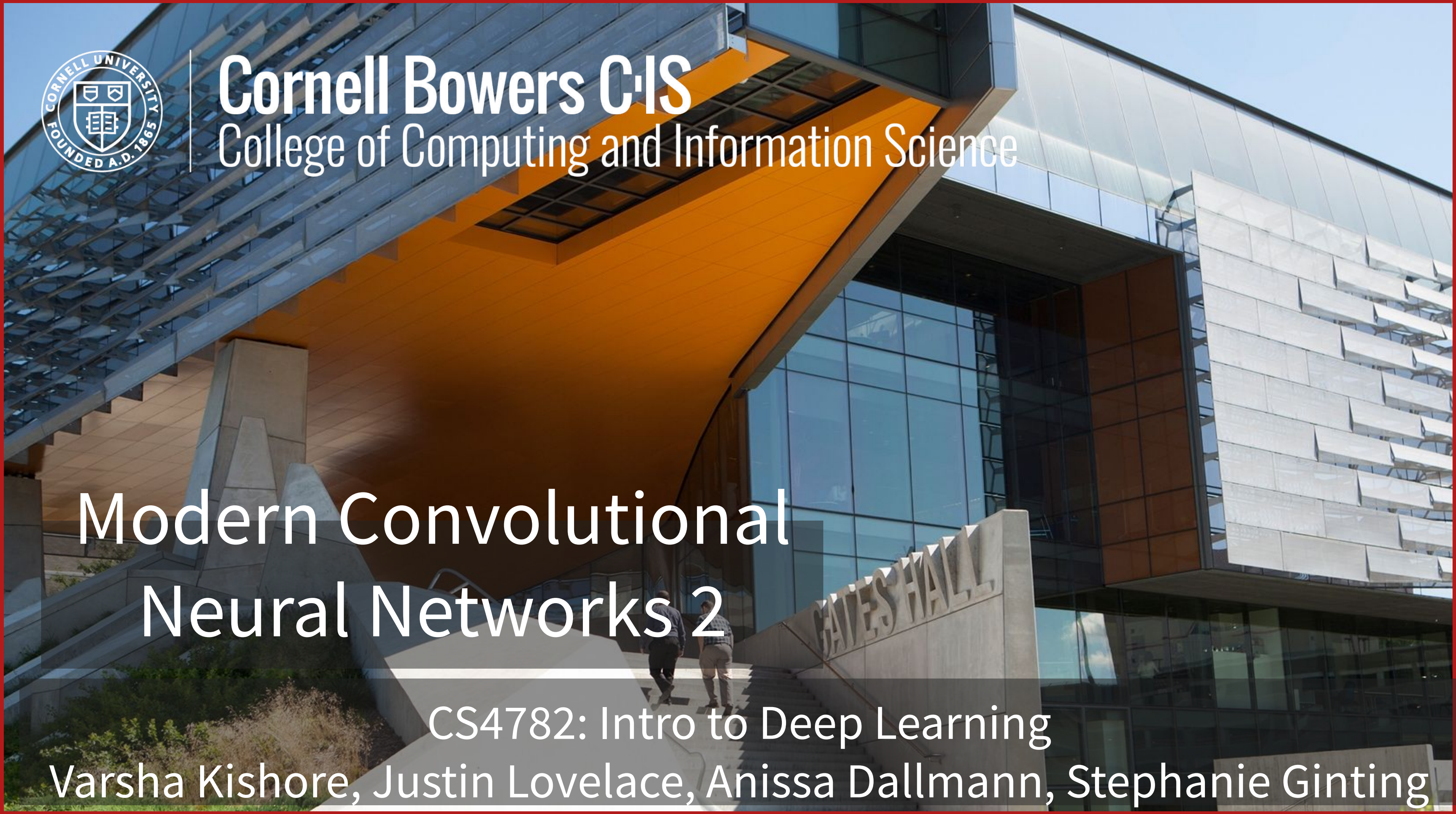


Cornell Bowers C·IS
College of Computing and Information Science

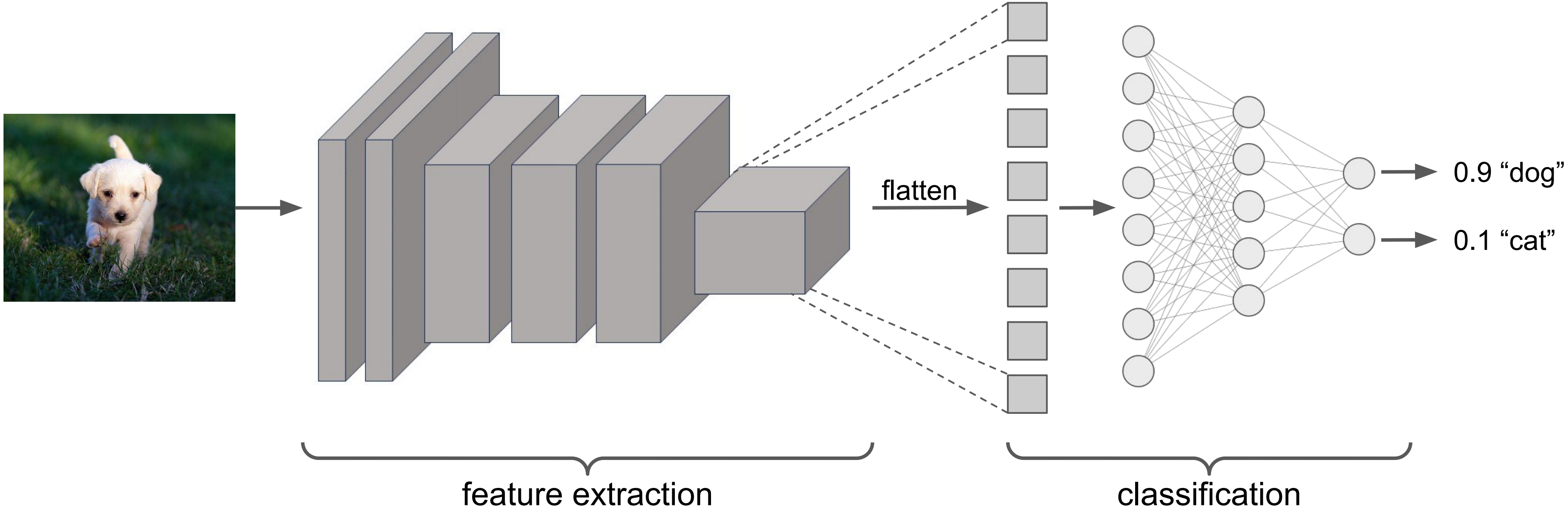
Modern Convolutional Neural Networks 2

CS4782: Intro to Deep Learning

Varsha Kishore, Justin Lovelace, Anissa Dallmann, Stephanie Ginting

Cornell Bowers C·IS

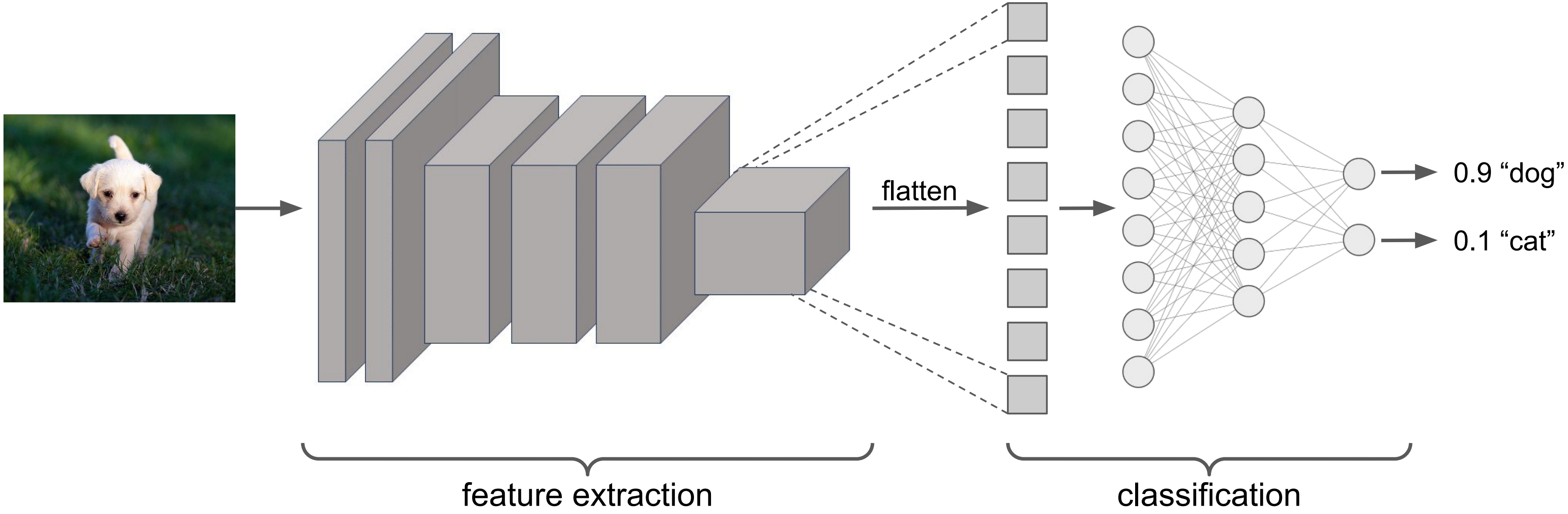
Review: Image Classification



Cornell Bowers C·IS

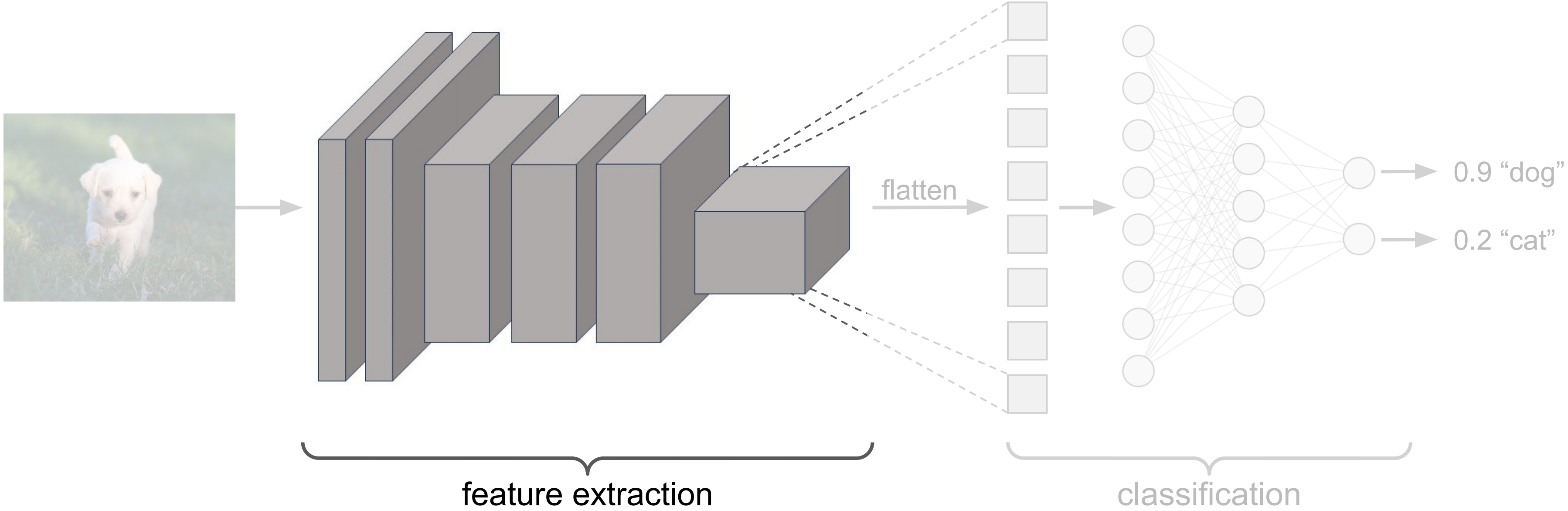
Image Classification

- Important: Everything is differentiable!
- Can calculate gradient of the loss with backpropagation
  - Train with SGD/Adam/etc.
  - Learn convolutional filters and classification head end-to-end!



