MVA Microsoft Virtual Academy

微软 DevX AI 系列课程

Convolutional Neural Networks

Microsoft

Jingdong Wang | Senior Researcher
jingdw@microsoft.com

Outline

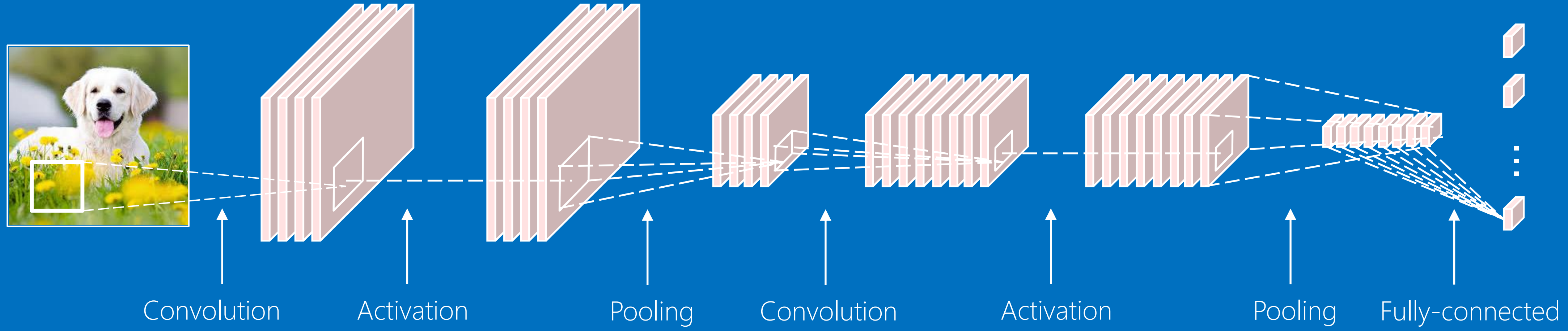
Convolutional Neural Networks

Microsoft Virtual Academy

01. Overview

Microsoft

An Example CNN Architecture



MVA Microsoft Virtual Academy

02. Layers

Microsoft

Convolution Layer – Convolution Operation

- Continuous domain

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau$$
$$= \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

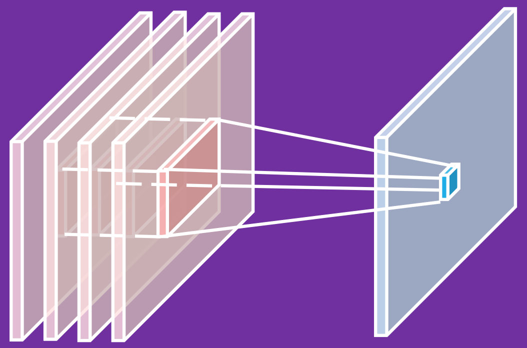
- Discrete domain

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[n-m]g[m]$$
$$= \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

- Cross-correlation, implemented in many DNN libraries

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[n+m]g[m]$$

$$(I * K)[x,y] = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I[(x+i, y+j]K[i,j]$$



An example

Convolution Layer – Convolution Operation

- Continuous domain

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

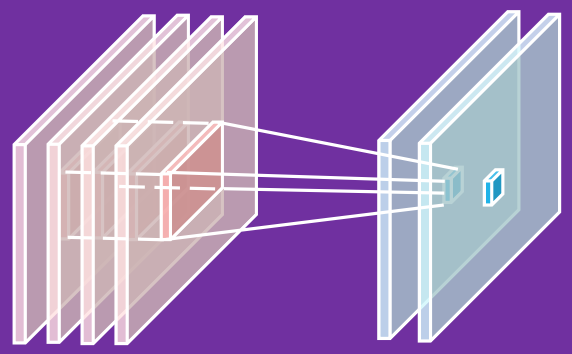
- Discrete domain

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[n-m]g[m] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

