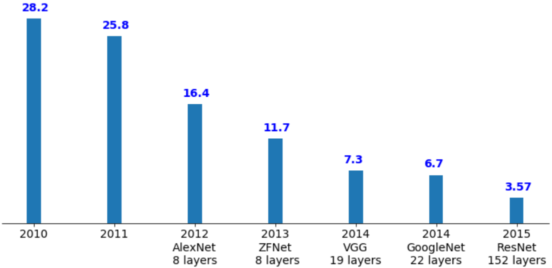
Residual Learning

Nan Ye

School of Mathematics and Physics
The University of Queensland

Recall: ILSVRC

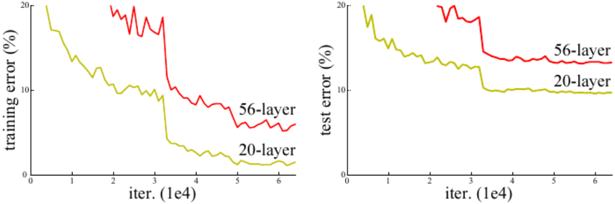
- ILSVRC (ImageNet Large Scale Visual Recognition Challenge) was an competition based on the ImageNet data.
- Top-5 classification error rates for the best systems



- ResNet was the best ImageNet object recognizer in 2015.

Deeper ⇒ Better Fit?

- Theory: a deeper network has a larger capacity ⇒ better fit on the training set.
- Practice: a very deep network may not even fit better than a shallower network on the training set



He, Zhang, Ren, and Sun, Deep residual learning for image recognition, 2016

- Optimization for deeper networks are hard
  - the exploding/vanishing gradient problem (as seen in RNN lectures)

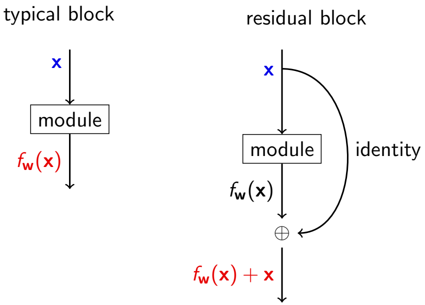
Recall: avoiding exploding/vanishing gradients for RNNs

- Truncated BPTT (backprop through time)
- Good initialization
- Gradient clipping
- Design a better architecture

ResNet: design an architecture that learns residuals

ResNet

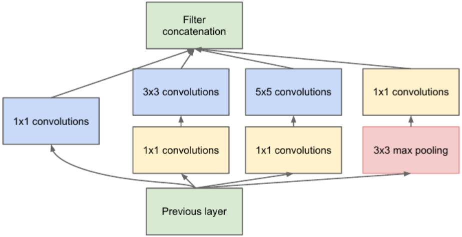
- The key idea of ResNet is to replace typical computational blocks by residual blocks.
- Each residual block computes a function $f_{\mathbf{w}}(\mathbf{x}) + \mathbf{x}$ (residual + identity), with identity implemented as a shortcut connection.



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

Shortcuts

- Shortcut connections were used in previous architectures such as the inception module in GoogLeNet



- ResNet's shortcut connection does not introduce new parameters nor computational complexity.

Why residual blocks

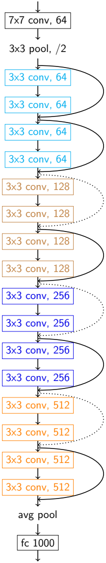
- Near-identity blocks can be used to represent very complex mappings between inputs and outputs
  - *a lot of empirical evidence and some theoretical justifications*
- Possibly, they are a *natural* building block for representing input-output mappings.
- Learning near-identity functions using standard neural net blocks is hard, but learning the residuals of such near-identity functions using standard neural net blocks turns out to be much easier.
  - *why? if there is a 20-layer solution, there is a 56-layer solution with additional identity layers, but learning such a network is harder*.

still an active area of research

Keeping the dimensions unchanged

- For fully connected layers, simply keep the output dimension the same as the input dimension.
- For convolutional layers, two options are often used
  - use 0 padding in convolution to keep the feature map size the same, and keep number of channels the same,
  - reduce feature map size and increase number of filters, implement the skip connection as a downsampling layer
- In general, the identity connection can be replaced by a linear projection, and a residual block computes $f_{\mathbf{w}}(\mathbf{x}) + P\mathbf{x}$ (in practice, P can be user-specified or learned).