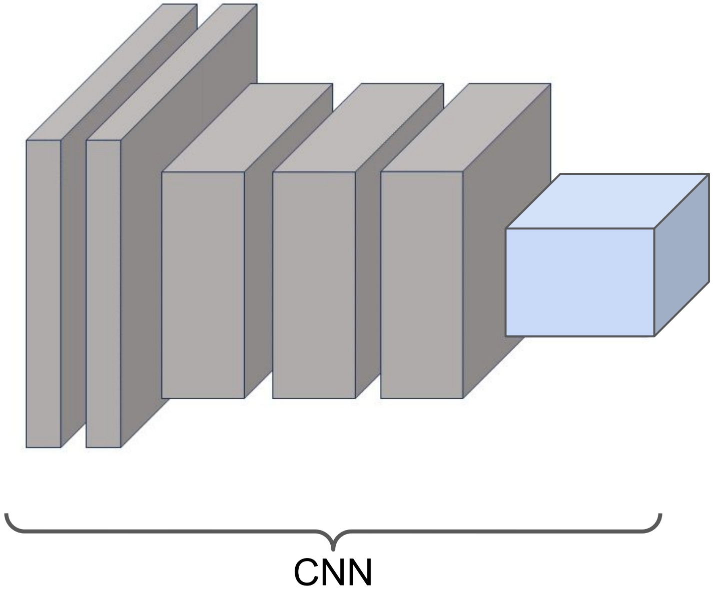
Cornell Bowers C·IS

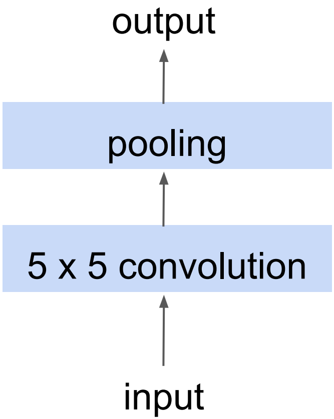
Deeper CNN Architectures



Cornell Bowers C·IS

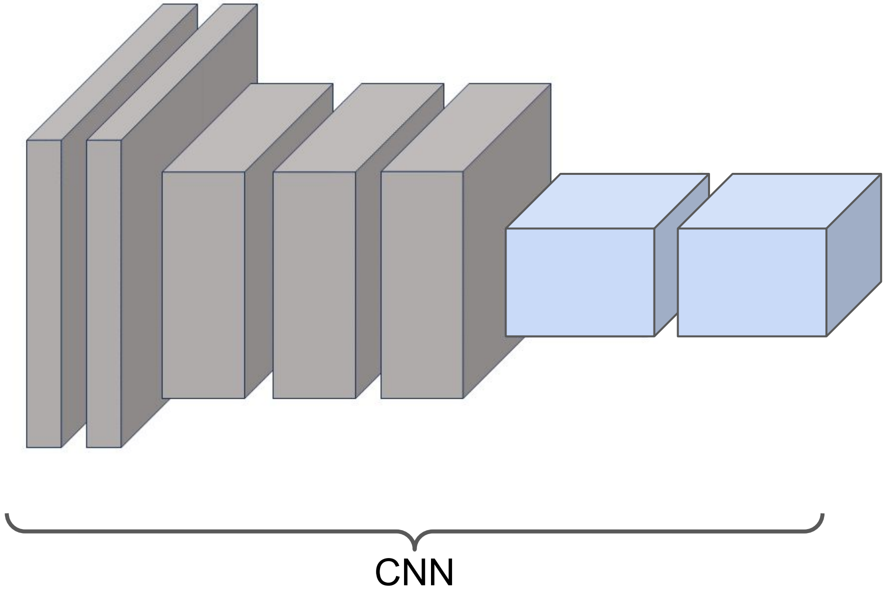
Deeper CNN Architectures

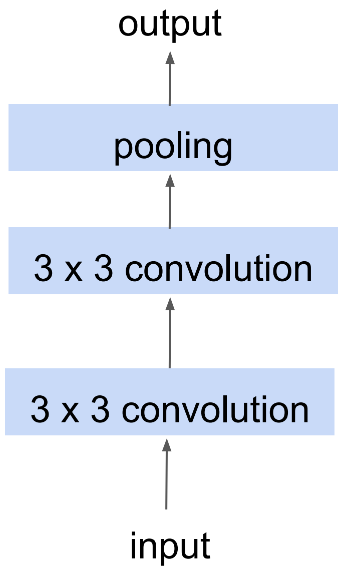




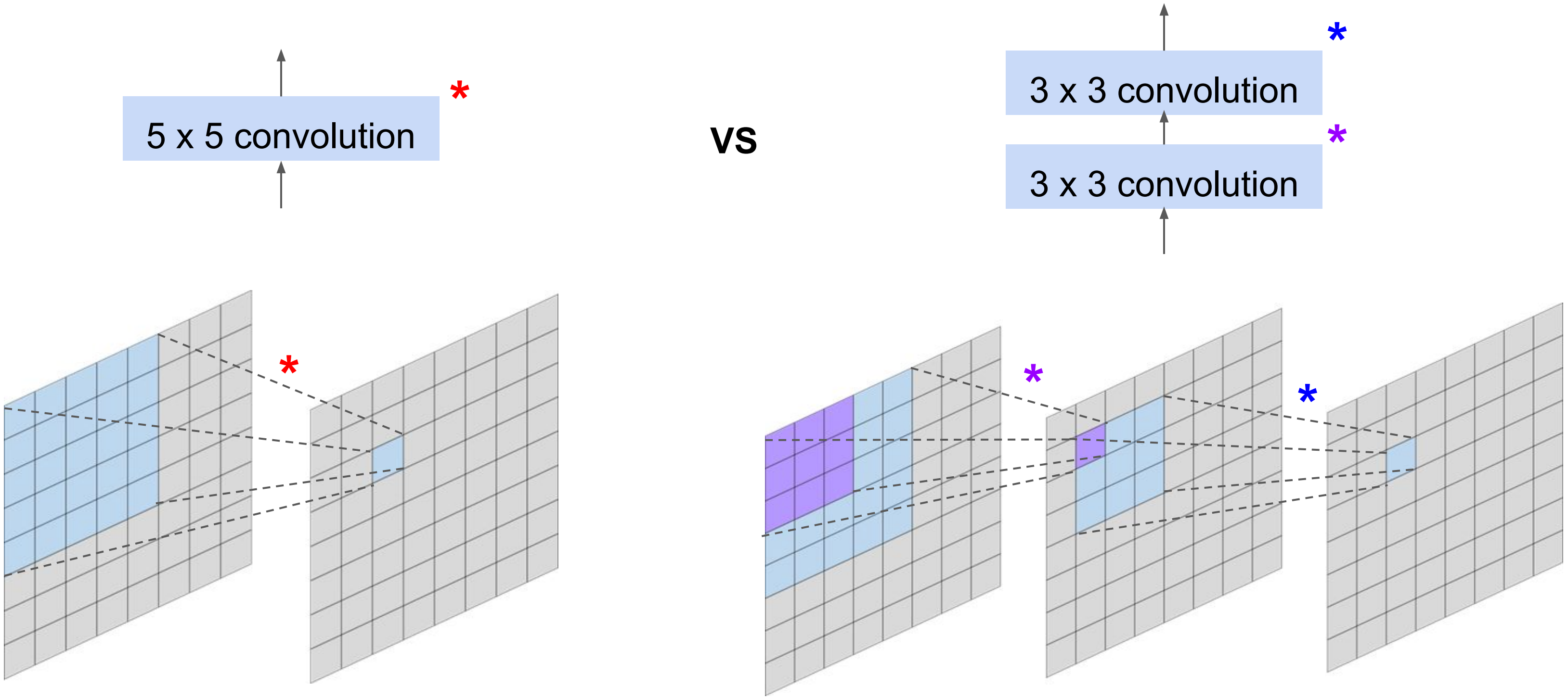
Cornell Bowers C·IS

Deeper CNN Architectures



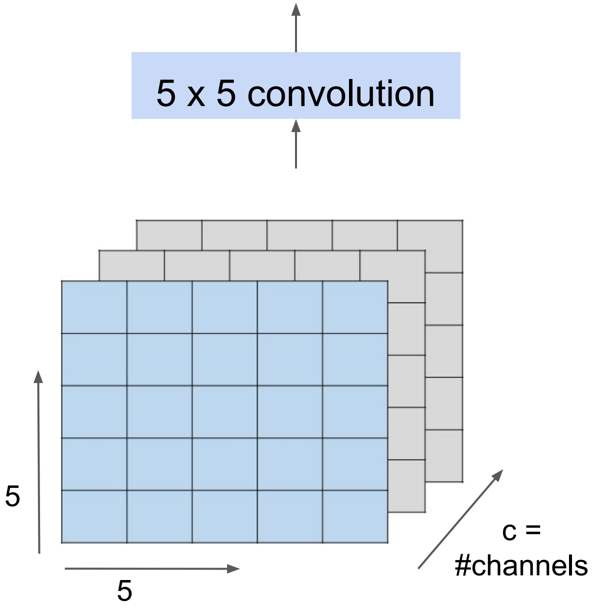


Deeper CNN Architectures



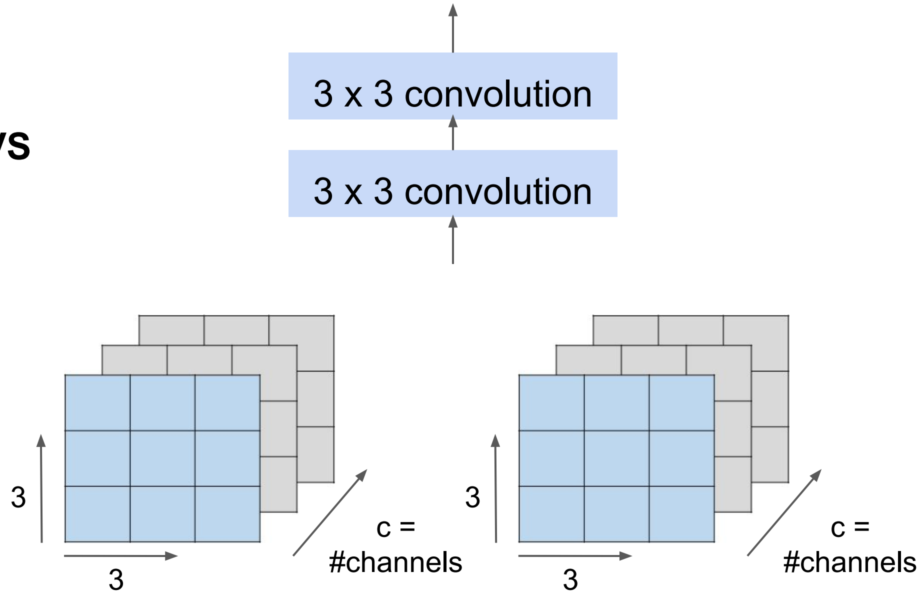
Cornell Bowers C·IS

Deeper CNN Architectures



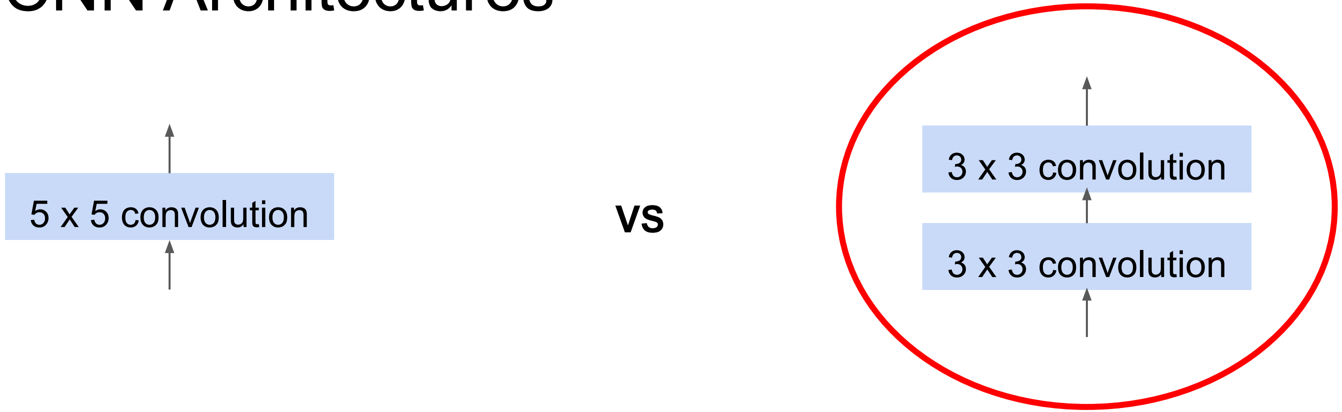
$$5 * 5 * c^2$$
$$= 25c^2 \text{ parameters}$$