- Cross-correlation, implemented in many DNN libraries

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[n+m]g[m]$$

$$(I * K)[x, y] = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I[(x+i, y+j]K[i, j]$$



An example

Convolution Layer – Convolution Operation

- Continuous domain

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau$$
$$= \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

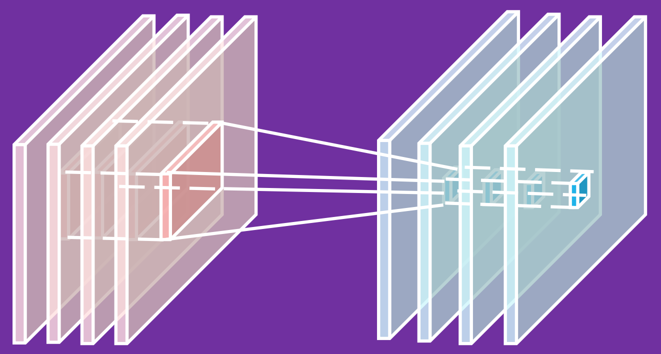
- Discrete domain

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[n-m]g[m]$$
$$= \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

- Cross-correlation, implemented in many DNN libraries

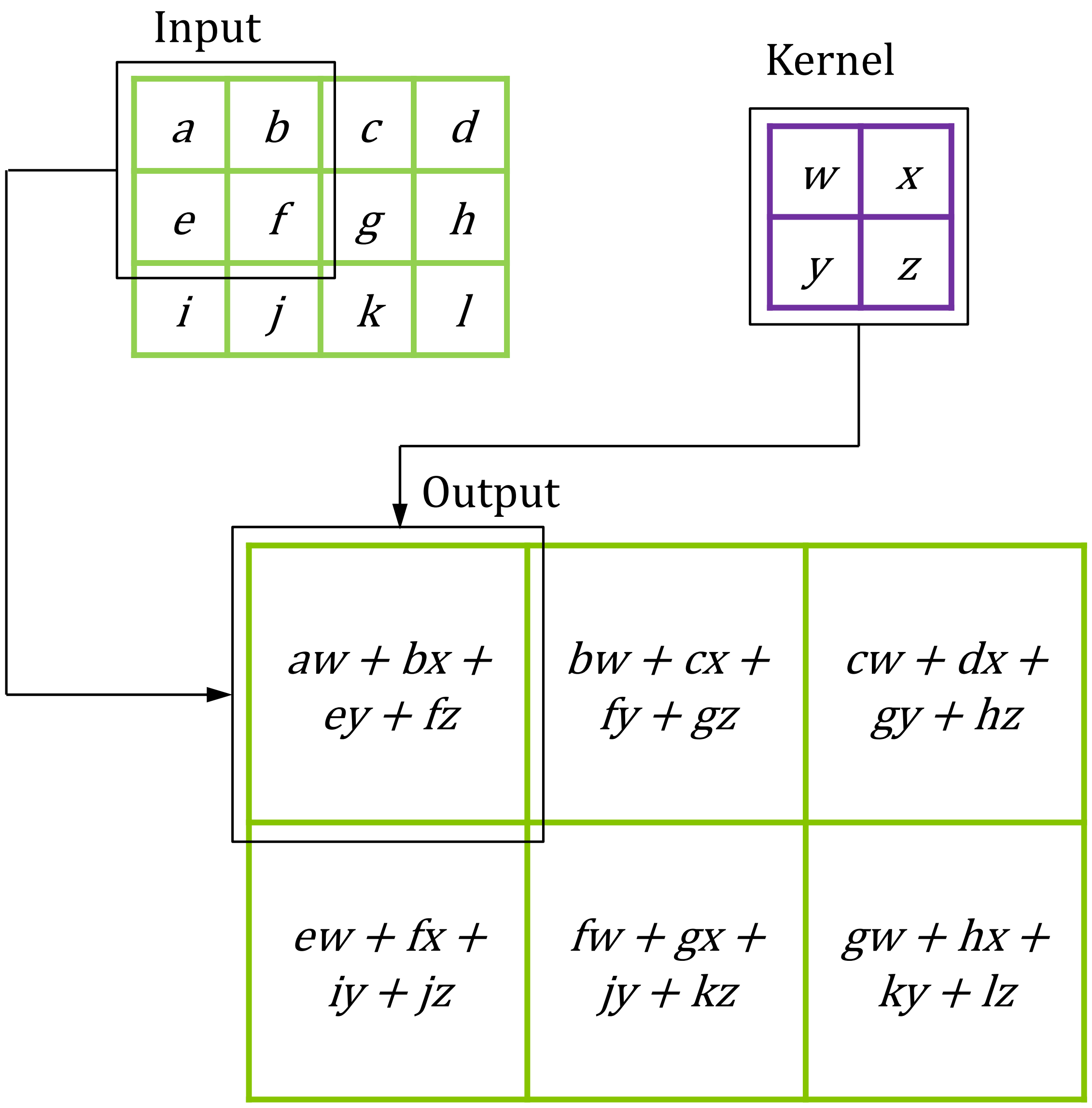
$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[n+m]g[m]$$

