SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of AI &DS

Course Name – 19AD602 DEEP LEARNING

III Year / VI Semester

Unit 3-DIMENSIONALITY REDUCTION
Topic: Introduction to Convnet-Inception, ResNet

Case Study:
A tech startup developed a facial recognition system using Inception for feature extraction and ResNet for face classification. Inception's multi-scale convolution modules improved feature representation, while ResNet's skip connections tackled vanishing gradient issues, enabling accurate and efficient recognition of diverse faces.

Activity: Practical Experiment

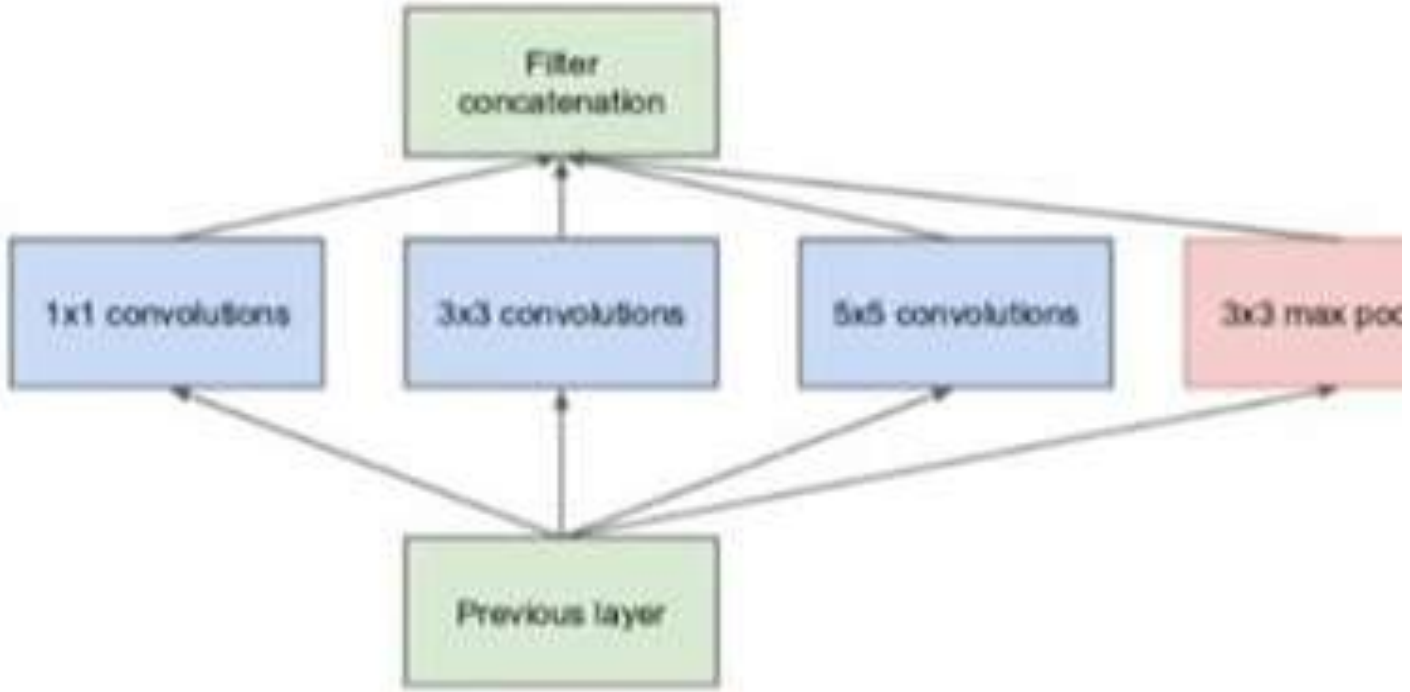
1. **Experiment Setup**: Train two image classification models on CIFAR-10:
   - One using the Inception architecture.
   - Another using ResNet-50.
2. **Objective**: Compare model performance in terms of accuracy, training time, and ability to generalize to unseen data.
3. **Deliverables**: Document key findings, including visualization of loss curves, accuracy trends, and qualitative analysis of model outputs.