Example: An 18-layer ResNet

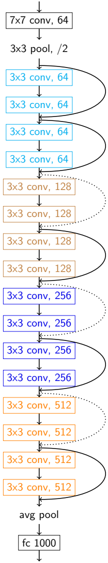


S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3)	64x112x112
MaxPool(3x3, S=2, P=1)	64x56x56
2 residual blocks, each has 2 Conv(3x3, 64, S=1, P=1) layers	64x56x56
2 residual blocks, each has 2 Conv(3x3, 128) layers	128x28x28
2 residual blocks, each has 2 Conv(3x3, 256) layers	256x14x14
2 residual blocks, each has 2 Conv(3x3, 512) layers	512x7x7
Global average pooling	512x1x1
Fully connected layer with 1000 outputs	1000

- While there are 20 layers in the table, the two pooling layers have no parameters are not counted.
- For the 2nd to the 4th residual blocks, all the convolutional filters have padding 1, but the strides are different.
- We walk though the architecture in next few slides.

Example: An 18-layer ResNet

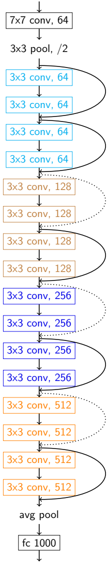


S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3) $\lfloor(224 + 2 * 3 - 7 + 2)/2\rfloor = 112$	64x112x112

Recall: $\lfloor \frac{N+2P-D(F-1)-1}{S} \rfloor + 1$, equal to $\lfloor \frac{N+2P-F+S}{S} \rfloor$ when $D = 1$.

Example: An 18-layer ResNet

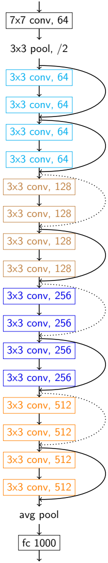


S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3)	64x112x112
MaxPool(3x3, S=2, P=1) $\lfloor (112 + 2 * 1 - 3 + 2)/2 \rfloor = 56$	64x56x56

Recall: $\lfloor \frac{N+2P-D(F-1)-1}{S} \rfloor + 1$, equal to $\lfloor \frac{N+2P-F+S}{S} \rfloor$ when $D = 1$.

Example: An 18-layer ResNet

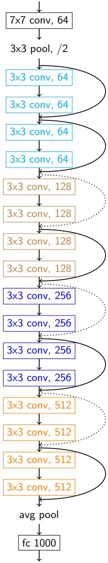


S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3)	64x112x112
MaxPool(3x3, S=2, P=1)	64x56x56
2 residual blocks, each has 2 Conv(3x3, 64, S=1, P=1) layers (56 + 2*1 - 3 + 1)/1 = 56 for both layers	64x56x56

Recall: $\lfloor \frac{N+2P-D(F-1)-1}{S} \rfloor + 1$, equal to $\lfloor \frac{N+2P-F+S}{S} \rfloor$ when $D = 1$.

Example: An 18-layer ResNet



S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3)	64x112x112
MaxPool(3x3, S=2, P=1)	64x56x56
2 residual blocks, each has 2 Conv(3x3, 64, S=1, P=1) layers	64x56x56
2 residual blocks, each has 2 Conv(3x3, 128) layers	128x28x28

$\lfloor (56 + 2 * 1 - 3 + 2)/2 \rfloor = 28$ for 1st layer in 1st block

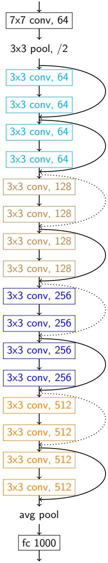
(28 + 2*1 - 3 + 1)/1 = 28 for other layers

skip connection for 1st block is not identity, but Conv(2x2,128, S=2, P=0)

why? to downsample input to match output size

Recall: $\lfloor \frac{N+2P-D(F-1)-1}{S} \rfloor + 1$, equal to $\lfloor \frac{N+2P-F+S}{S} \rfloor$ when $D = 1$.