$$(I * K)[x, y] = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I[(x+i, y+j]K[i, j]$$

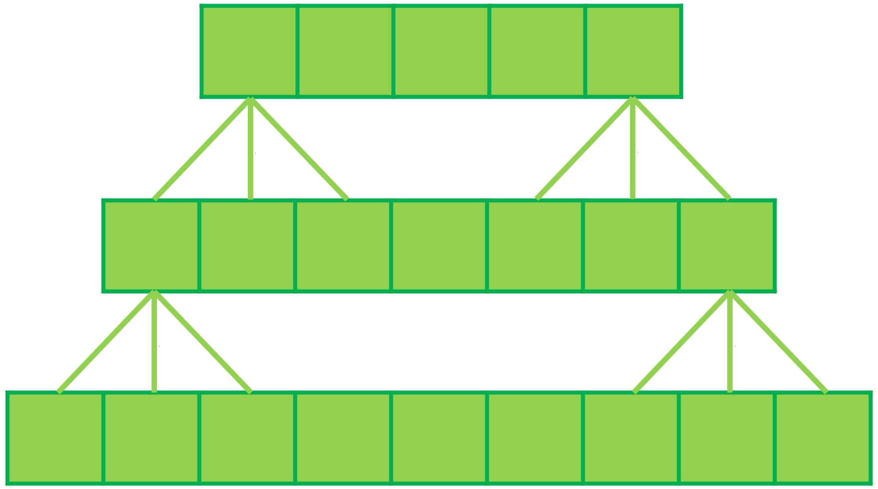


An example

Convolution Layer – An Example of Convolution Operation

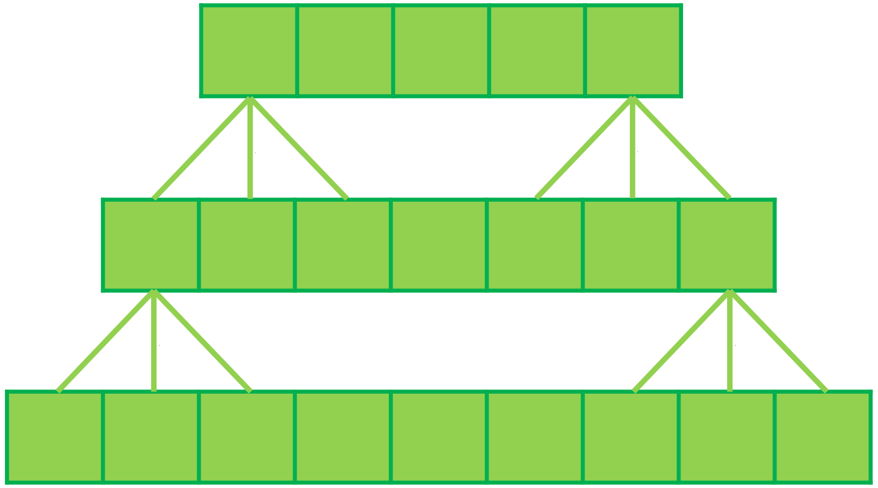