Inception Network

- Motivation:
  - Kernels with different sizes because object is distributed differently in different images
  - Deep networks also cause learning problems and overfitting
- Solution:
  - Filters / Kernels with different sizes on same level, i.e. widen network instead of going deeper

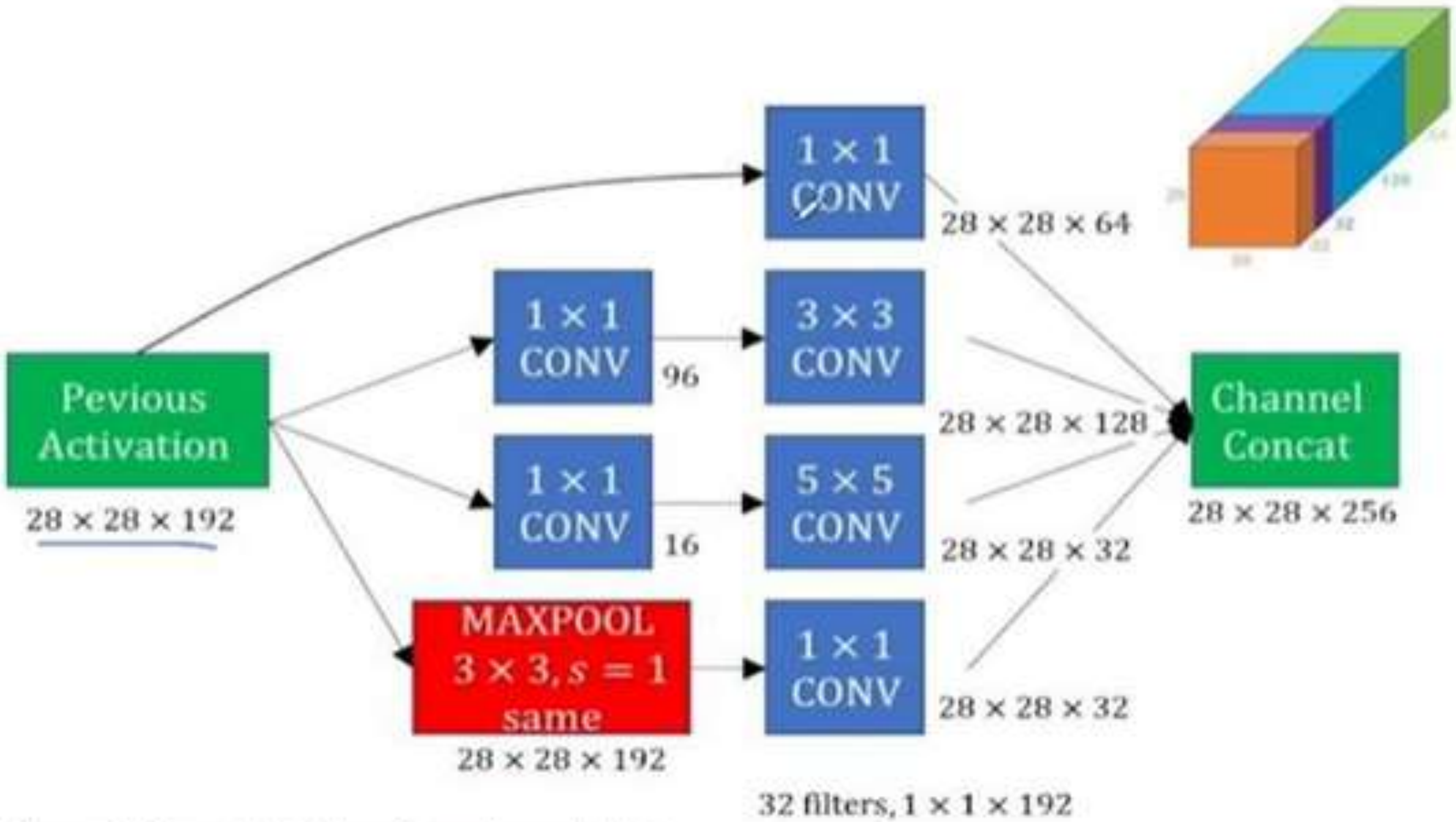
Inception Network

- Convolution with different sizes
- Along with max pooling
- All output are concatenated



