeper networks
- Are suprisingly robust to layers being dropped or reordered
- Seem to be function approximations using iterative refinement

Stage 3:
Survey of Architectures

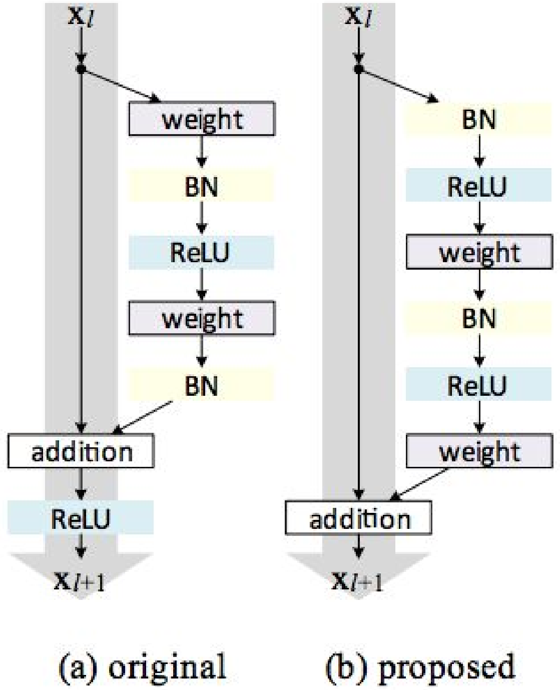
Recap (General Principles in NN Design)

- Reduce filter sizes (except possibly at the lowest layer), factorize filters aggressively
- Use 1x1 convolutions to reduce and expand the number of feature maps judiciously
- Use skip connections and/or create multiple paths through the network

(Professor Lazebnik’s slides)

What are the current trends?

- **Some make minor modifications to ResNets**
- Biggest trend is to split of into several branches, and merge through summation
- A couple architectures go crazy with branch & merge, without explicit identity connections



Identity Mappings in Deep Residual Networks
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

What are the current trends?

- Some make minor modifications to ResNets
- **Biggest trend is to split of into several branches, and merge through summation**
- A couple architectures go crazy with branch & merge, without explicit identity connections

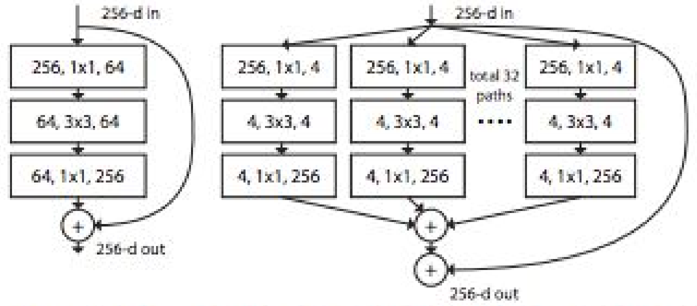
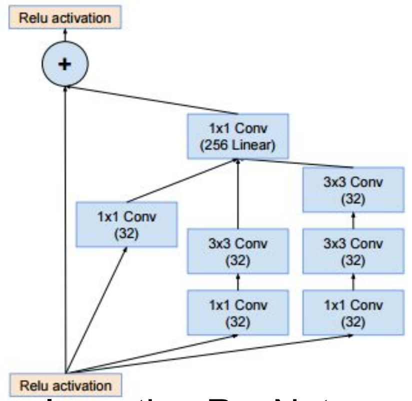


Figure 1. **Left**: A block of ResNet [13]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He.
Aggregated Residual Transformations for Deep Neural Networks

What are the current trends?



Inception ResNet

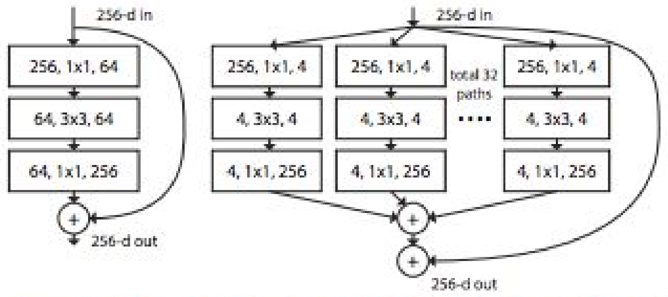


Figure 1. **Left**: A block of ResNet [13]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

ResNeXt

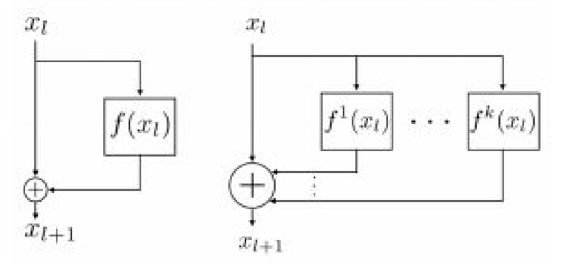
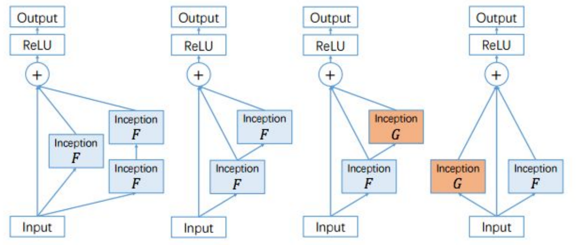


Figure 2: A residual block (left) versus a multi-residual block (right).

MultiResNet



(a) *poly-2* (b) *poly-2* (c) *mpoly-2* (d) *2-way*

Figure 4: Examples of PolyInception structures.

PolyNet

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He. **Aggregated Residual Transformations for Deep Neural Networks**
Masoud Abdi, Saeid Nahavandi. Multi-Residual Networks: **Improving the Speed and Accuracy of Residual Networks**
Xingcheng Zhang, Zhizhong Li, Chen Change Loy, Dahua Lin. **PolyNet: A Pursuit of Structural Diversity in Very Deep Networks**
C. Szegedy et al., **Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning**

What are the current trends?