Convolution Layer – Zero Padding



w/o zero padding

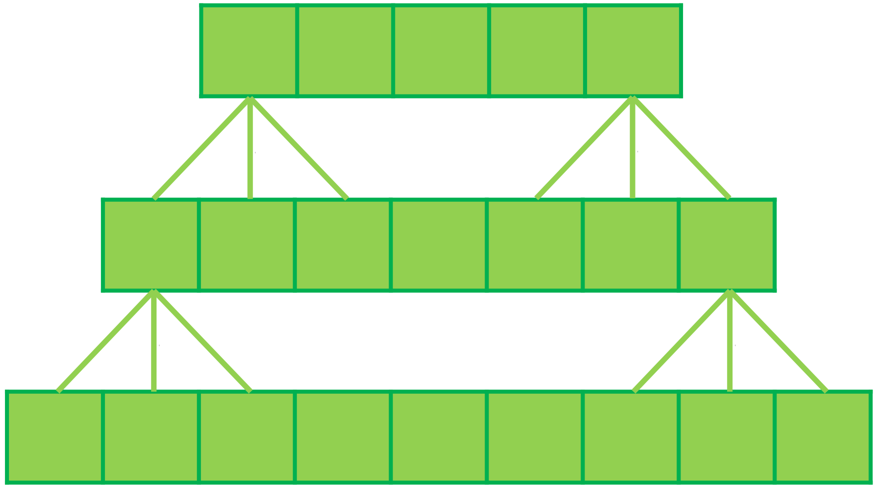
Convolution Layer – Zero Padding



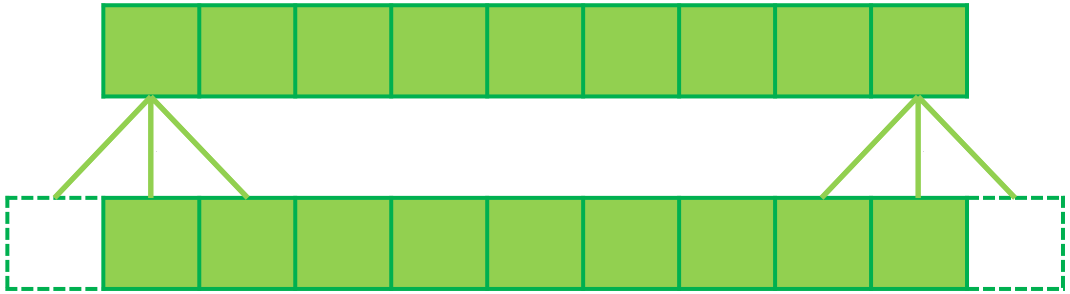
w/o zero padding

w/ zero padding

Convolution Layer – Zero Padding