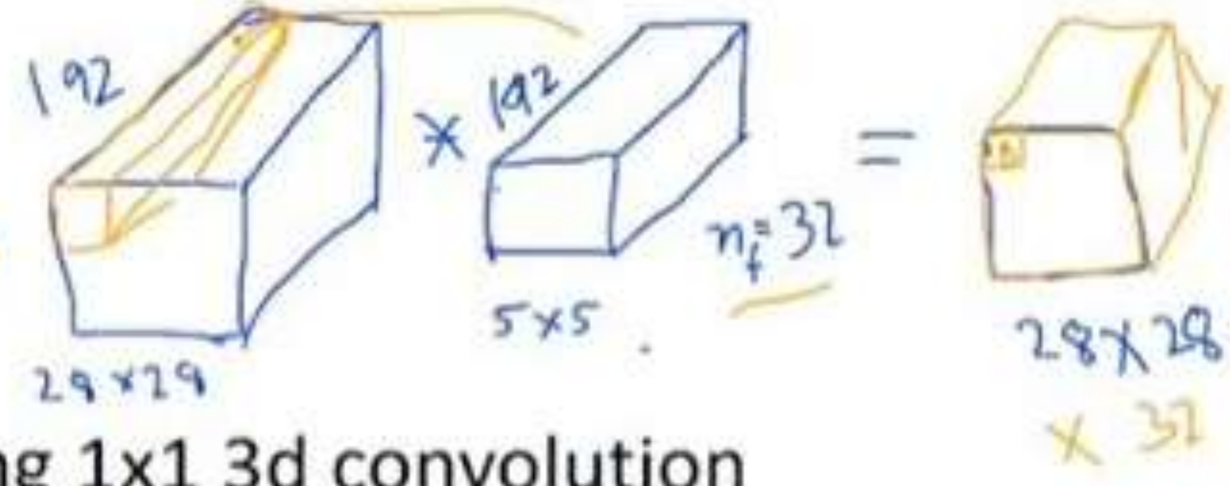
(a) Inception module, naïve version

Inception Layer Example



Inception v1 (GoogleNet has 9 such modules)

Computational Complexity



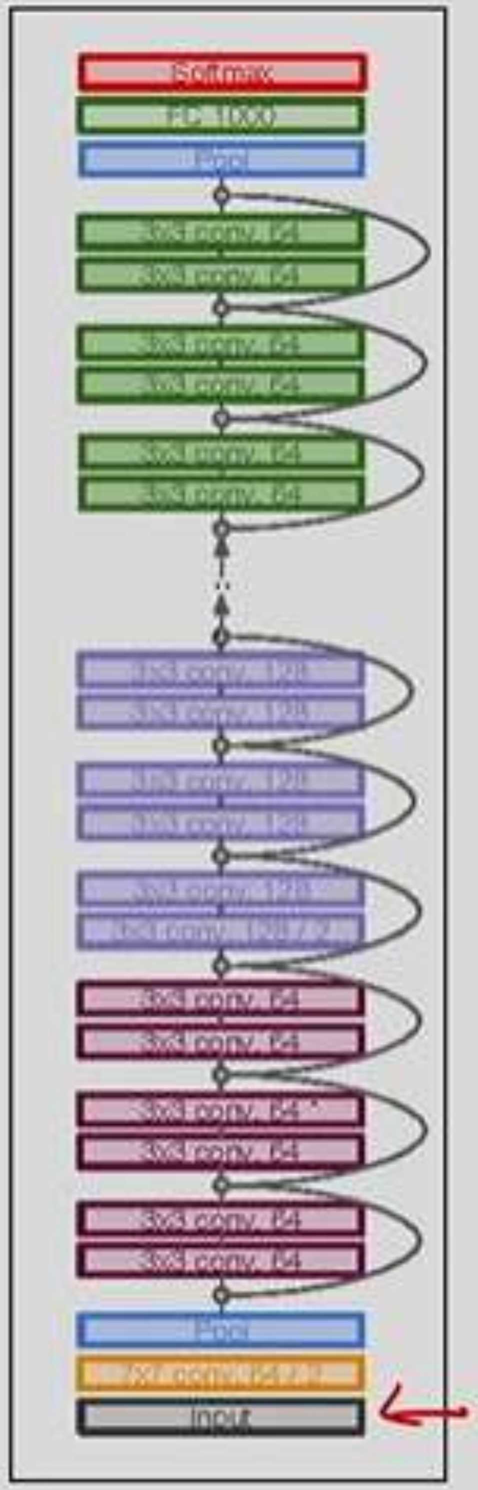
- Reducing computational Complexity using 1x1 3d convolution