- Some make minor modifications to ResNets
- Biggest trend is to split of into several branches, and merge through summation
- **A couple architectures go crazy with branch & merge, without explicit identity connections**

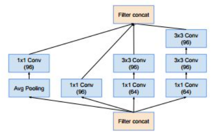


Figure 4. The schema for 35 × 35 grid modules of the pure Inception-v4 network. This is the Inception-A block of Figure 9.

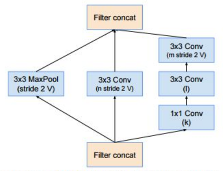


Figure 7. The schema for 35 × 35 to 17 × 17 reduction module. Different variants of this blocks (with various number of filters) are used in Figure 9, and 15 in each of the new Inception(-v4, -ResNet-v1, -ResNet-v2) variants presented in this paper. The k, l, m, n numbers represent filter bank sizes which can be looked up in Table 1.

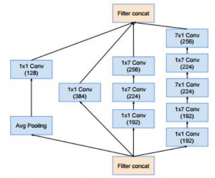


Figure 5. The schema for 17 × 17 grid modules of the pure Inception-v4 network. This is the Inception-B block of Figure 9.

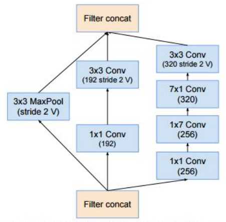


Figure 8. The schema for 17 × 17 to 8 × 8 grid-reduction module. This is the reduction module used by the pure Inception-v4 network in Figure 9.

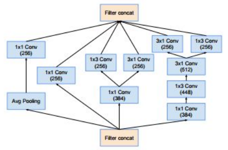
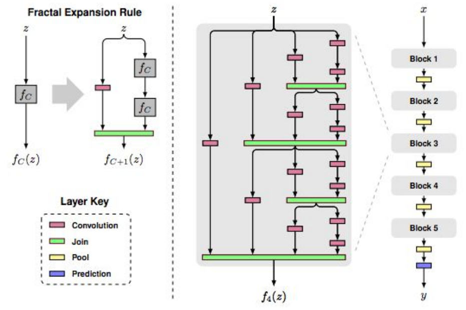


Figure 6. The schema for 8×8 grid modules of the pure Inception-v4 network. This is the Inception-C block of Figure 9.

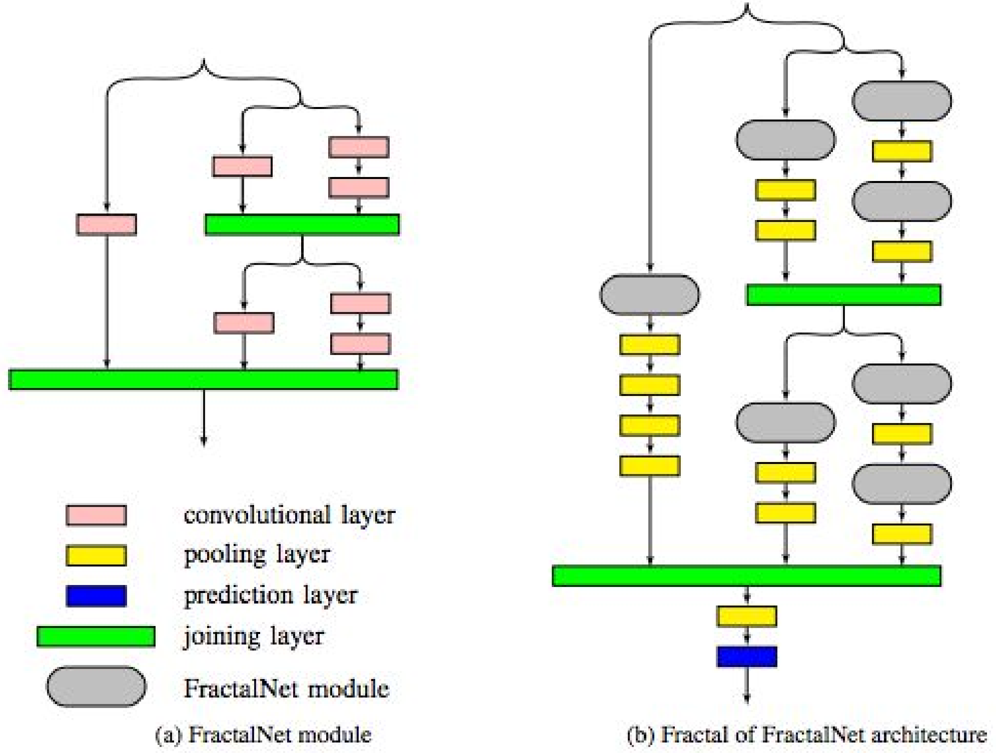


Gustav Larsson, Michael Maire, Gregory Shakhnarovich
FractalNet: Ultra-Deep Neural Networks without Residuals