Example: An 18-layer ResNet

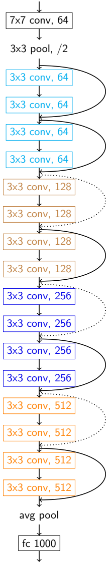


S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3)	64x112x112
MaxPool(3x3, S=2, P=1)	64x56x56
2 residual blocks, each has 2 Conv(3x3, 64, S=1, P=1) layers	64x56x56
2 residual blocks, each has 2 Conv(3x3, 128) layers	128x28x28
2 residual blocks, each has 2 Conv(3x3, 256) layers	256x14x14
$\lfloor (28 + 2 * 1 - 3 + 2)/2 \rfloor = 14$ for 1st layer in 1st block	
(14 + 2*1 + 1 - 3)/1 = 14 for other layers	
skip connection for 1st block is not identity, but Conv(2x2,256, S=2, P=0)	

Recall: $\lfloor \frac{N+2P-D(F-1)-1}{S} \rfloor + 1$, equal to $\lfloor \frac{N+2P-F+S}{S} \rfloor$ when $D = 1$.

Example: An 18-layer ResNet



S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3)	64x112x112
MaxPool(3x3, S=2, P=1)	64x56x56
2 residual blocks, each has 2 Conv(3x3, 64, S=1, P=1) layers	64x56x56
2 residual blocks, each has 2 Conv(3x3, 128) layers	128x28x28
2 residual blocks, each has 2 Conv(3x3, 256) layers	256x14x14
2 residual blocks, each has 2 Conv(3x3, 512) layers	512x7x7

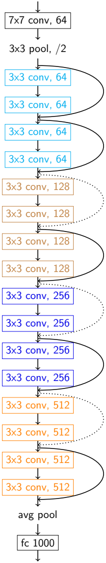
$\lfloor (14 + 2 * 1 - 3 + 2)/2 \rfloor = 7$ for 1st layer in 1st block

(7 + 2*1 + 1 - 3)/1 = 7 for other layers

skip connection for 1st block is not identity, but Conv(2x2,512, S=2, P=0)

Recall: $\lfloor \frac{N+2P-D(F-1)-1}{S} \rfloor + 1$, equal to $\lfloor \frac{N+2P-F+S}{S} \rfloor$ when $D = 1$.

Example: An 18-layer ResNet

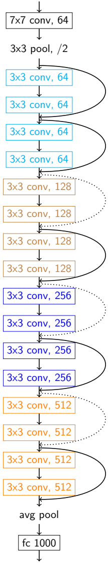


S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3)	64x112x112
MaxPool(3x3, S=2, P=1)	64x56x56
2 residual blocks, each has 2 Conv(3x3, 64, S=1, P=1) layers	64x56x56
2 residual blocks, each has 2 Conv(3x3, 128) layers	128x28x28
2 residual blocks, each has 2 Conv(3x3, 256) layers	256x14x14
2 residual blocks, each has 2 Conv(3x3, 512) layers	512x7x7
Global average pooling Each feature map is averaged	512x1x1

Recall: $\lfloor \frac{N+2P-D(F-1)-1}{S} \rfloor + 1$, equal to $\lfloor \frac{N+2P-F+S}{S} \rfloor$ when $D = 1$.

Example: An 18-layer ResNet

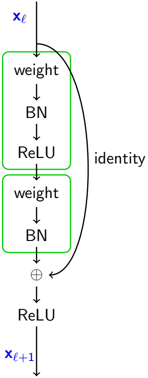


S: stride; P: padding

layer	output size
input	3x224x224
Conv(7x7, 64, S=2, P=3)	64x112x112
MaxPool(3x3, S=2, P=1)	64x56x56
2 residual blocks, each has 2 Conv(3x3, 64, S=1, P=1) layers	64x56x56
2 residual blocks, each has 2 Conv(3x3, 128) layers	128x28x28
2 residual blocks, each has 2 Conv(3x3, 256) layers	256x14x14
2 residual blocks, each has 2 Conv(3x3, 512) layers	512x7x7
Global average pooling	512x1x1
Fully connected layer with 1000 outputs	1000

- The ResNet18 architecture in the original paper in fact applies batch normalization after each convolution, but before activation.

- Full architecture of a residual block in ResNet18



Equations relating $\mathbf{x}_\ell$ (l-th layer output) and $\mathbf{x}_{\ell+1}$

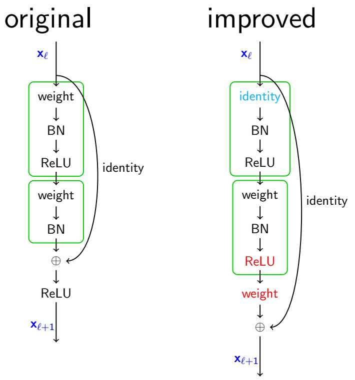
$$\mathbf{y}_\ell = \mathbf{x}_\ell + f(\mathbf{x}_\ell),$$
$$\mathbf{x}_{\ell+1} = \text{ReLU}(\mathbf{y}_\ell)$$

NB. Sometimes identity is replaced by downsampling.