- 28 x 28 x 192 ---- > 5x5 Conv, nc=32, same --- > 28 x 28 x 32
  - Computation complexity = 120 million (multiplications) 5x5x192 x 28x28x32 = 120,422,400

- Using 1 x 1 Convolution
  - 28 x 28 x 192 ---- > 1x1 Conv, nc=16, --- > Intermediate is 28x28x16, --- > 5x5 Conv, nc=32, same --- > 28 x 28 x 32
  - 2.4M +10 M = 12.4 Million (multiplications) 1x1x192 x 28x28x16 + 5x5x16 x 28x28x32

ResNet

- *Deep Residual Learning for Image Recognition - Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; 2015*
- Extremely deep network – 152 layers
- Deeper neural networks are more difficult to train.
- Deep networks suffer from vanishing and exploding gradients.
- Present a residual learning framework to ease the training of networks that are substantially deeper than those used previously.

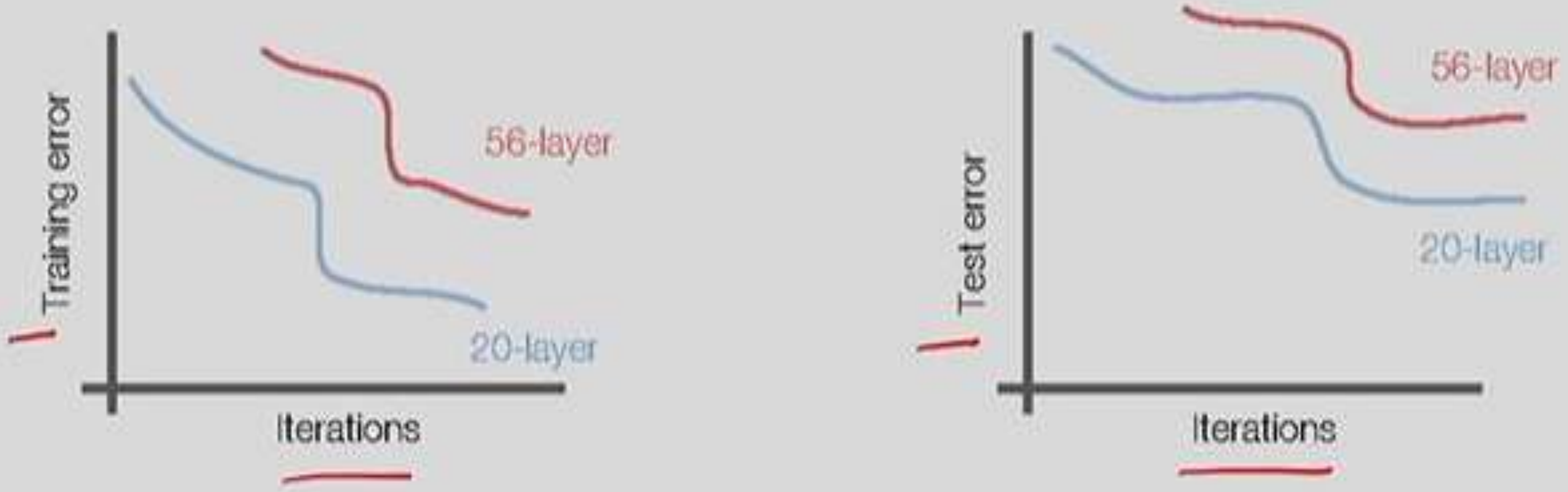


ResNet

- ILSVRC’15 classification winner (3.57% top 5 error, humans generally hover around a 5-10% error rate)
  Swept all classification and detection competitions in ILSVRC’15 and COCO’15!

ResNet

- What happens when we continue stacking deeper layers on a convolutional neural network?



- 56-layer model performs worse on both training and test error

-> The deeper model performs worse (not caused by overfitting)!

ResNet

- **Hypothesis**: The problem is an optimization problem. Very deep networks are harder to optimize.
- **Solution**: Use network layers to fit residual mapping instead of directly trying to fit a desired underlying mapping.

- We will use **skip connections** allowing us to take the activation from one layer and feed it into another layer, much deeper into the network.
- Use layers to fit residual $F(x) = H(x) - x$ instead of $H(x)$ directly