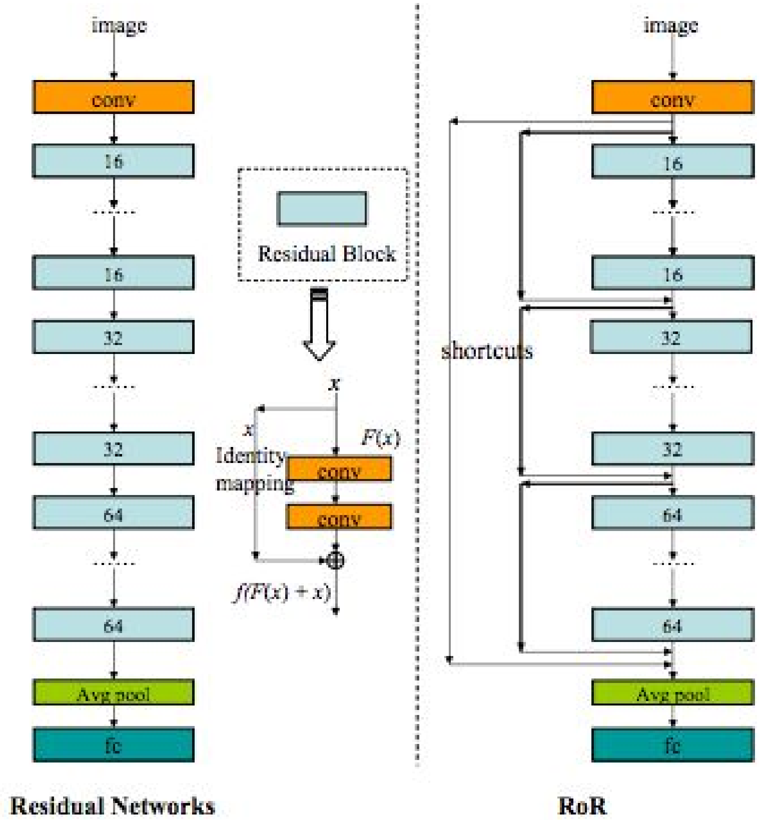
C. Szegedy et al., Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

Some try going meta..

Fractal of Fractals



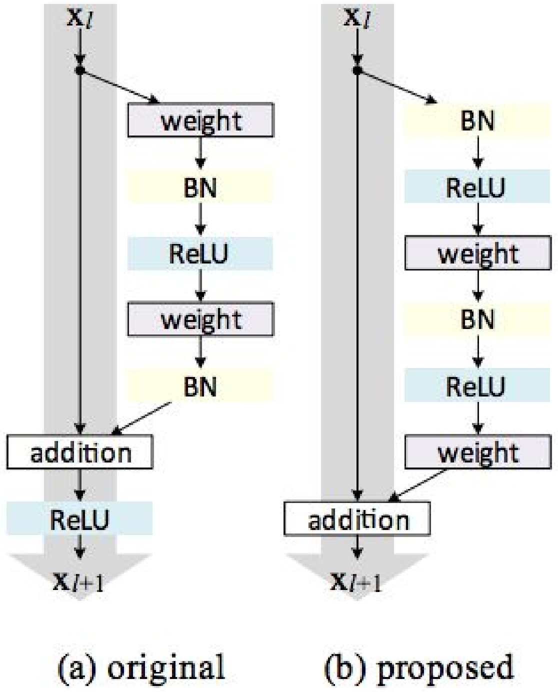
Residuals of Residuals



Leslie N. Smith, Nicholay Topin, Deep Convolutional Neural Network Design Patterns

ResNet tweaks: Change order

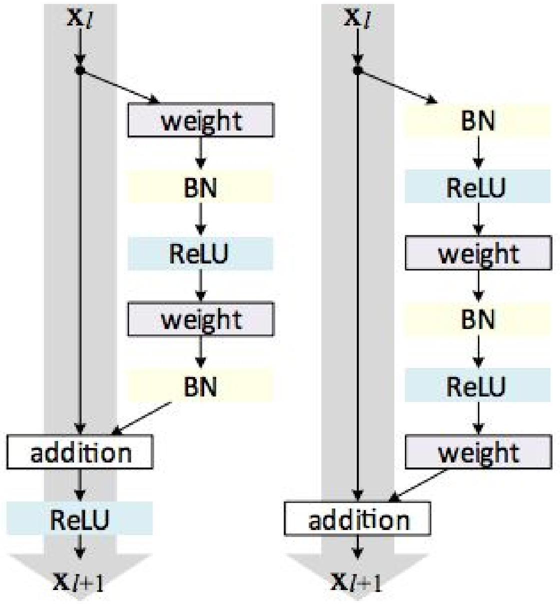
- Pre-activation ResNets
  - **Same components as original, order of BN, ReLU, and conv changed**
  - Idea is to have more direct path for input identity to propagate
  - Resulted in deeper, more accurate networks on ImageNet/CIFAR



Identity Mappings in Deep Residual Networks
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

ResNet tweaks: Change order

- Pre-activation ResNets
    - Same components as original, order of BN, ReLU, and conv changed
    - **Idea is to have more direct path for input identity to propagate**
    - Resulted in deeper, more accurate networks on ImageNet/CIFAR

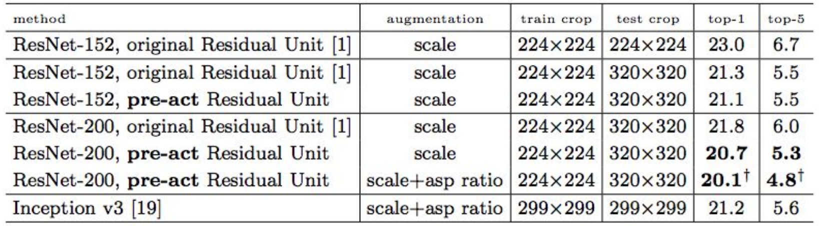


(a) original (b) proposed

Identity Mappings in Deep Residual Networks
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

ResNet tweaks: Change order

- Pre-activation ResNets
  - Same components as original, order of BN, ReLU, and conv changed
  - Idea is to have more direct path for input identity to propagate
  - **Resulted in deeper, more accurate networks on ImageNet/CIFAR**