vs



$$2 * 3 * 3 * c^2$$
$$= 18c^2 \text{ parameters}$$

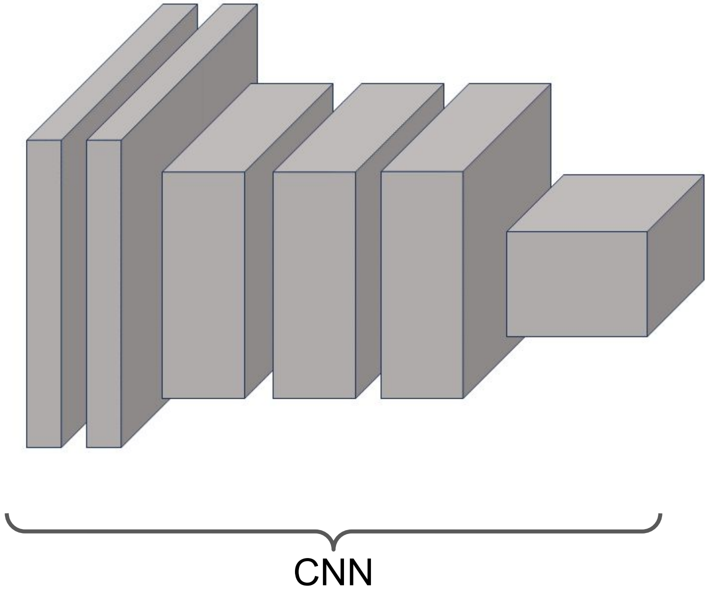
Deeper CNN Architectures



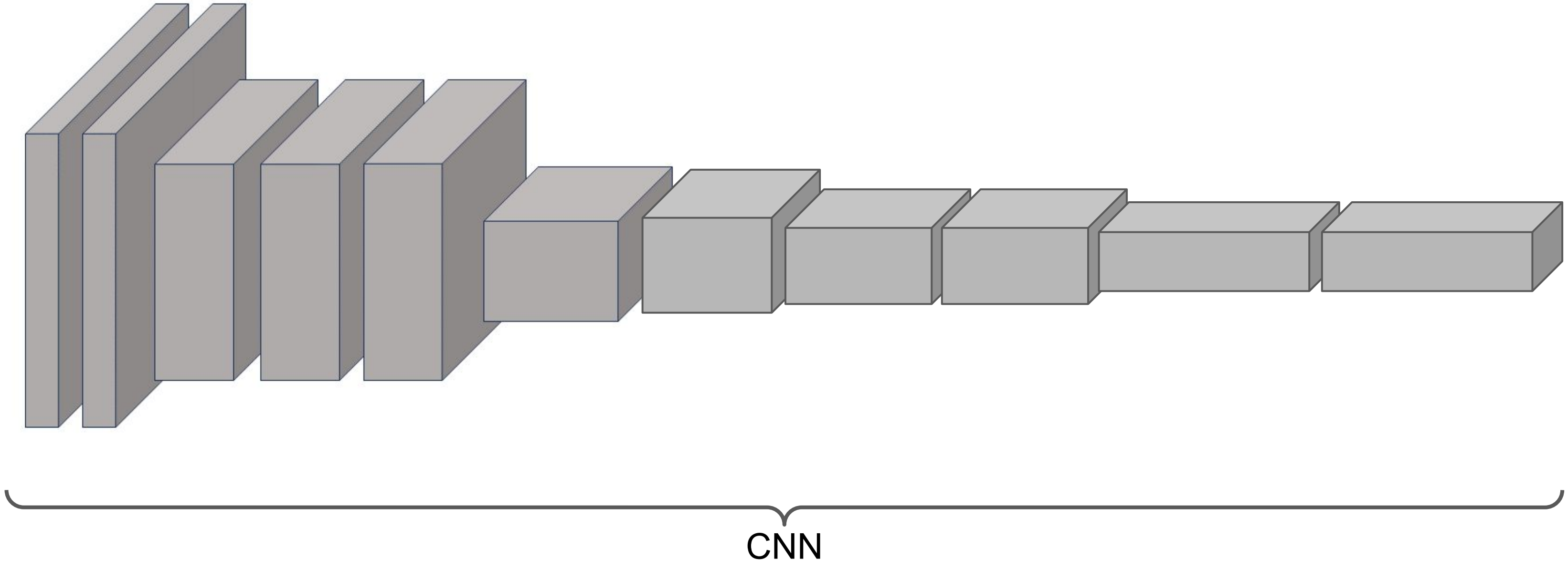
Performed better!

Cornell Bowers C·IS

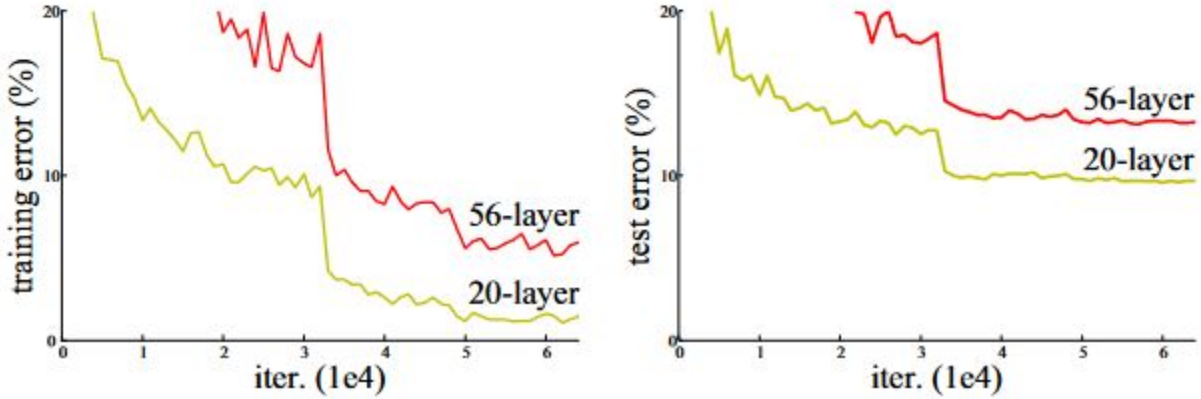
Deeper == better



Deeper == better



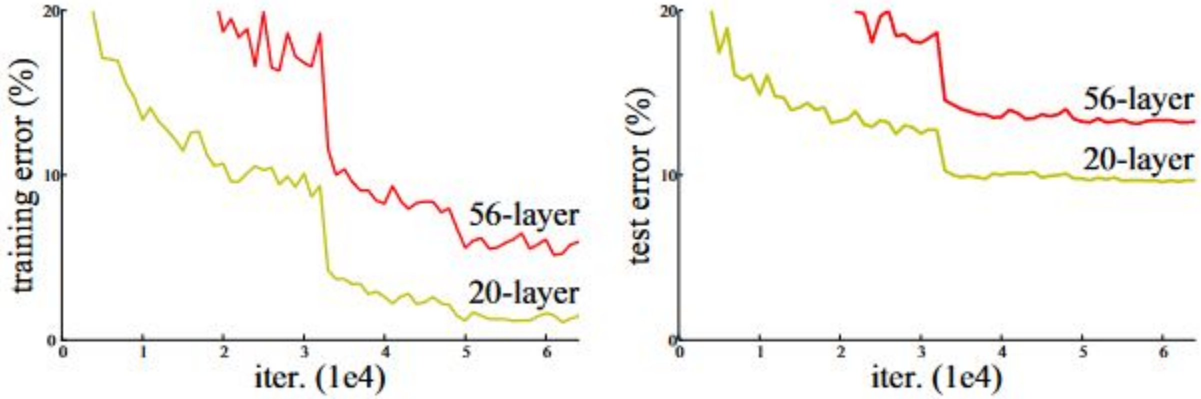
Deeper == better?



56 layer CNN has higher training and test error than 20 layer CNN on CIFAR-10 dataset for image classification

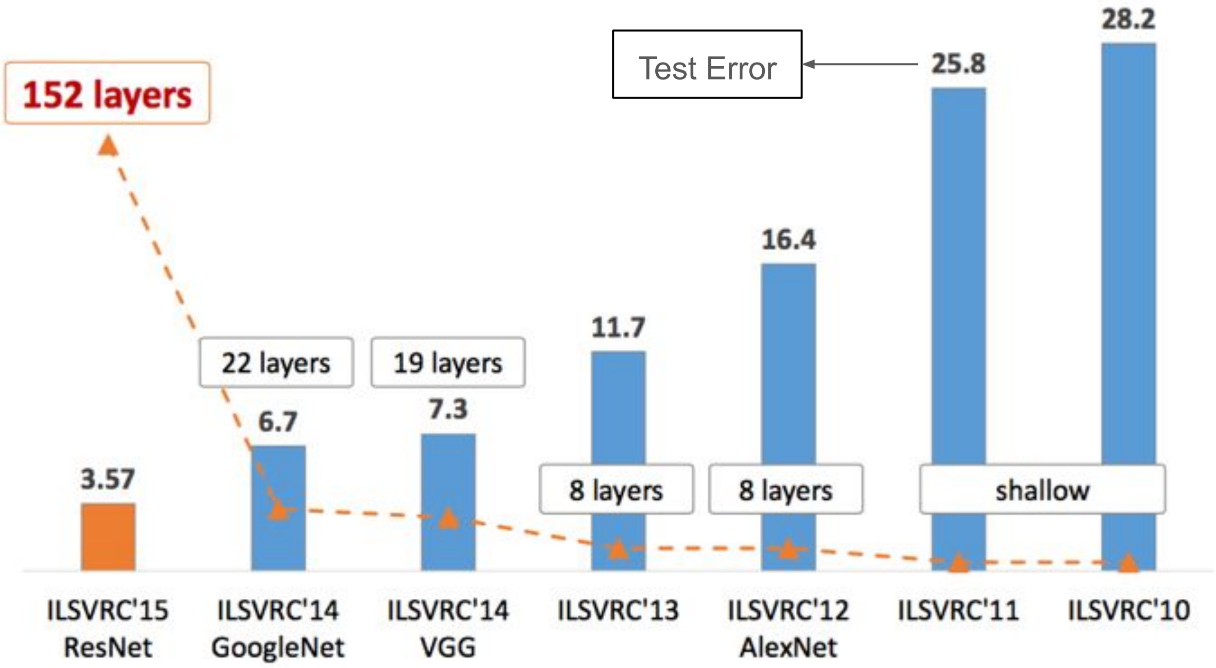
[He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.]

Discuss: How can a larger network achieve a higher training error?



56 layer CNN has higher training and test error than 20 layer CNN on CIFAR-10 dataset for image classification

ImageNet Classification Challenge: Deeper == better



[Nguyen, Kien & Fookes, Clinton & Ross, Arun & Sridharan, Sridha. (2017). Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective. IEEE Access. PP. 1-1. 10.1109/ACCESS.2017.2784352.]

GoogLeNet/Inception Net

Goal: given a fixed computational budg

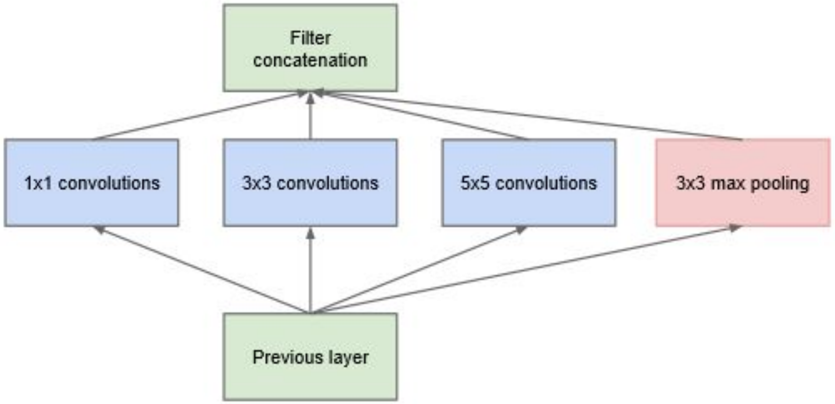
network

=> Deeper networks with computationa

In this paper, we will focus on an efficient deep neural network architecture for computer vision, codenamed Inception, which derives its name from the Network in network paper by Lin et al [12] in conjunction with the famous "we need to go deeper" internet meme [1]. In our case, the word

[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

Inception Module



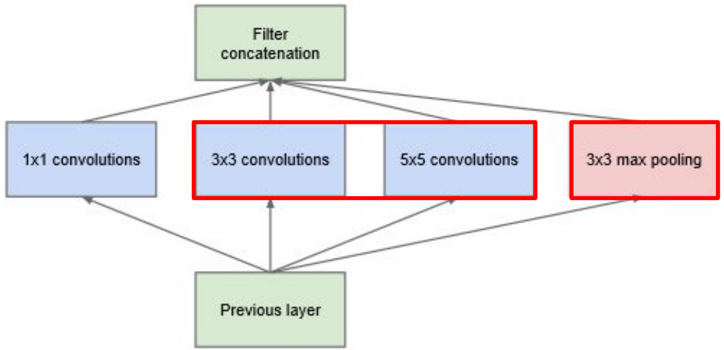
Inception module = main building blocks

[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

Cornell Bowers C·IS

Inception Module

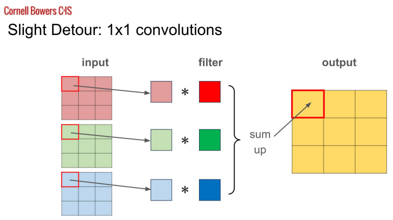
Still expensive!

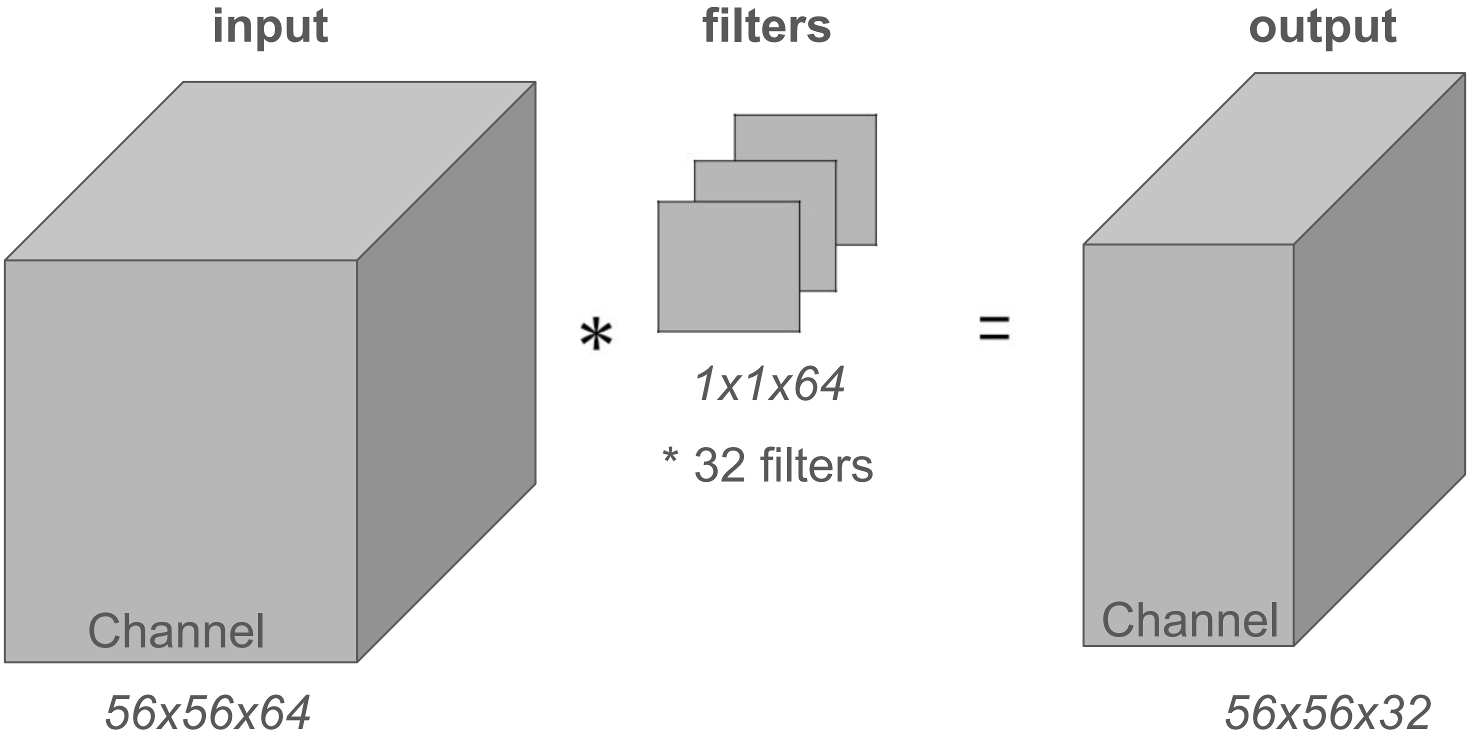


- 3x3 and 5x5 convolutions have large number of operations
- Output of pooling layer increases the output channel dimension when concatenated