i/p

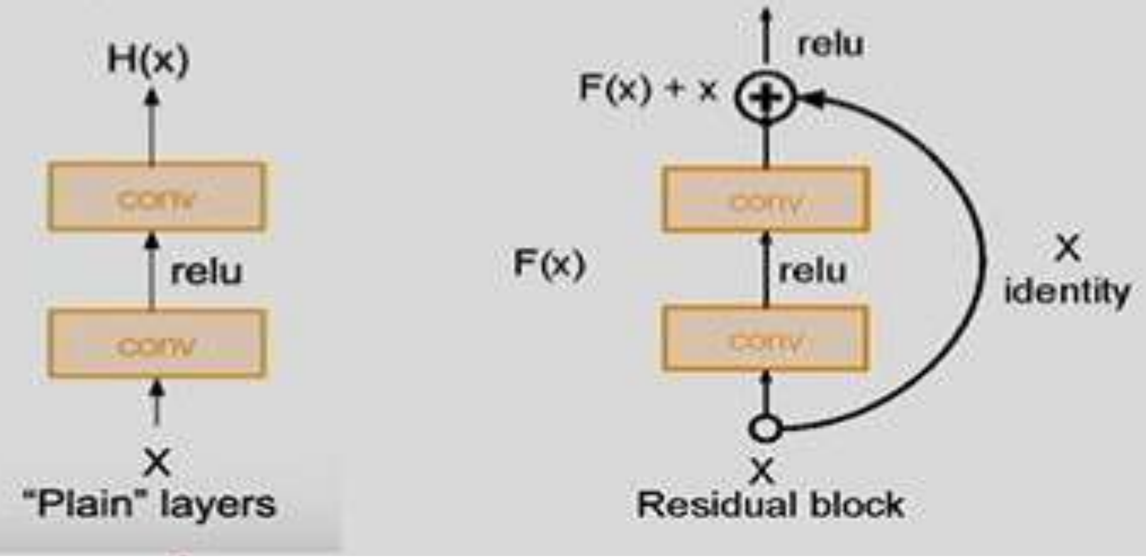
o/p

ResNet

Residual Block

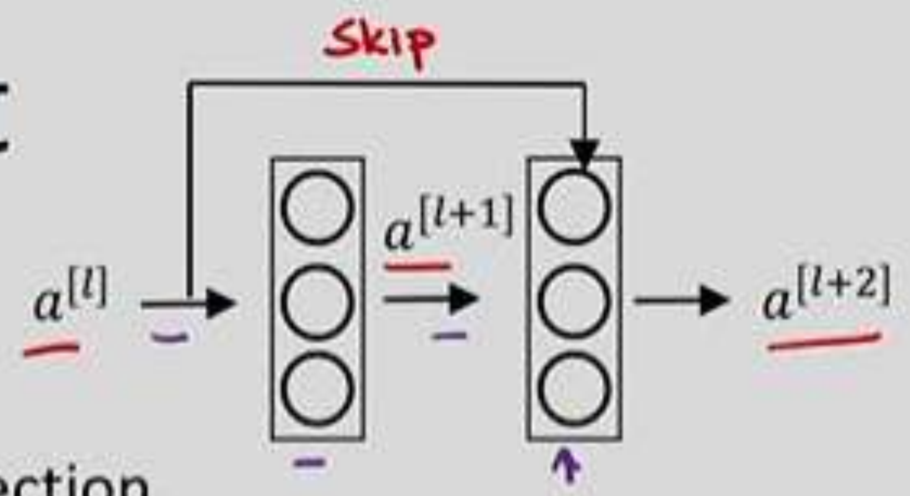
Input x goes through conv-relu-conv series and gives us F(x). That result is then added to the original input x. Let's call that H(x) = F(x) + x.

In traditional CNNs, H(x) would just be equal to F(x). So, instead of just computing that transformation (straight from x to F(x)), we're computing the term that we have to *add*, F(x), to the input, x.



[He et al., 2015]

ResNet



Short cut/ skip connection