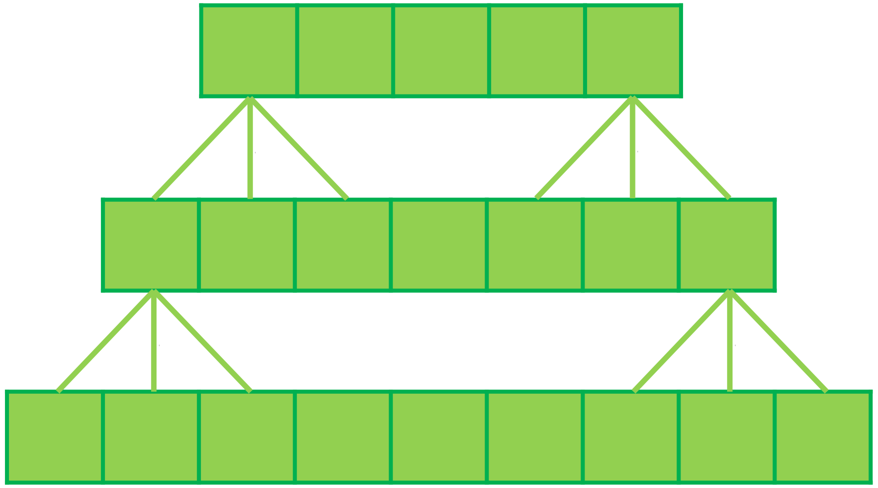


w/o zero padding

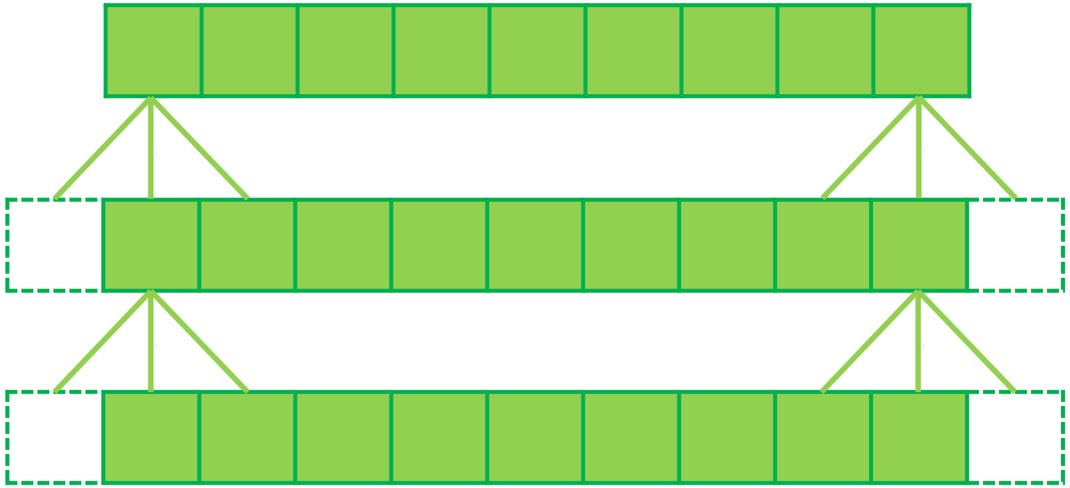


w/ zero padding

Convolution Layer – Zero Padding

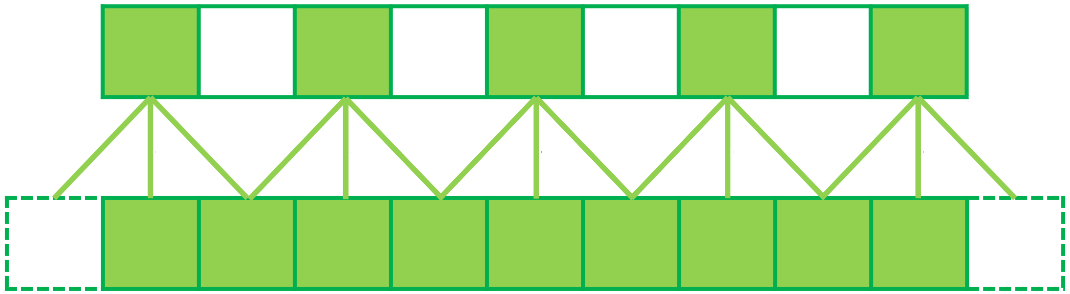


w/o zero padding

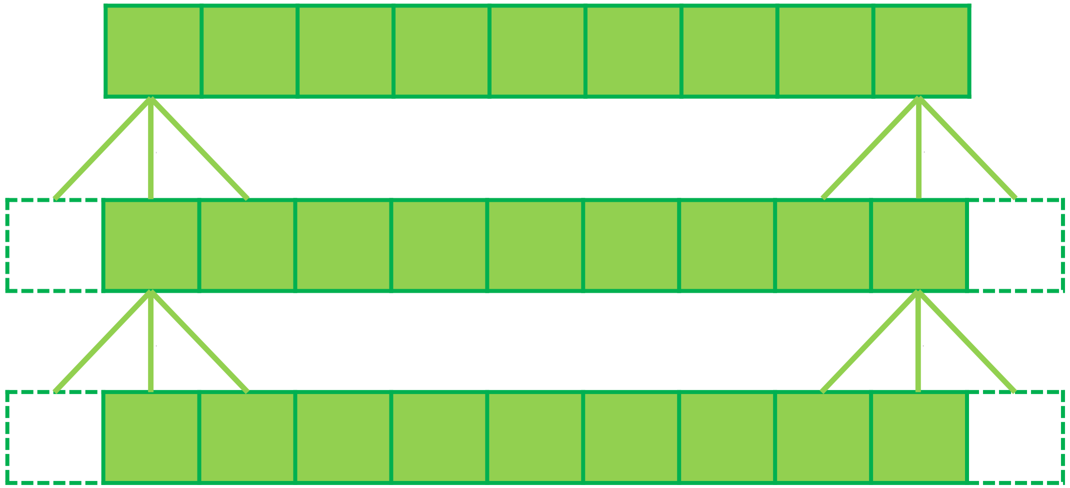


w/ zero padding