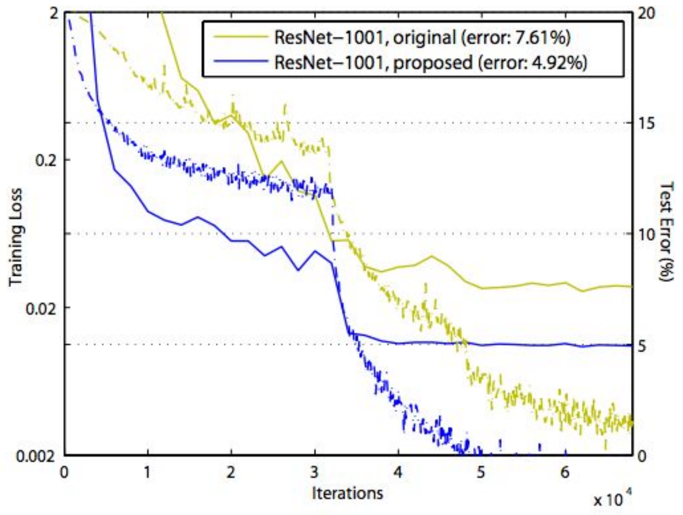
method	augmentation	train crop	test crop	top-1	top-5
ResNet-152, original Residual Unit [1]	scale	224×224	224×224	23.0	6.7
ResNet-152, original Residual Unit [1]	scale	224×224	320×320	21.3	5.5
ResNet-152, **pre-act** Residual Unit	scale	224×224	320×320	21.1	5.5
ResNet-200, original Residual Unit [1]	scale	224×224	320×320	21.8	6.0
ResNet-200, **pre-act** Residual Unit	scale	224×224	320×320	**20.7**	**5.3**
ResNet-200, **pre-act** Residual Unit	scale+asp ratio	224×224	320×320	**20.1**†	**4.8**†
Inception v3 [19]	scale+asp ratio	299×299	299×299	21.2	5.6

ImageNet performance

Identity Mappings in Deep Residual Networks
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

ResNet tweaks: Change order

- Pre-activation ResNets
  - Same components as original, order of BN, ReLU, and conv changed
  - Idea is to have more direct path for input identity to propagate
  - **Resulted in deeper, more accurate networks on ImageNet/CIFAR**

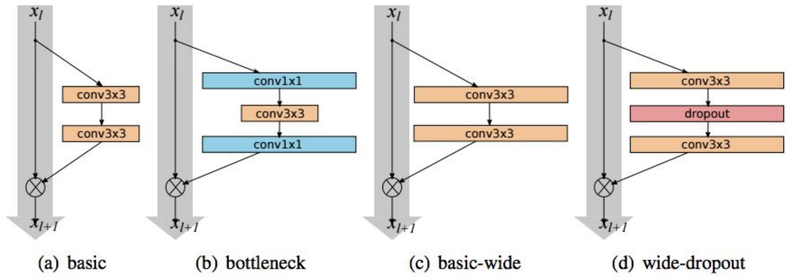


CIFAR-10 performance

Identity Mappings in Deep Residual Networks
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

ResNet tweaks: Wide ResNets

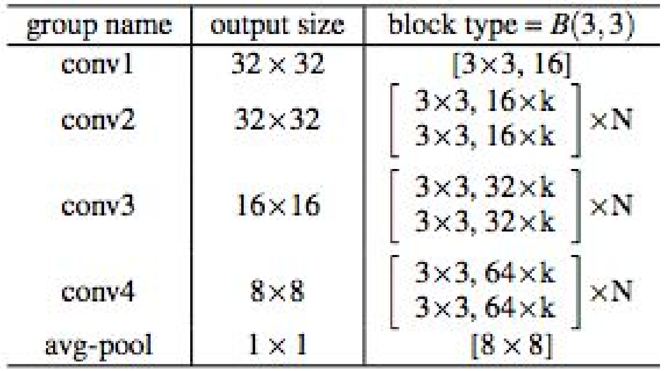
- **Use pre-activation ResNet's basic block with more feature maps**
- Used parameter "k" to encode width
- Investigated relationship between width and depth to find a good tradeoff



Sergey Zagoruyko, Nikos Komodakis
Wide Residual Networks

ResNet tweaks: Wide ResNets

- Use pre-activation ResNet's basic block with more feature maps
- **Used parameter "k" to encode width**
- Investigated relationship between width and depth to find a good tradeoff

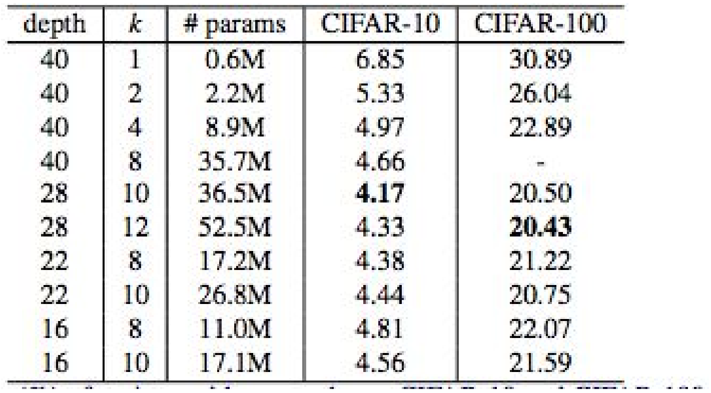


group name	output size	block type = $B(3,3)$
conv1	32×32	$[3\times3, 16]$
conv2	32×32	$\begin{bmatrix} 3\times3, 16\times k \\ 3\times3, 16\times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3\times3, 32\times k \\ 3\times3, 32\times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3\times3, 64\times k \\ 3\times3, 64\times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

Sergey Zagoruyko, Nikos Komodakis
Wide Residual Networks

ResNet tweaks: Wide ResNets

- Use pre-activation ResNet's basic block with more feature maps
- Used parameter "k" to encode width
- **Investigated relationship between width and depth to find a good tradeoff**