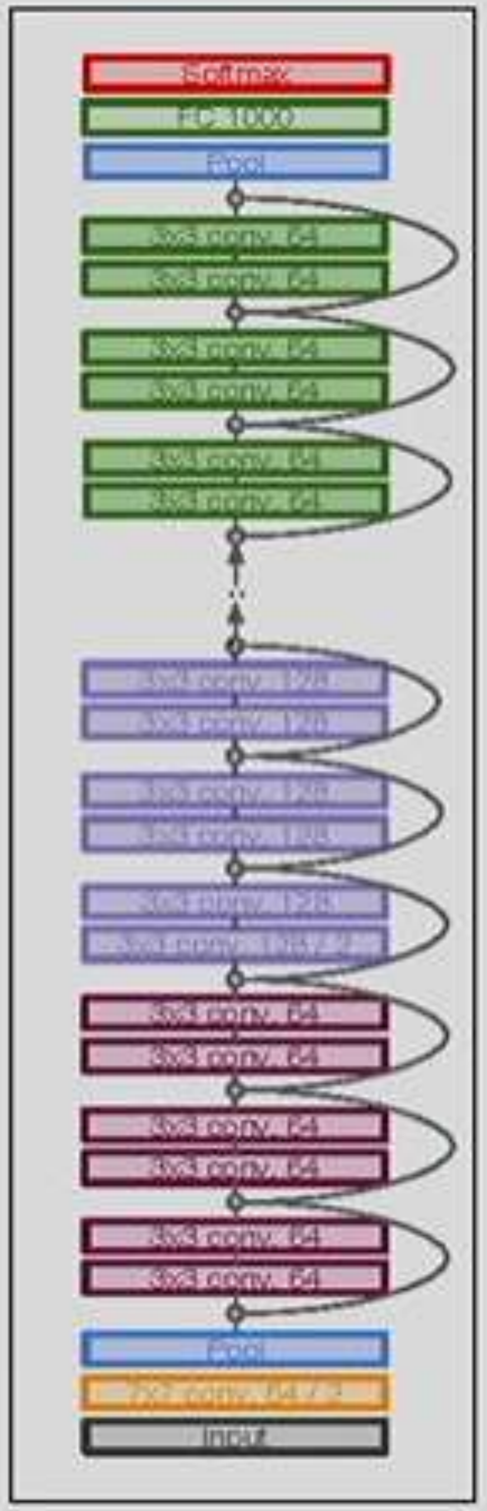
$a^{[l]} \rightarrow$ Linear $\rightarrow$ ReLU $\rightarrow$ Linear $\rightarrow$ ReLU $\rightarrow a^{[l+2]}$

$a^{[l+1]}$

$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \qquad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]}$$

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

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



ResNet

Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (in each dimension)
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes)

THANK YOU