Case Studies

- Classic Networks :
  - → LeNet - 5
  - → AlexNet
  - → VGG
- ResNet (152 - layer)
- Inception

Classic LeNet-5

60k Parameters

32x32x1 Input $\xrightarrow[f: 5x5,\ s=1]{Conv}$ 28x28x6

$n_H \times n_W \downarrow$

$\xrightarrow[f=2,\ s=2]{Avg\ Pool}$ 14x14x6

$n_C \uparrow$

NonLinearity After Pooling

$\xrightarrow[f: 5x5,\ s=1]{Conv}$ 10x10x16

$\xrightarrow[f=2,\ s=2]{Avg\ Pool}$ 5x5x16

NonLinearity After Pooling

$\xrightarrow{FC}$ 120

$\xrightarrow{FC}$ 84

$\longrightarrow$ $\hat{y}$

Paper Section 2 & 3

AlexNet

~ 60 M Parameters

227x227x3 $\xrightarrow[11x11,\ s=4]{Conv}$ 55x55x96

$\xrightarrow[3x3,\ s=2]{Max\ Pool}$ 27x27x96

$\xrightarrow[5x5,\ same]{Conv}$ 27x27x256

$\xrightarrow[3x3,\ s=2]{Max\ Pool}$ 13 x 13 x 256

$\xrightarrow[3x3,\ same]{Conv}$ 13x13 x384

$\xrightarrow[3x3]{Conv}$ 13 x 13 x 384

$\xrightarrow[3x3]{Conv}$ 13 x 13 x 256

$\xrightarrow[3x3\ s=2]{Max\ Pool}$ 6 x 6 x 256

$\xrightarrow{Unroll}$ 9216

$\xrightarrow{FC}$ 4096

$\xrightarrow{FC}$ 4096

$\longrightarrow$ Softmax 1000

Easy to Read

Relu Activation

Much bigger than LeNet

Multiple GPU

Local Response Normalization

VGG-16 (16 Layers)

~138 M Parameters

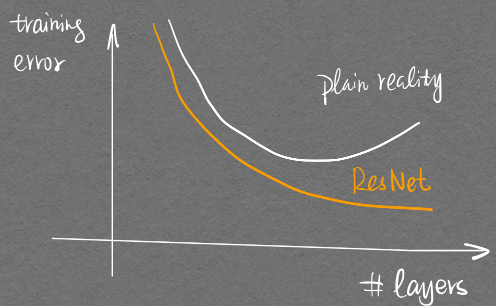
$n_H, n_W \downarrow$ $n_C \uparrow$

ResNets

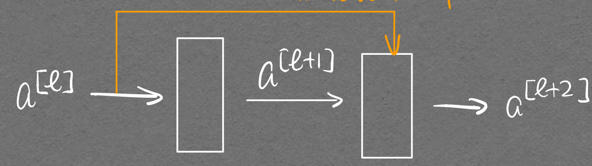
Deep Neural Network

Vanishing Gradients
Exploding Gradients } → ResNets

He et al. Deep Residual Networks for Image Recognition 2015



Residual Block

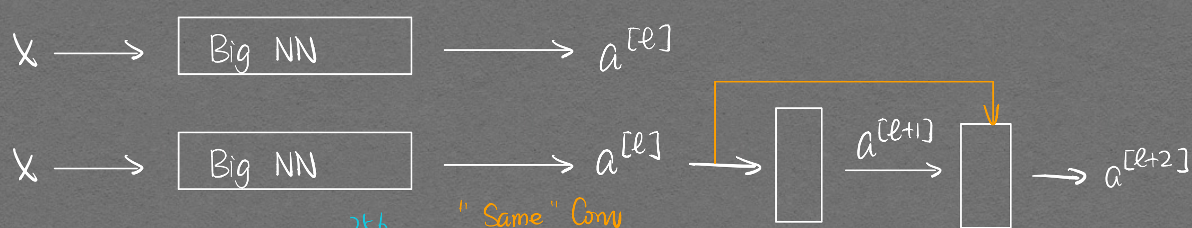


$a^{[l]} \to \text{linear} \to \text{Relu} \to \text{linear} \to \text{Relu} \to a^{[l+2]}$

main Path

$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]}$$
$$a^{[l+1]} = g(z^{[l+1]})$$
$$z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+1]}$$
$$a^{[l+2]} = g(z^{[l+2]}) \times$$
$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

Why ResNets Work?



"Same" Conv
for adding the two term

$$a^{[l+2]} = g(\underbrace{z^{[l+2]}}_{256} + \underbrace{a^{[l]}}_{128})$$

$W_s \in \mathbb{R}^{256 \times 128}$ Train or Fix (0)

$$= g(W^{[l+2]} a^{[l+1]} + b^{[l+2]} + W_s\, a^{[l]})$$

g() : Relu Activation

If $W^{[l+2]}, b^{[l+2]} = 0$

$$= g(a^{[l]})$$
$$= a^{[l]} \quad ?$$

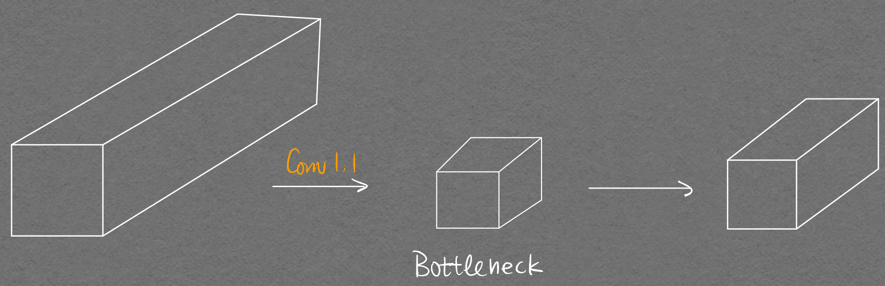
Identity function is easy for Residual block to learn!

1×1 Convolution / Network in Network

Adds non-linearity to neural network

Shrink / keep the same / Increase n_c

Inception Network



Using Open-Source Implementations

git clone

Transfer Learning

Download code & weights

Remove the original softmax layer

Put on your own softmax layer

Freeze weights in previous layers (trainable = 0 , freeze = 1)

Trick: precompute activations and save them on the disk.

More Data → Fewer Freeze Layers → Train more layers

(Extreme Case: initialize with others' parameters and retrain all layers)

Data Augmentation

Mirroring

Random Cropping

Color Shifting (加滤镜) PCA Color Augmentation

The State of Computer Vision

Little Data ⟷ Lots of Data

More hand-engineering ("Hacks")

Object Detection

Image Recognition

Speech Recognition

Simpler Algorithms

Less Hand-engineering

Two Sources of Knowledge:

→ Labeled Data

→ Hand Engineered Features / Neural Network Architecture

Tips for doing well on benchmarks / winning competitions

1. Ensembling
2. Multi-crop at Test time   Run classifier on multiple versions of test images and average results
3. Use Open Source Code