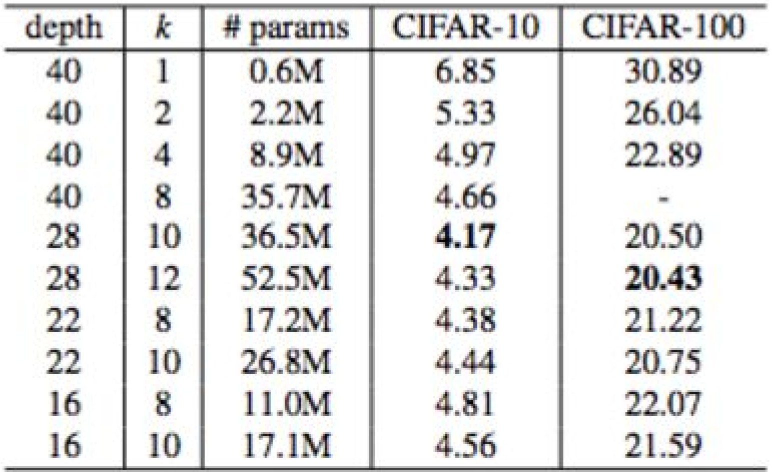


depth	k	# params	CIFAR-10	CIFAR-100
40	1	0.6M	6.85	30.89
40	2	2.2M	5.33	26.04
40	4	8.9M	4.97	22.89
40	8	35.7M	4.66	-
28	10	36.5M	**4.17**	20.50
28	12	52.5M	4.33	**20.43**
22	8	17.2M	4.38	21.22
22	10	26.8M	4.44	20.75
16	8	11.0M	4.81	22.07
16	10	17.1M	4.56	21.59

Sergey Zagoruyko, Nikos Komodakis
Wide Residual Networks

ResNet tweaks: Wide ResNets

- Use pre-activation ResNet's basic block with more feature maps
- Used parameter "k" to encode width
- **Investigated relationship between width and depth to find a good tradeoff**

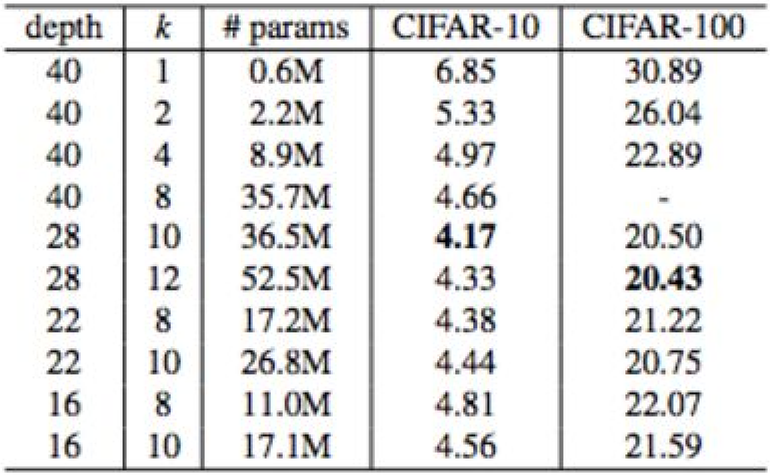


depth	k	# params	CIFAR-10	CIFAR-100
40	1	0.6M	6.85	30.89
40	2	2.2M	5.33	26.04
40	4	8.9M	4.97	22.89
40	8	35.7M	4.66	-
28	10	36.5M	**4.17**	20.50
28	12	52.5M	4.33	**20.43**
22	8	17.2M	4.38	21.22
22	10	26.8M	4.44	20.75
16	8	11.0M	4.81	22.07
16	10	17.1M	4.56	21.59

Sergey Zagoruyko, Nikos Komodakis
Wide Residual Networks

ResNet tweaks: Wide ResNets

- **These obtained state of the art results on CIFAR datasets**
- Were outperformed by bottlenecked networks on ImageNet
- Best results on ImageNet were obtained by widening ResNet-50

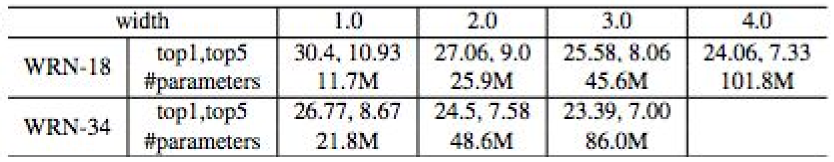


depth	*k*	# params	CIFAR-10	CIFAR-100
40	1	0.6M	6.85	30.89
40	2	2.2M	5.33	26.04
40	4	8.9M	4.97	22.89
40	8	35.7M	4.66	-
28	10	36.5M	**4.17**	20.50
28	12	52.5M	4.33	**20.43**
22	8	17.2M	4.38	21.22
22	10	26.8M	4.44	20.75
16	8	11.0M	4.81	22.07
16	10	17.1M	4.56	21.59

Sergey Zagoruyko, Nikos Komodakis
Wide Residual Networks

ResNet tweaks: Wide ResNets

- These obtained state of the art results on CIFAR datasets
- **Were outperformed by bottlenecked networks on ImageNet**
- Best results on ImageNet were obtained by widening ResNet-50

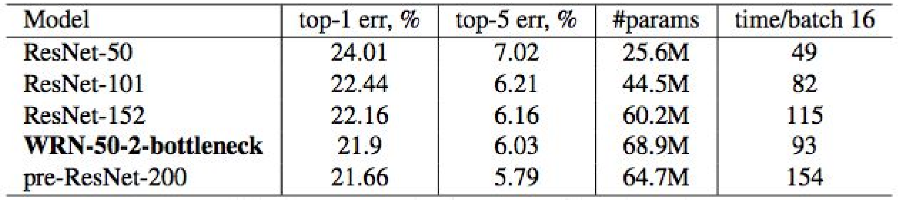


width		1.0	2.0	3.0	4.0
WRN-18	top1,top5	30.4, 10.93	27.06, 9.0	25.58, 8.06	24.06, 7.33
	#parameters	11.7M	25.9M	45.6M	101.8M
WRN-34	top1,top5	26.77, 8.67	24.5, 7.58	23.39, 7.00	
	#parameters	21.8M	48.6M	86.0M	

Sergey Zagoruyko, Nikos Komodakis
Wide Residual Networks

ResNet tweaks: Wide ResNets

- These obtained state of the art results on CIFAR datasets
- Were outperformed by bottlenecked networks on ImageNet
- **Best results on ImageNet were obtained by widening ResNet-50**



Model	top-1 err, %	top-5 err, %	#params	time/batch 16
ResNet-50	24.01	7.02	25.6M	49
ResNet-101	22.44	6.21	44.5M	82
ResNet-152	22.16	6.16	60.2M	115
WRN-50-2-bottleneck	21.9	6.03	68.9M	93
pre-ResNet-200	21.66	5.79	64.7M	154

"With widening factor of 2.0 the resulting WRN-50-2-bottleneck outperforms ResNet-152 having 3 times less layers, and being significantly faster."