[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

Remember: 1x1 convolutions





Discuss: Impact of Dimension Reduction

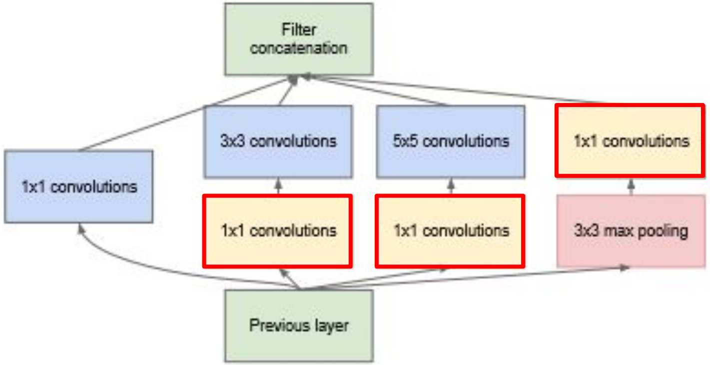
Assume you have an input feature map with 256 channels/features.

Compare the parameter counts from:

1. 3x3 conv with 256 filters
2. 1x1 conv with 64 filters → 3x3 conv with 64 filters → 1x1 conv with 256 filters

Inception Module

Solution: Inception module with dimension reduction

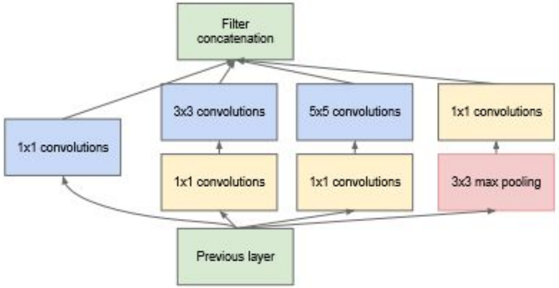


- “Bottleneck” with 1x1 convolutions to reduce dimensions

[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

GoogLeNet Architecture

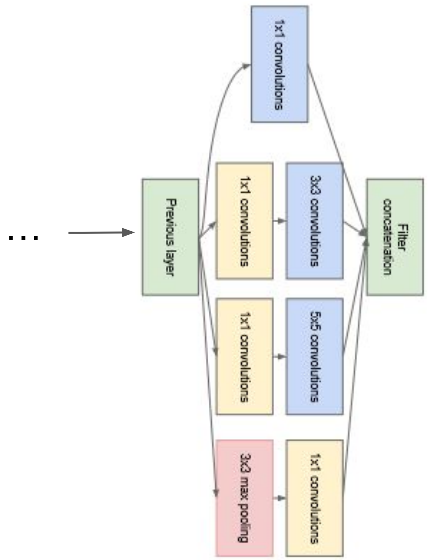
Key idea: stack inception modules together



[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

GoogLeNet Architecture

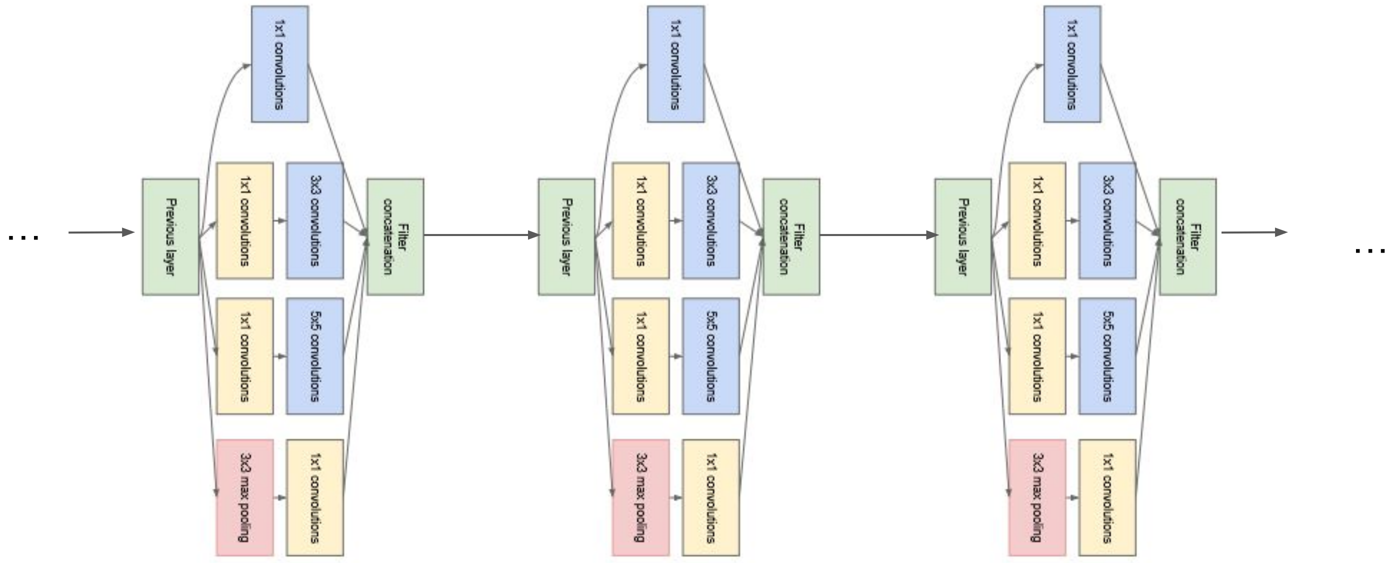
Key idea: stack inception modules together



[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

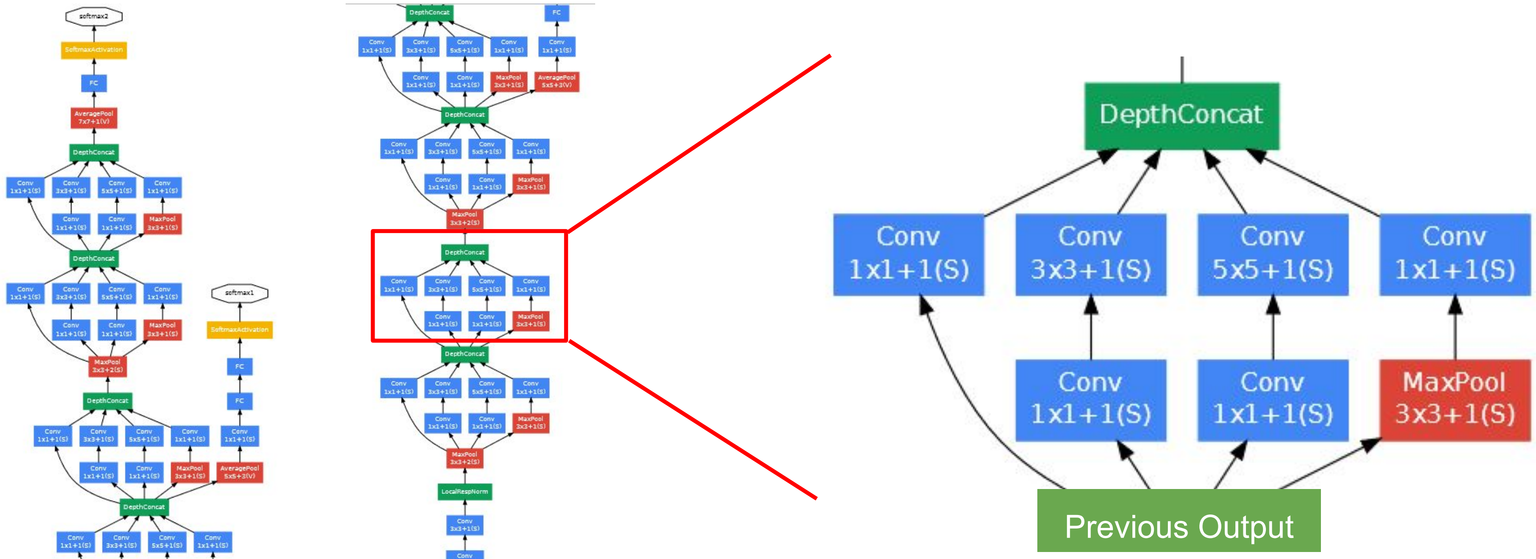
GoogLeNet Architecture

Key idea: stack inception modules together



[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

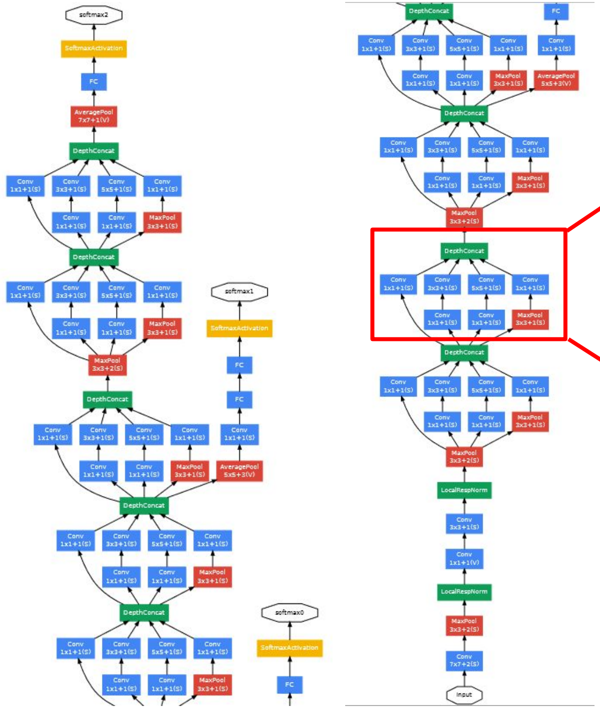
The Entire GoogLeNet Architecture

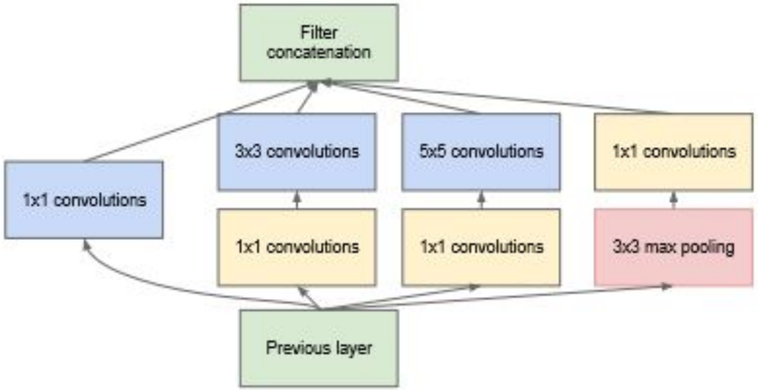


Inception Module

[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

The Entire GoogLeNet Architecture





Inception Module

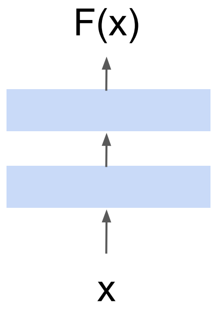
[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

Cornell Bowers C·IS

CNN Architectures

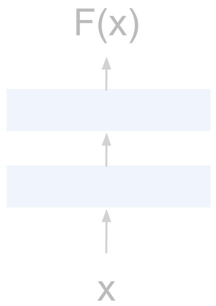
“Plain” CNN

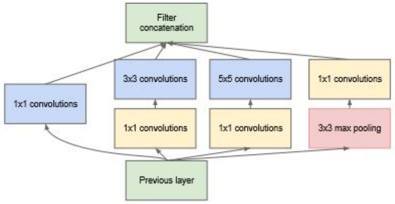
Simple connection from previous to next layer