Convolution Layer – Striding

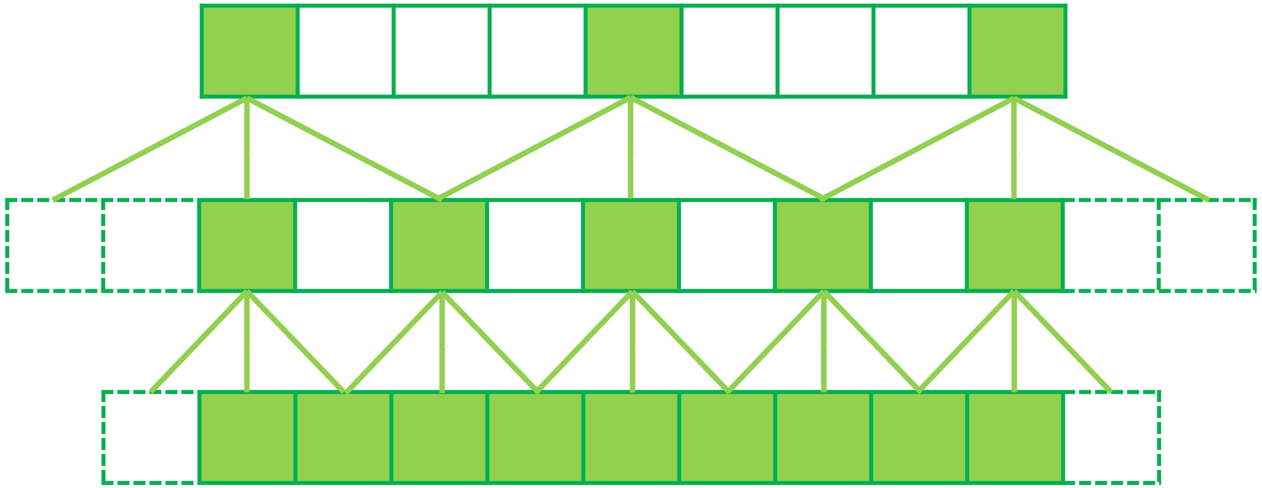


stride = 2

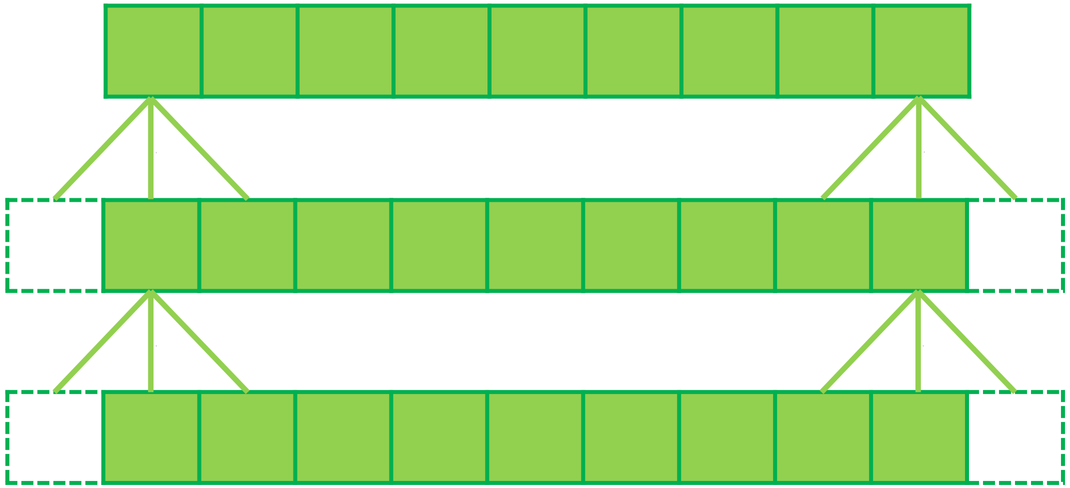


stride = 1

Convolution Layer – Striding



stride = 2



stride = 1

Activation Layer

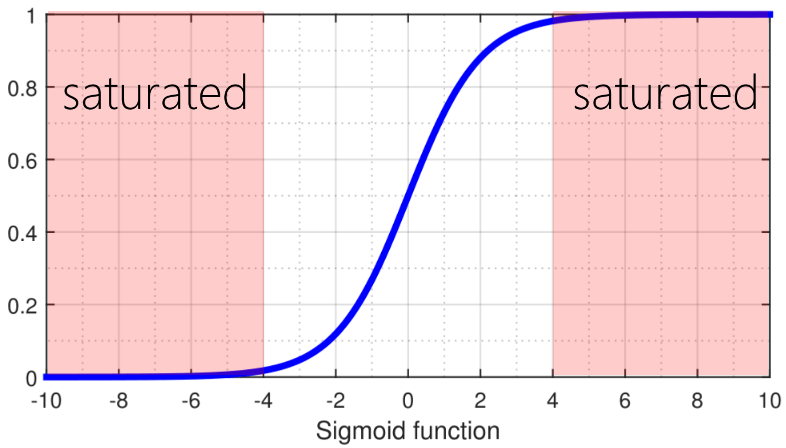