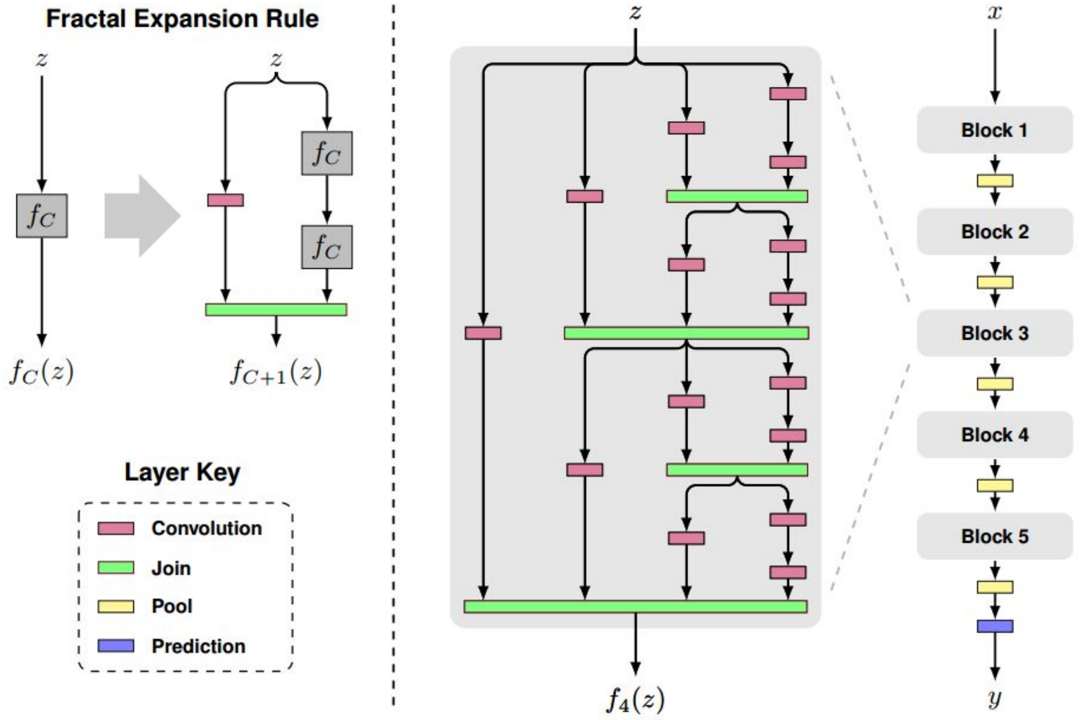
Sergey Zagoruyko, Nikos Komodakis
Wide Residual Networks

Aside from ResNets

FractalNet and DenseNet

FractalNet

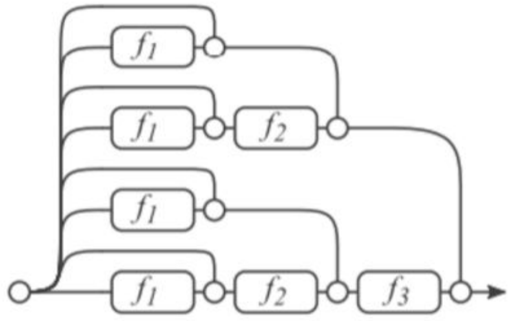
- A competitive extremely deep architecture that does not rely on residuals



Gustav Larsson, Michael Maire, Gregory Shakhnarovich FractalNet: Ultra-Deep Neural Networks without Residuals

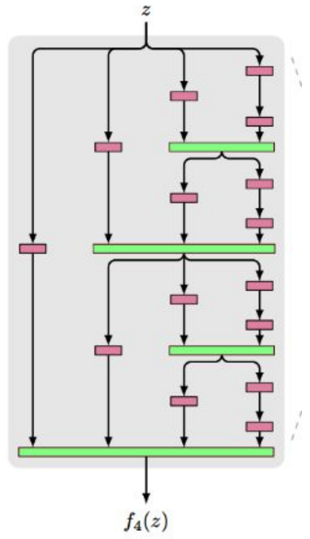
FractalNet

- A competitive extremely deep architecture that does not rely on residuals
- Interestingly, its architecture is similar to an unfolded ResNet