CNN Architectures

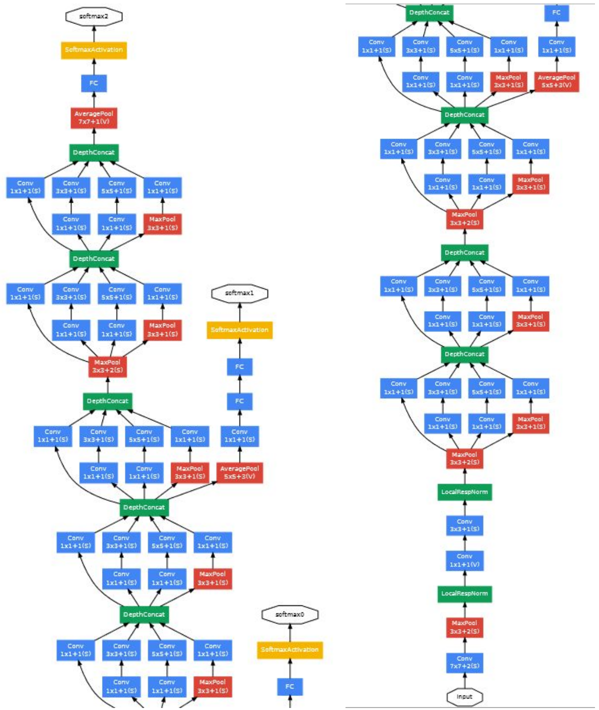
"Plain" CNN	**GoogLeNet**
Simple connection from previous to next layer	1x1, 3x3, 5x5 convolutions and pooling between each layer



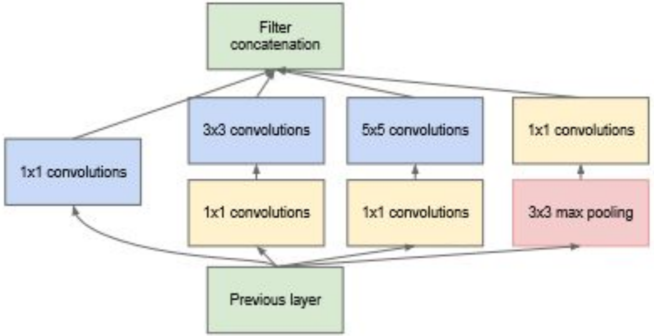


Cornell Bowers C·IS

The Entire GoogleNet Architecture

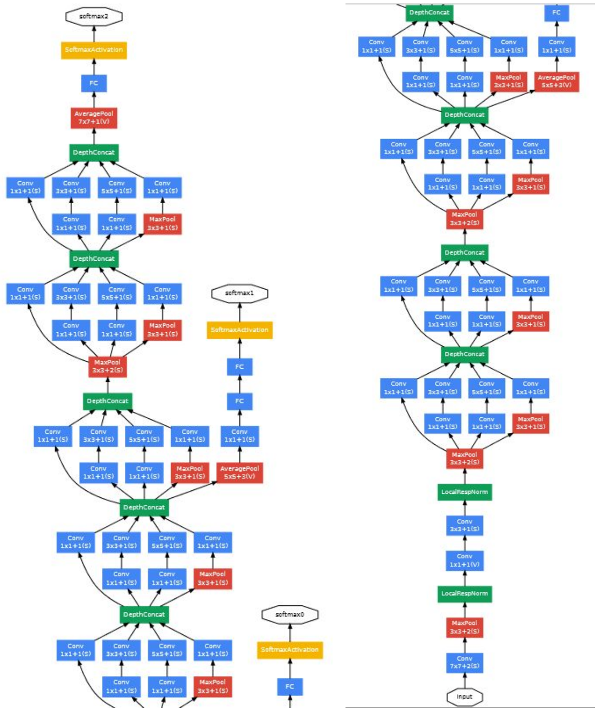


Very complicated - how exactly did this architecture solve the problem?

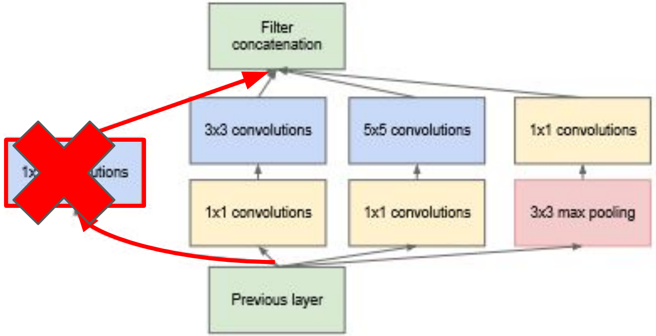


[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

The Entire GoogleNet Architecture



Very complicated - how exactly did this architecture solve the problem?

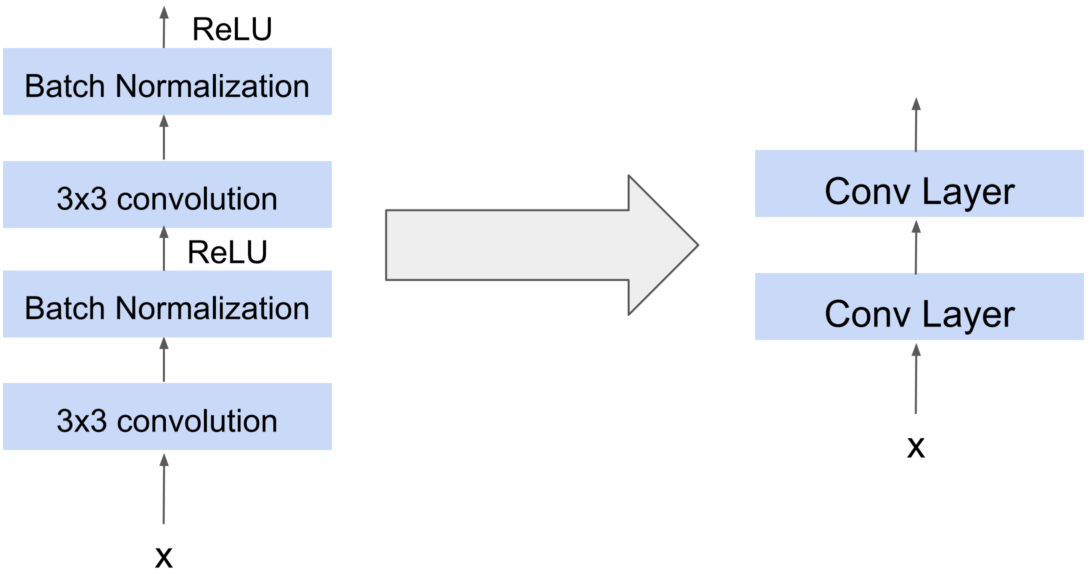


Residual connections: connect layers directly

[Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.]

Cornell Bowers CIS

Aside: Conv Layer Abstraction



Cornell Bowers C·IS

Backpropagation

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{[3]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[3]}} \frac{\partial \mathbf{a}^{[3]}}{\partial \mathbf{W}^{[3]}} = \delta^{[3]} (\mathbf{z}^{[2]})^T$$

Algorithm Backward Pass through MLP (Detailed)

1: **Input:** $\{\mathbf{z}^{[1]}, \ldots, \mathbf{z}^{[L]}\}$, $\{\mathbf{a}^{[1]}, \ldots, \mathbf{a}^{[L]}\}$, loss gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[L]}}$

2: $\delta^{[L]} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[L]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[L]}} \frac{\partial \mathbf{z}^{[L]}}{\partial \mathbf{a}^{[L]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[L]}} \odot \sigma^{[L]\prime}(\mathbf{a}^{[L]})$ ▷ Error term

3: **for** $l = L$ **to** 1 **do**

4: $\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{[l]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[l]}} \frac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{W}^{[l]}} = \delta^{[l]} (\mathbf{z}^{[l-1]})^T$ ▷ Gradient of weights

5: $\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{[l]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[l]}} \frac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{b}^{[l]}} = \delta^{[l]}$ ▷ Gradient of biases

6: $\frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[l-1]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[l]}} \frac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{z}^{[l-1]}} = (\mathbf{W}^{[l]})^T \delta^{[l]}$

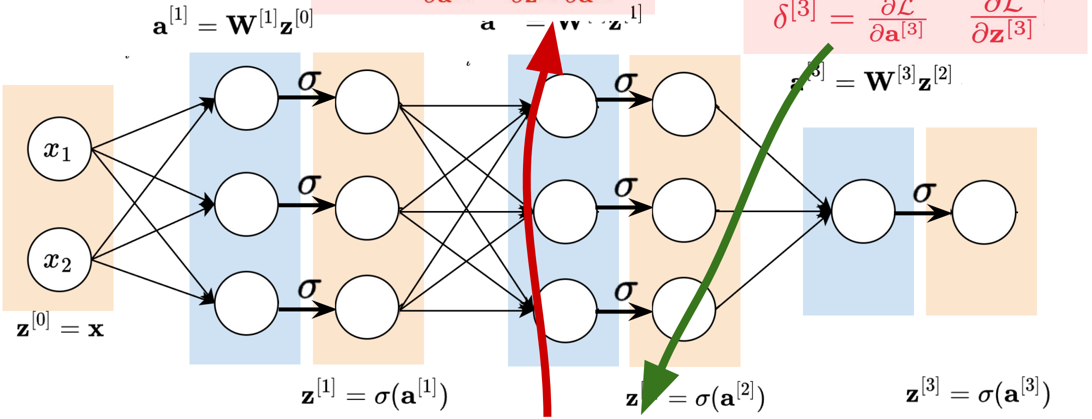
7: $\delta^{[l-1]} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[l-1]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[l-1]}} \frac{\partial \mathbf{z}^{[l-1]}}{\partial \mathbf{a}^{[l-1]}} = ((\mathbf{W}^{[l]})^T \delta^{[l]}) \odot \sigma^{[l-1]\prime}(\mathbf{a}^{[l-1]})$

8: **end for**

9: **Output:** $\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{[1:L]}}, \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{[1:L]}}$

$$\delta^{[2]} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[2]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[2]}} \frac{\partial \mathbf{z}^{[2]}}{\partial \mathbf{a}^{[2]}}$$

$$\delta^{[3]} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[3]}} \quad \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[3]}}$$



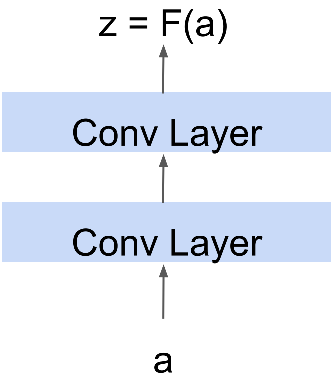
$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[2]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{[3]}} \frac{\partial \mathbf{a}^{[3]}}{\partial \mathbf{z}^{[2]}} = (W^{[3]})^T \delta^{[3]}$$

$$\mathcal{L}(\mathbf{z}^{[3]}, \mathbf{y})$$

We can directly compute $\frac{\partial \mathcal{L}}{\partial \mathbf{z}^{[3]}}$!

Backpropagation through “plain” conv layers

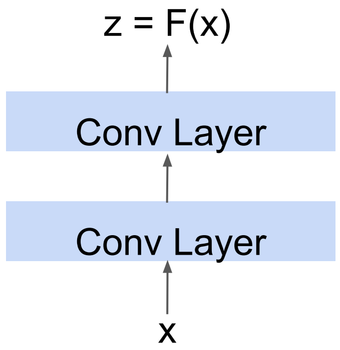
“Plain” layers



$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial a} = \frac{\partial L}{\partial z}\boxed{(F'(a))}$$

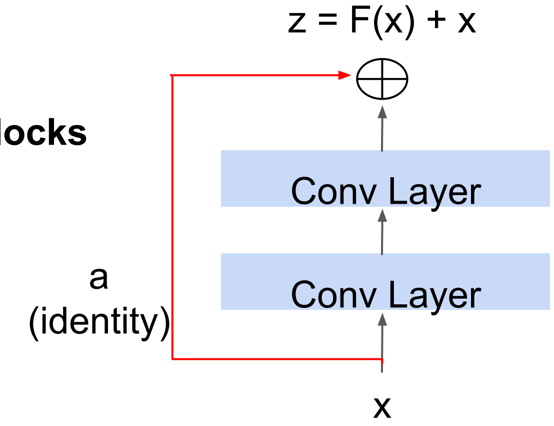
Backpropagation through Residual blocks

“Plain” layers



$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x} = \frac{\partial L}{\partial z}\boxed{F'(x)}$$

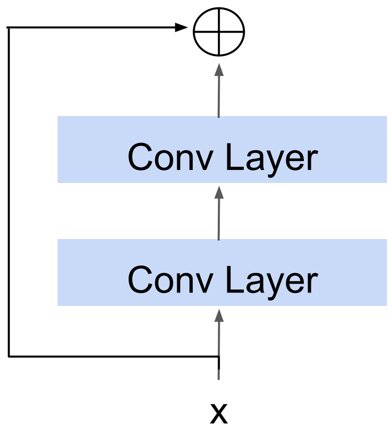
Residual Blocks



$$\frac{\partial L}{\partial x} =$$

ResNet

Stack residual blocks together!