- ResNet architectures in the original paper (including the 152-layer ImageNet winning architecture)

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Improved Residual Block



Original residual block

$$\mathbf{y}_\ell = \mathbf{x}_\ell + f(\mathbf{x}_\ell),$$
$$\mathbf{x}_{\ell+1} = \mathrm{ReLU}(\mathbf{y}_\ell)$$

ReLU prevents direct information flow between $\mathbf{x}_\ell$ and $\mathbf{x}_{\ell+1}$ for both forward and backward propagation.

Idea: move ReLU and weight around to improve information flow.

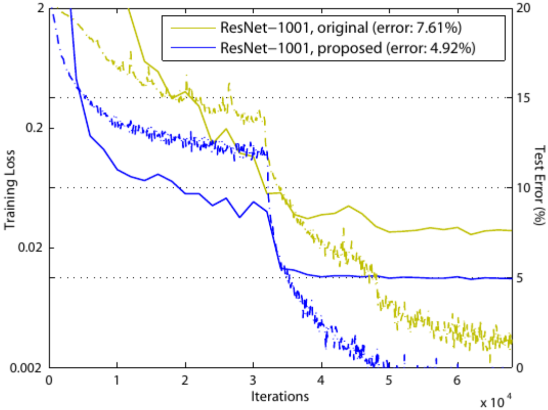
Improved residual block

$$\mathbf{x}_{\ell+1} = \mathbf{x}_\ell + f(\mathbf{x}_\ell),$$

Direct infomation flow between $\mathbf{x}_\ell$ and $\mathbf{x}_{\ell+1}$!

NB. The identity layer is only there for better comparison of the differences.

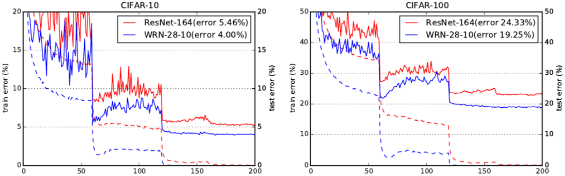
He et al., Identity mappings in deep residual networks, 2016



- Original design overfits with 200 layers.
- Improved design makes 1001-layer ResNet trainable with good generalization performance (similar to original 152-layer ResNet).

Wide Residual Networks

- Wider residual blocks (F x k filters instead of F filters in each layer)



solid: train; dashed: test

  28-layer wide ResNet ($k = 10$) outperforms 164-layer orignal ResNet

- Perhaps residuals are the important factor, not depth.
- Increasing width instead of depth is more computationally efficient.

Zagoruyko and Komodakis, Wide residual networks, 2016

Highway Networks

- Highway networks use shortcut connections (same as ResNet), but use data-dependent gates to control information flow (ResNet uses parameter-free shortcuts).
- A block $f_{\mathbf{w}}(\mathbf{x})$ is modified to compute $f_{\mathbf{w}}(\mathbf{x}) \odot T_{\mathbf{w}_T}(\mathbf{x}) + \mathbf{x} \odot C_{\mathbf{w}_C}(\mathbf{x})$.
  - $\odot$ is the Hadamard product
  - $T_{\mathbf{w}_T}$ is a nonlinear transformation (transform gate)
  - $C_{\mathbf{w}_C}$ is a nonlinear transformation (carry gate)
- Typically, $C = 1 - T$, thus the block can smoothly vary its behavior between identity and $f_{\mathbf{w}}(\mathbf{x})$.

Srivastava, Greff, and Schmidhuber, Highway networks, 2015

Your Turn

Which of the following statement is correct? (Multiple choice)

(a) Deep CNNs are usually easy to train.

(b) Shortcut connections are used in AlexNet.

(c) In ResNet, the shortcut connection may not be an identity and may be learned.

(d) GoogLeNet is a wide residual network.

What You Need to Know

- Deeper networks are generally harder to train
- Shortcut connection, residual block and ResNet
- Relatives: improved residual block, wide residual networks, highway networks.