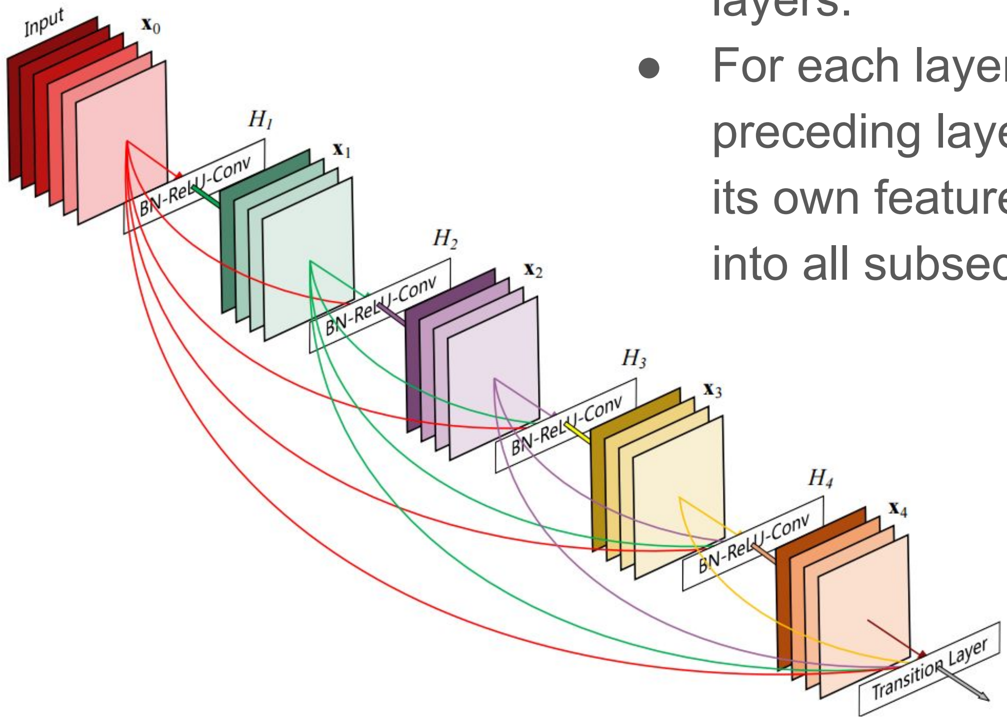
(b) Unraveled view of (a)

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016



Gustav Larsson, Michael Maire, Gregory Shakhnarovich FractalNet: Ultra-Deep Neural Networks without Residuals

DenseNet (Within a DenseBlock)



- Every layer is connected to all other layers.
- For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers.

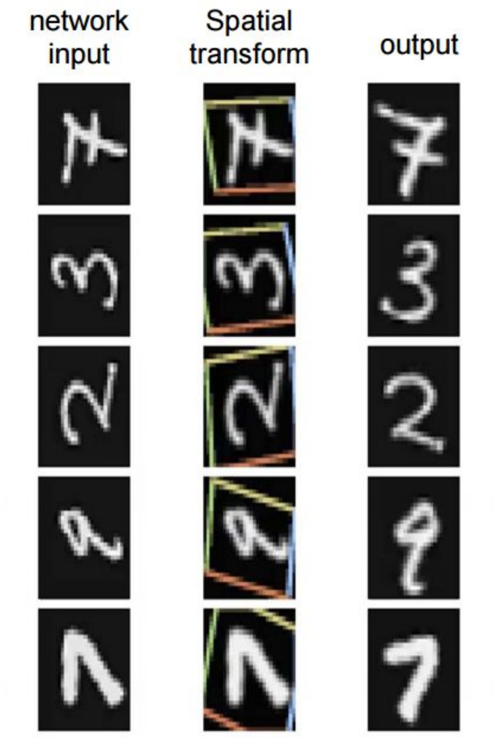
DenseNet

- Alleviate the vanishing-gradient problem.
- Strengthen feature propagation.
- Encourage feature reuse.
- **Substantially** reduce the number of parameters.

Bonus Material!

We’ll cover spatial transformer networks (briefly)

Spatial Transformer Networks



A module to provide spatial transformation capabilities on individual data samples.

Idea:

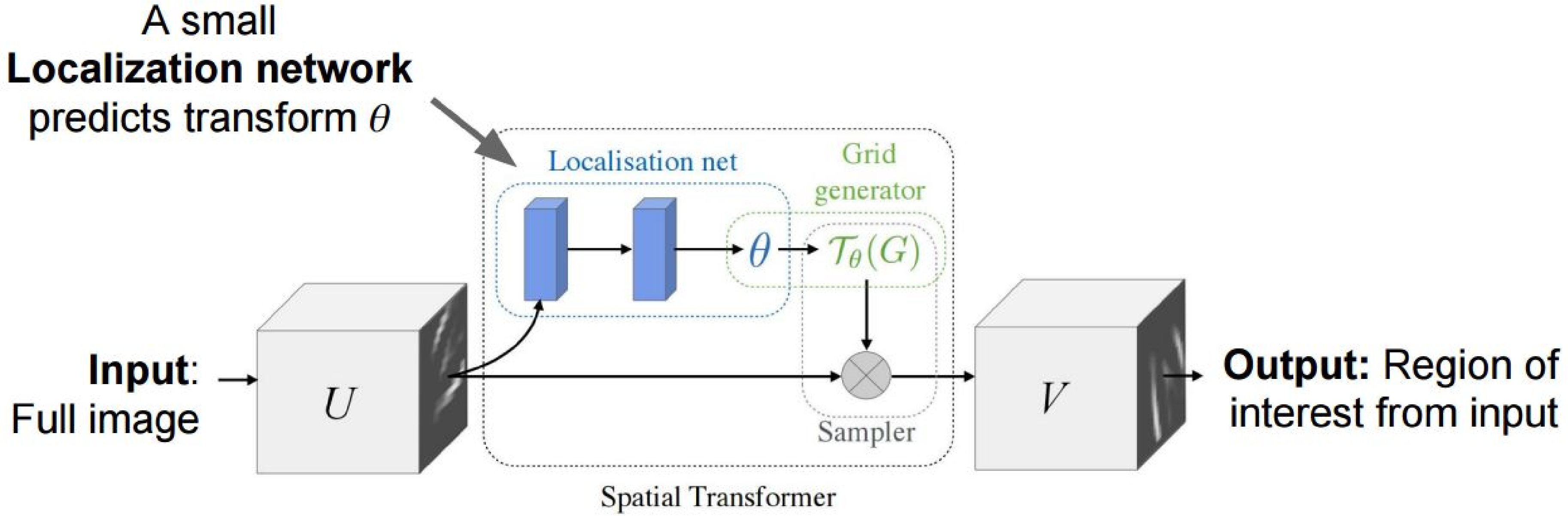
Function mapping pixel coordinates of output to pixel coordinates of input.