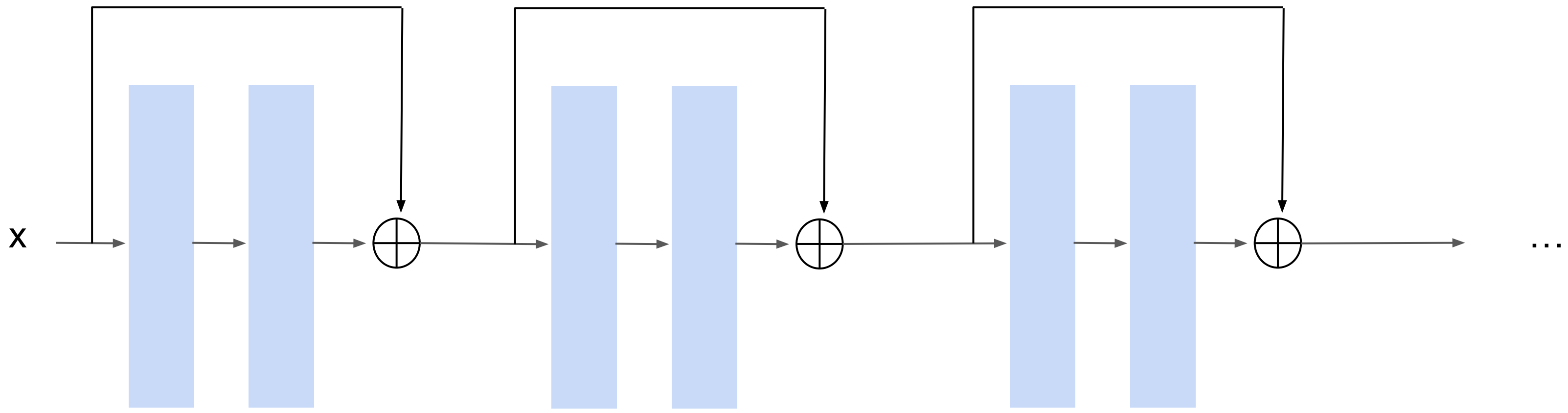


[He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.]

Cornell Bowers C·IS

ResNet

Stack residual blocks together!

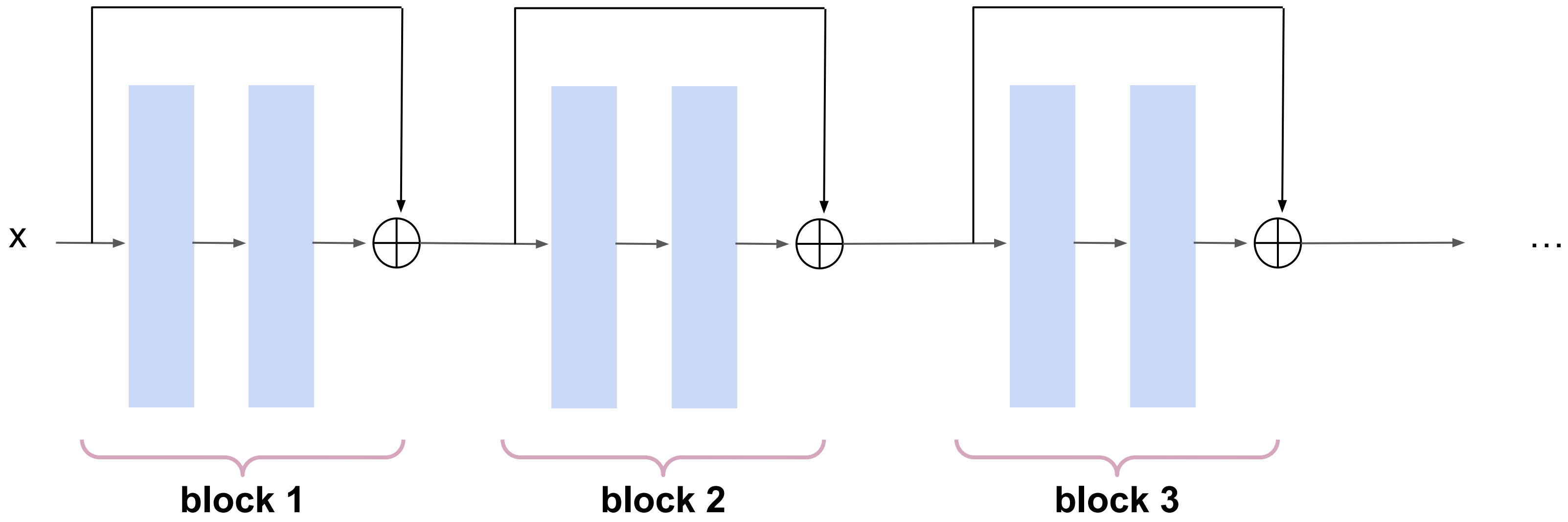


[He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.]

Cornell Bowers C·IS

ResNet

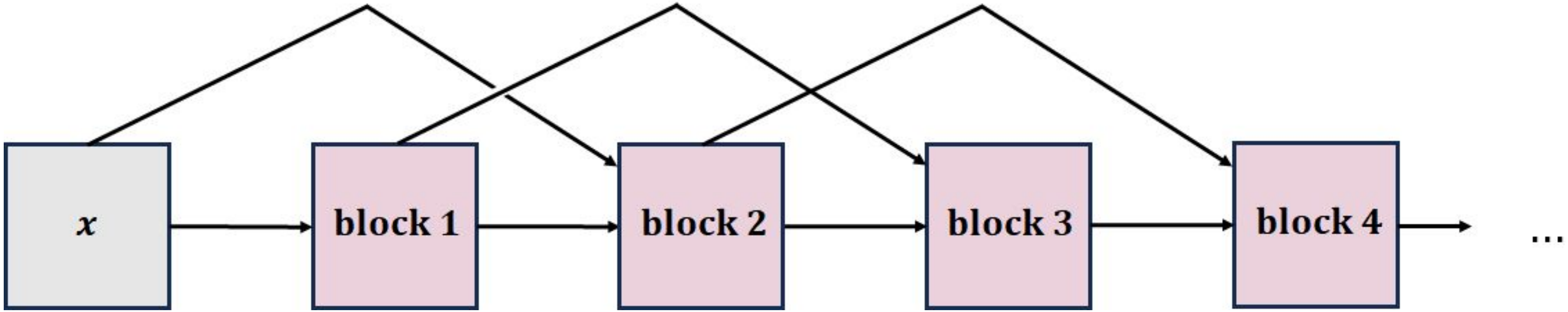
Stack residual blocks together!



[He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.]

ResNet

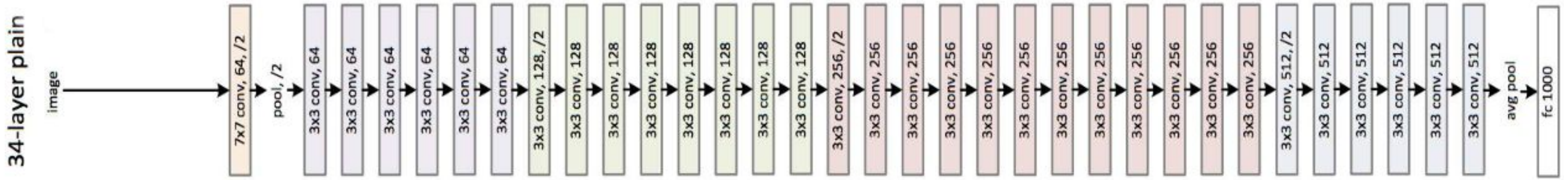
Stack residual blocks together!



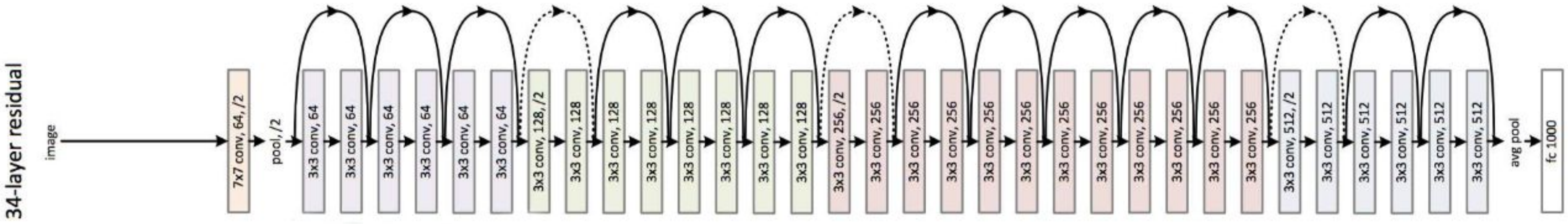
[He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.]

Full ResNet Architecture

“Plain” Network

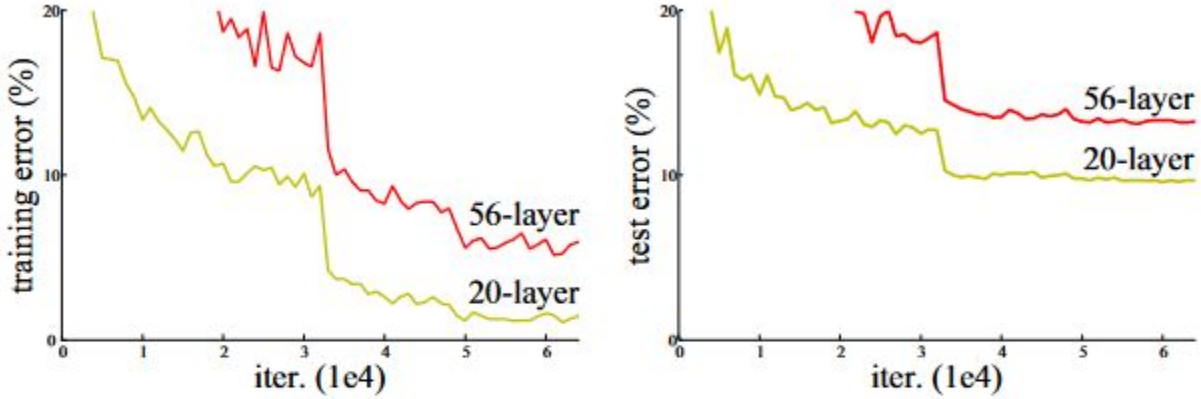


ResNet



[He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.]

Recall: How can a larger network achieve a higher training error?



56 layer CNN has higher training and test error than 20 layer CNN on CIFAR-10 dataset for image classification

Cornell Bowers C·IS

Deeper == better

Can train deeper models!

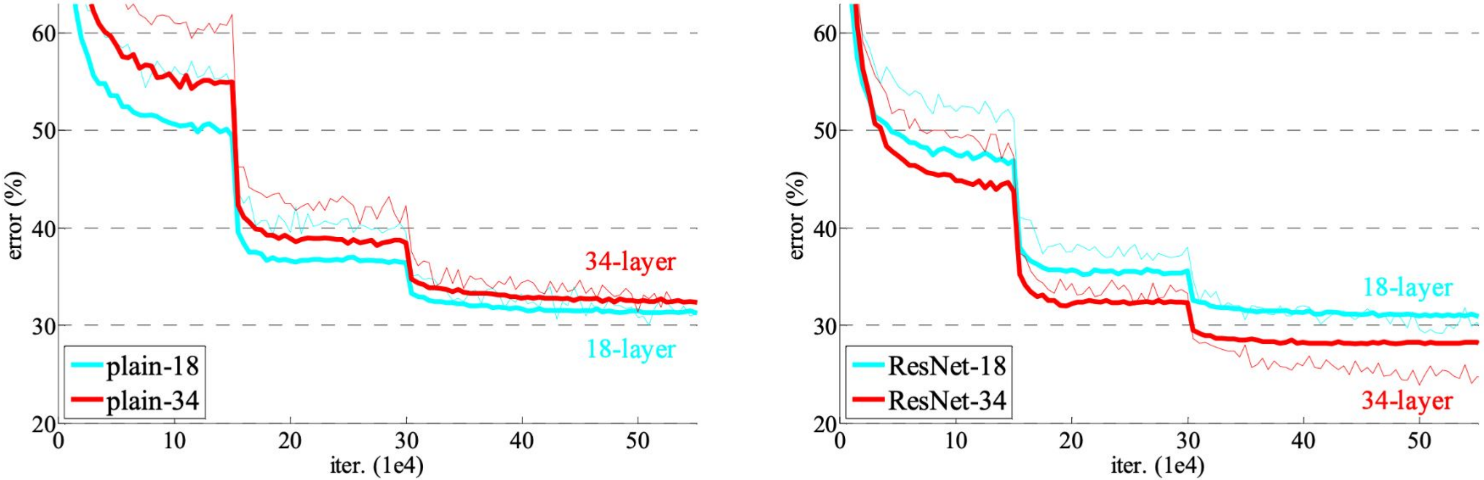
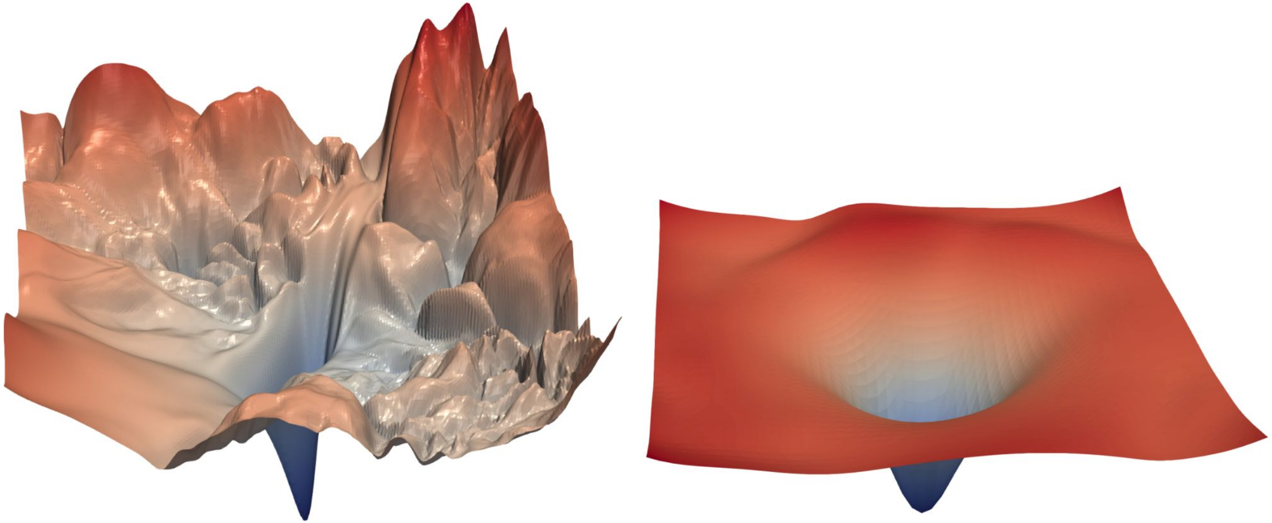


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

Cornell Bowers C·IS

Visualizing the Effect of Skip Connections

Makes optimization easier!