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

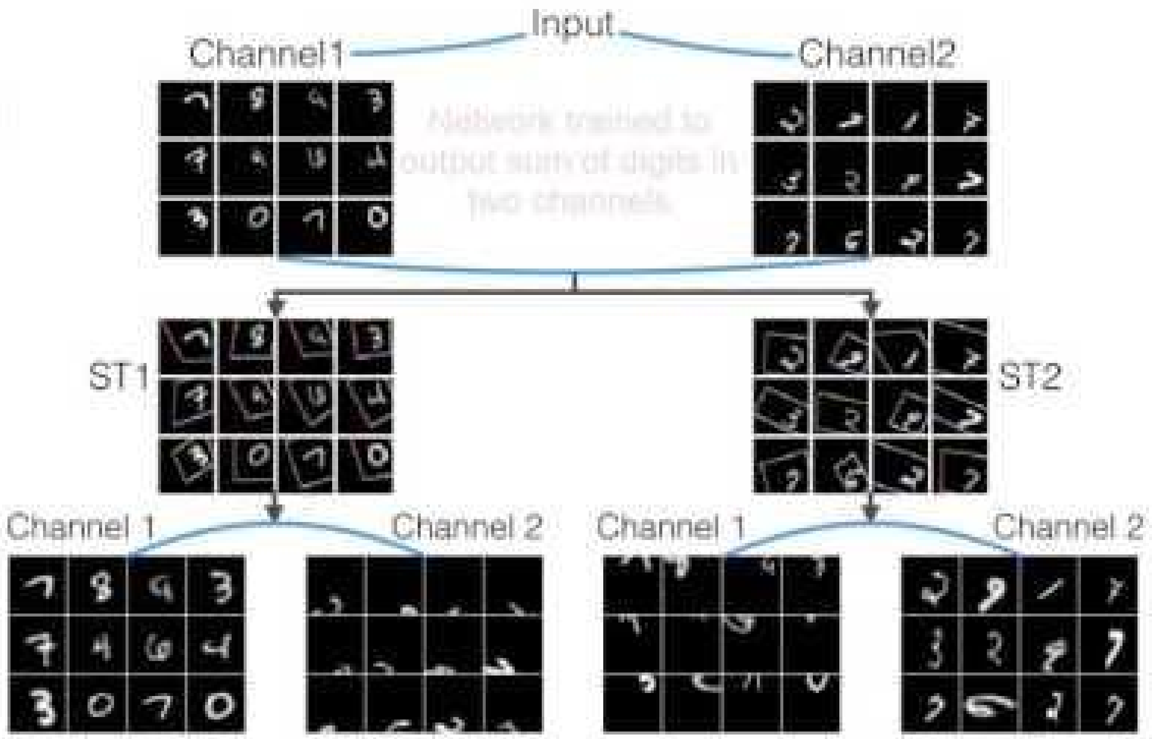
Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu
Spatial Transformer Networks

Spatial transform by how much?

The localisation network function can take any form, such as a fully-connected network or a convolutional network, but should include a final regression layer to produce the transformation parameters θ.



MNIST Addition



Concluding Remarks

- At surface level, there’s tons of new architectures that are very different
- Upon closer inspection, most of them are reapplying well established principles
- Universal principles seem to be having shorter subpaths through the networks
- Identity propagation (Residuals, Dense Blocks) seem to make training easier

References

- Sergey Ioffe,Christian Szegedy,Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, http://jmlr.org/proceedings/papers/v37/ioffe15.pdf
- K. He, X. Zhang, S. Ren, and J. Sun, Deep Residual Learning for Image Recognition, CVPR 2016 https://arxiv.org/abs/1512.03385
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Identity Mappings in Deep Residual Networks https://arxiv.org/abs/1603.05027
- Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, Aggregated Residual Transformations for Deep Neural Networks, https://arxiv.org/pdf/1611.05431v1.pdf
- Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu: Spatial Transformer Networks https://arxiv.org/abs/1506.02025
- Leslie N. Smith, Nicholay Topin, Deep Convolutional Neural Network Design Patterns, https://arxiv.org/abs/1611.00847
- Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning https://arxiv.org/abs/1602.07261
- Gustav Larsson, Michael Maire, Gregory Shakhnarovich, FractalNet: Ultra-Deep Neural Networks without Residuals https://arxiv.org/abs/1605.07648
- Gao Huang, Zhuang Liu, Kilian Q. Weinberger, Laurens van der Maaten: Densely Connected Convolutional Networks https://arxiv.org/pdf/1608.06993v3.pdf
- Rupesh Kumar Srivastava, Klaus Greff, Jürgen Schmidhuber: Highway Networks https://arxiv.org/abs/1505.00387
- Xingcheng Zhang, Zhizhong Li, Chen Change Loy, Dahua Lin, PolyNet: A Pursuit of Structural Diversity in Very Deep Networks, https://arxiv.org/abs/1611.05725
- Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. https://arxiv.org/abs/1412.6806
- Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, https://arxiv.org/pdf/1605.06431v2.pdf
- Klaus Greff, Rupesh K. Srivastava & Jürgen Schmidhuber, Highway and Residual Networks Learn Unrolled Iterative Estimation,https://arxiv.org/pdf/1612.07771v1.pdf
- Min Lin, Qiang Chen, Shuicheng Yan, Network In Network, https://arxiv.org/abs/1312.4400
- Brian Chu, Daylen Yang, Ravi Tadinada, Visualizing Residual Networks, https://arxiv.org/abs/1701.02362
- Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter: Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) https://arxiv.org/abs/1511.07289
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, https://arxiv.org/abs/1502.01852
- Anish Shah, Eashan Kadam, Hena Shah, Sameer Shinde, Sandip Shingade, Deep Residual Networks with Exponential Linear Unit, https://arxiv.org/abs/1604.04112