(a) without skip connections

(b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

[Li, Hao, et al. "Visualizing the loss landscape of neural nets." Advances in neural information processing systems 31 (2018).]

Cornell Bowers C·IS

Stochastic Depth

Still have long training times! Solution: stochastic depth

[Huang, Gao, et al. "Deep networks with stochastic depth." *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14.* Springer International Publishing, 2016.]

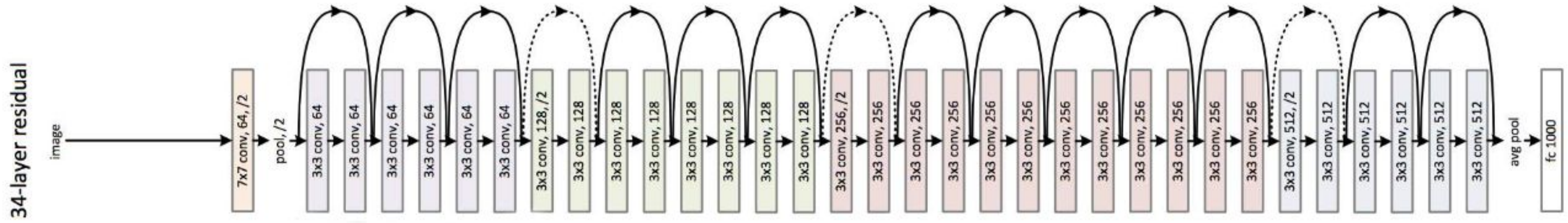
Stochastic Depth

During training, randomly drop Residual Blocks using skip connections

Like dropout but with residual blocks instead of individual neurons

Another benefit: robustness/mitigating overfitting

Drop probability for layer l (out of L): $$p_l = 1 - \frac{l}{L}(1 - p_L)$$



[Huang, Gao, et al. "Deep networks with stochastic depth." *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14.* Springer International Publishing, 2016.]

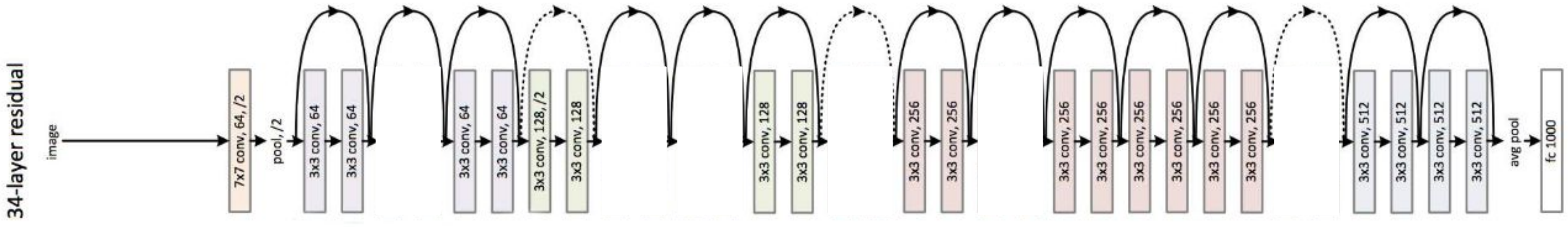
Stochastic Depth

During training, randomly drop Residual Blocks using skip connections

Like dropout but with residual blocks instead of individual neurons

Another benefit: robustness/mitigating overfitting

Drop probability for layer l (out of L): $$p_l = 1 - \frac{l}{L}(1 - p_L)$$



[Huang, Gao, et al. "Deep networks with stochastic depth." *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14.* Springer International Publishing, 2016.]

Cornell Bowers C·IS

Stochastic Depth

Increases training loss, but… decreases te

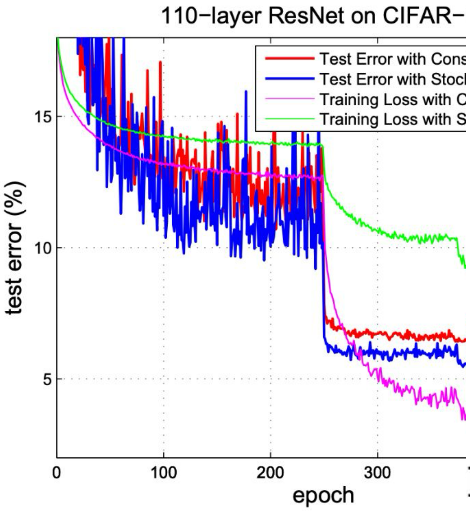


Fig. 3. Test error on CIFAR-1
data augmentation, correspond

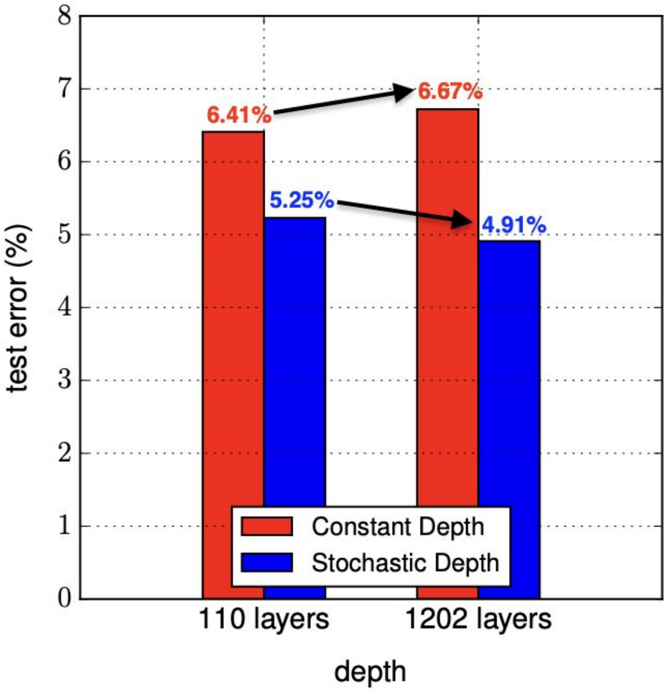


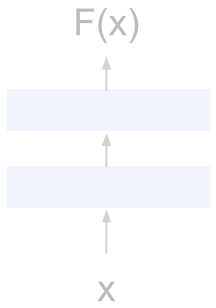
Fig. 5. With stochastic depth, the 1202-layer ResNet still significantly improves over the 110-layer one.

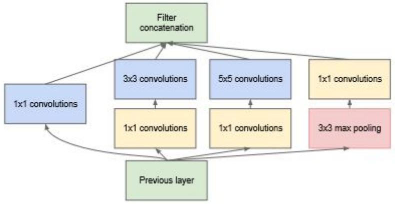
raining loss

[Huang, Gao, et al. "Deep networks with stochastic depth." *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14.* Springer International Publishing, 2016.]

CNN Architectures

“Plain” CNN	**GoogLeNet**
Simple connection from previous to next layer	1x1, 3x3, 5x5 convolutions and pooling between each layer

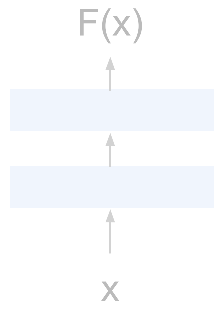


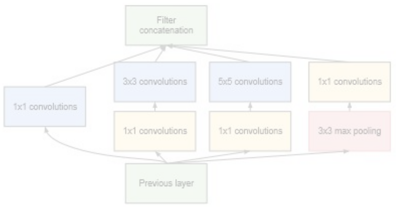


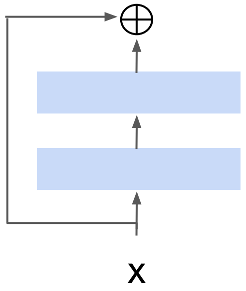
Cornell Bowers C·IS

CNN Architectures

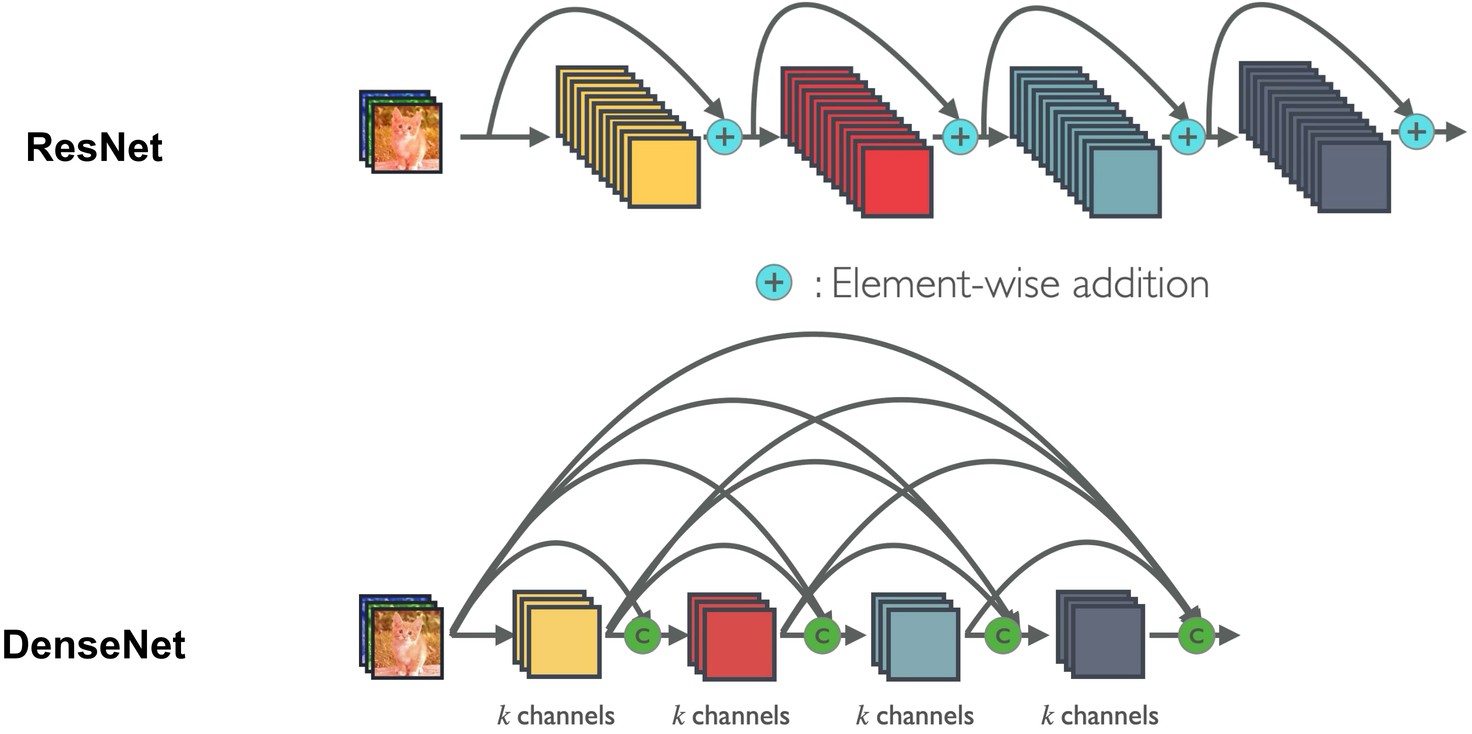
"Plain" CNN	GoogLeNet	**ResNet**
Simple connection from previous to next layer	1x1, 3x3, 5x5 convolutions and pooling between each layer	Skip connections Add output of previous layer to next layer







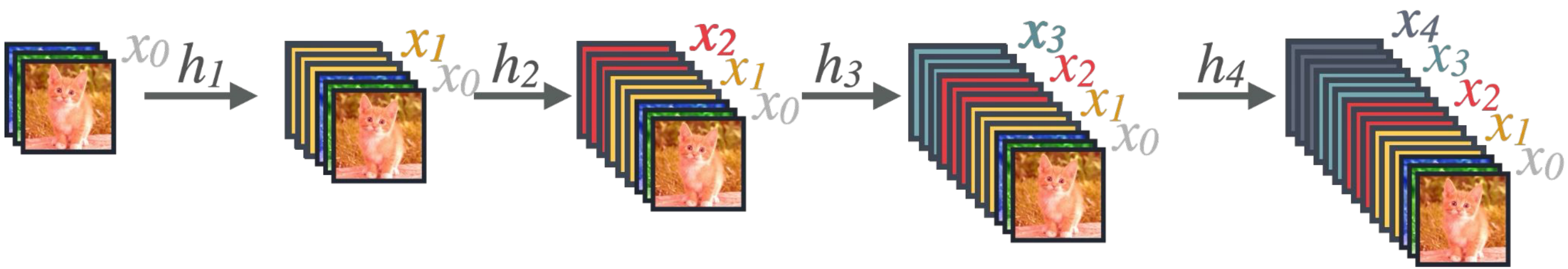
From ResNets to DenseNets



[Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.]

DenseNets

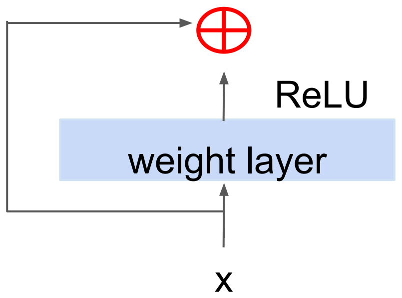
Feature concatenation



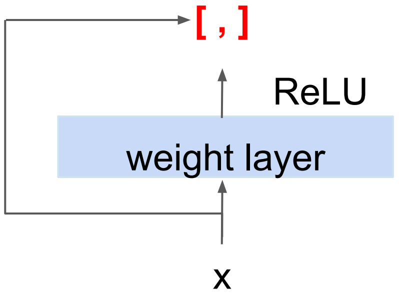
Cornell Bowers C·IS

Dense Blocks

To create dense connections, dense blocks use the same structure as residual blocks, but concatenate (denoted by [,]) inputs instead of simply adding them



Residual Blocks



Dense Blocks

[Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.]

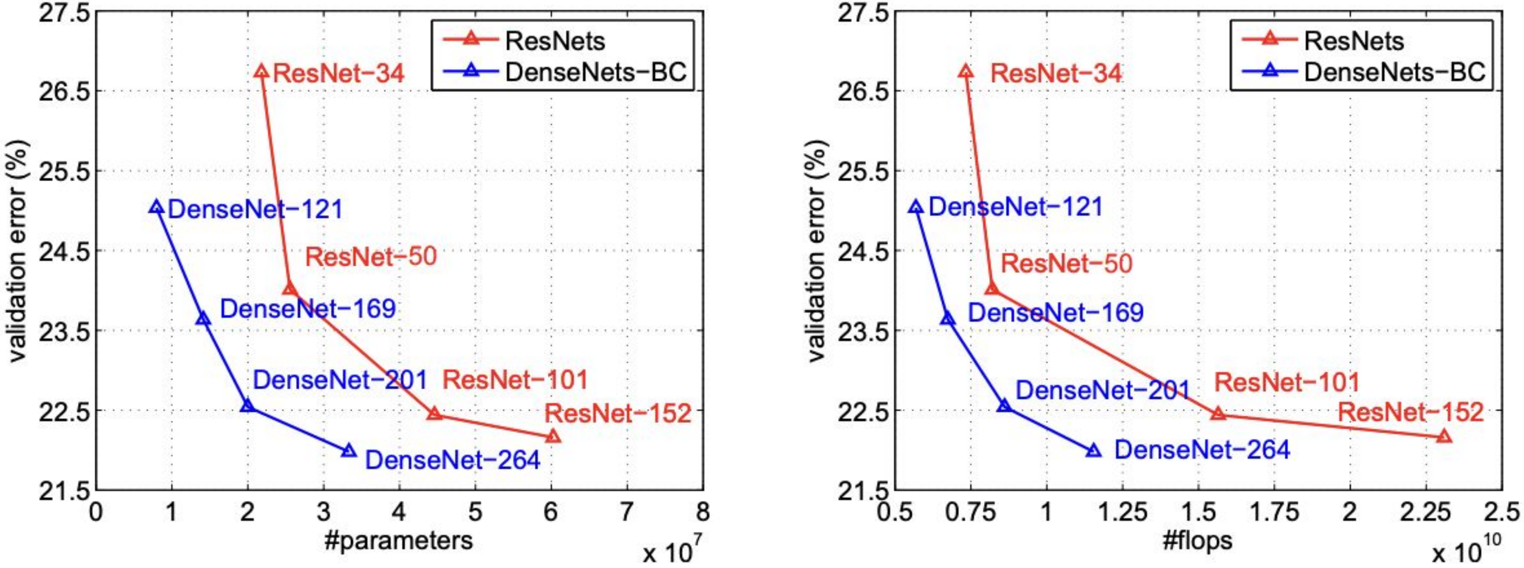
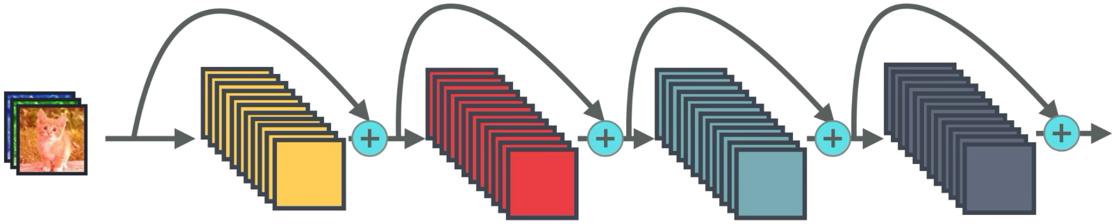


Figure 3: Comparison of the DenseNets and ResNets top-1 error rates (single-crop testing) on the ImageNet validation dataset as a function of learned parameters (*left*) and FLOPs during test-time (*right*).

Cornell Bowers C·IS

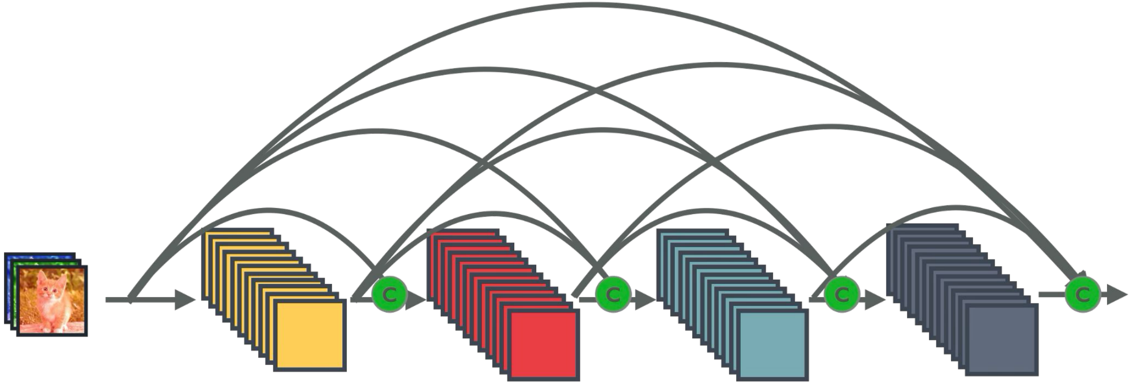
Discussion: What design choices might allow a ~100-layer DenseNet to have fewer parameters than a ~100-layer ResNet?

ResNet



: Element-wise addition

DenseNet

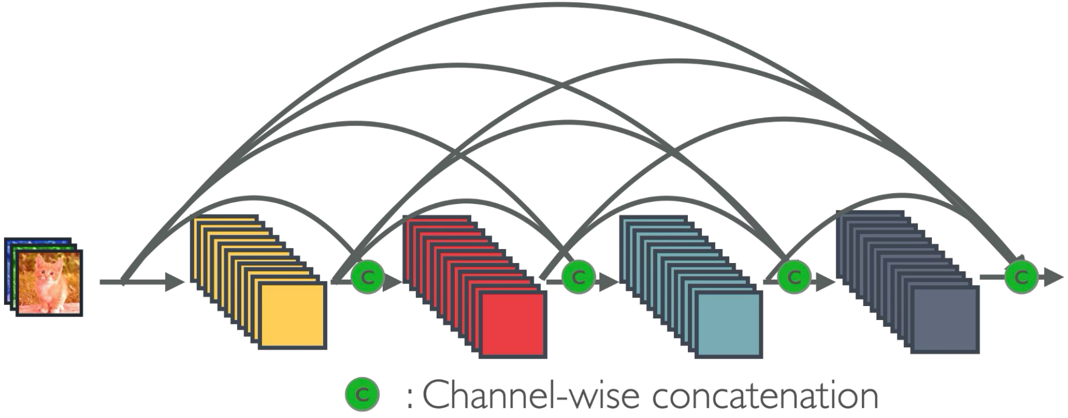


: Channel-wise concatenation

Dense Connections

Each layer has access to every other layer before it, which:

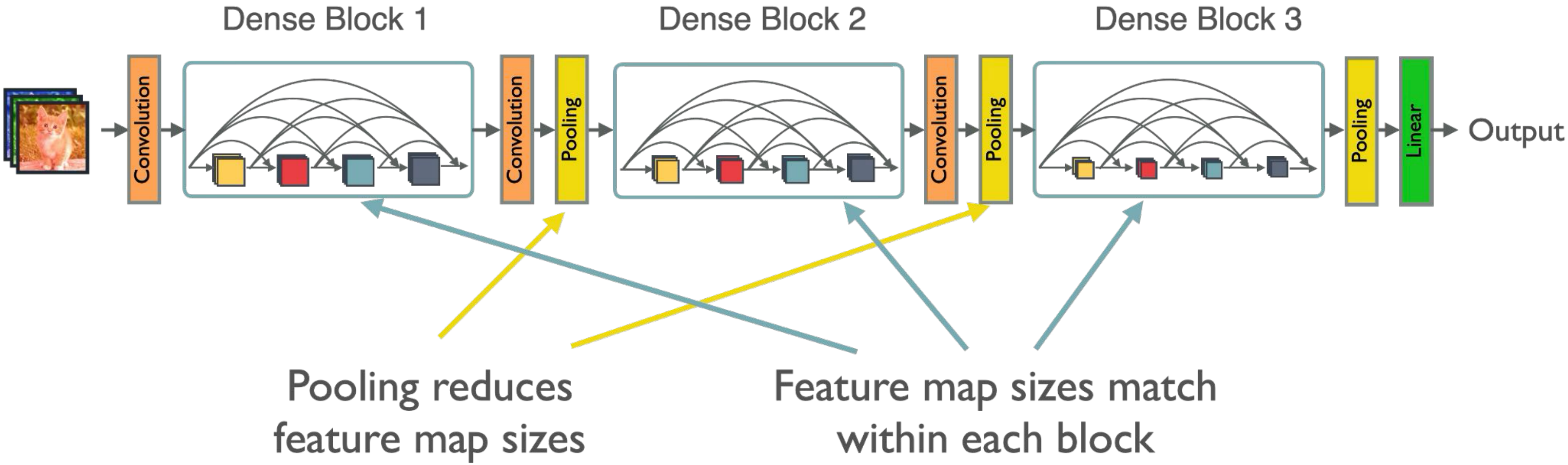
- maximizes information flow
- allows for feature-map reuse
- less parameters to learn
- alleviates vanishing gradient



[Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.]

Cornell Bowers C·IS

DenseNets

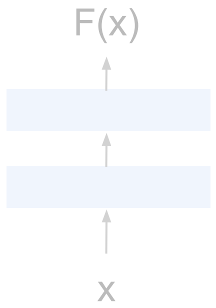


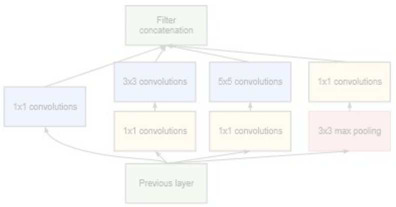
[Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.]

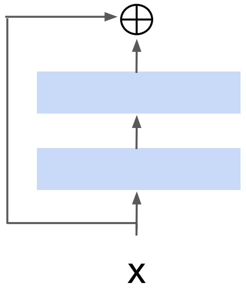
Cornell Bowers C·IS

CNN Architectures

"Plain" CNN	GoogLeNet	ResNet
Simple connection from previous to next layer	1x1, 3x3, 5x5 convolutions and pooling between each layer	Skip connections Add output of previous layer to next layer





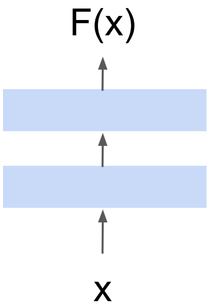


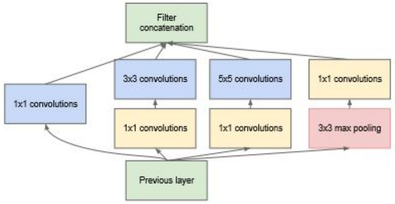
CNN Architectures

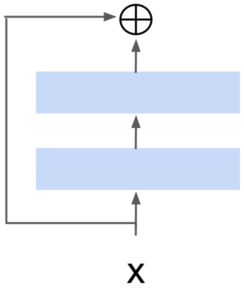
"Plain" CNN	GoogLeNet	ResNet	**DenseNet**
Simple connection from previous to next layer	1x1, 3x3, 5x5 convolutions and pooling between each layer	Skip connections Add output of previous layer to next layer	Dense connections Concatenate output of previous layer to next layer

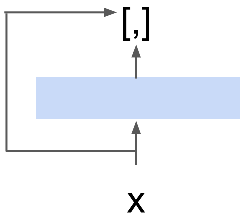
Summary of Models

"Plain" CNN	Google Net	ResNet	DenseNet
Simple connection from previous to next layer	1x1, 3x3, 5x5 convolutions and pooling between each layer	Skip connections Add output of previous layer to next layer	Dense connections Concatenate output of previous layer to next layer









Cornell Bowers C·IS

Summary

- Deep CNNs outperform shallow CNNs
- But…
  - Harder optimization problem!
- Residual (and dense) connections make training easier!
  - Can train networks with 100s of layers!
- Stochastic depth let’s you train deeper networks faster
  - 1000+ layers!
- In general…
  - Build large networks as stacks of (many!) simple building blocks