- Sigmoid

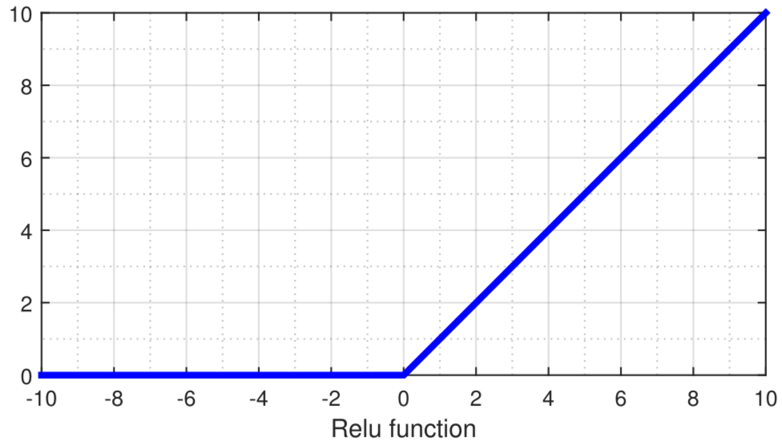
$$\boxed{S(x) = \frac{1}{1 + e^{-x}}}$$

- Rectified Linear Units (ReLU) √

$$\boxed{\mathrm{ReLU}(x) = \max(0, x)}$$

- Others
  - Tanh
  - Leaky ReLU (Parametric ReLU)
  - ELU
  - …





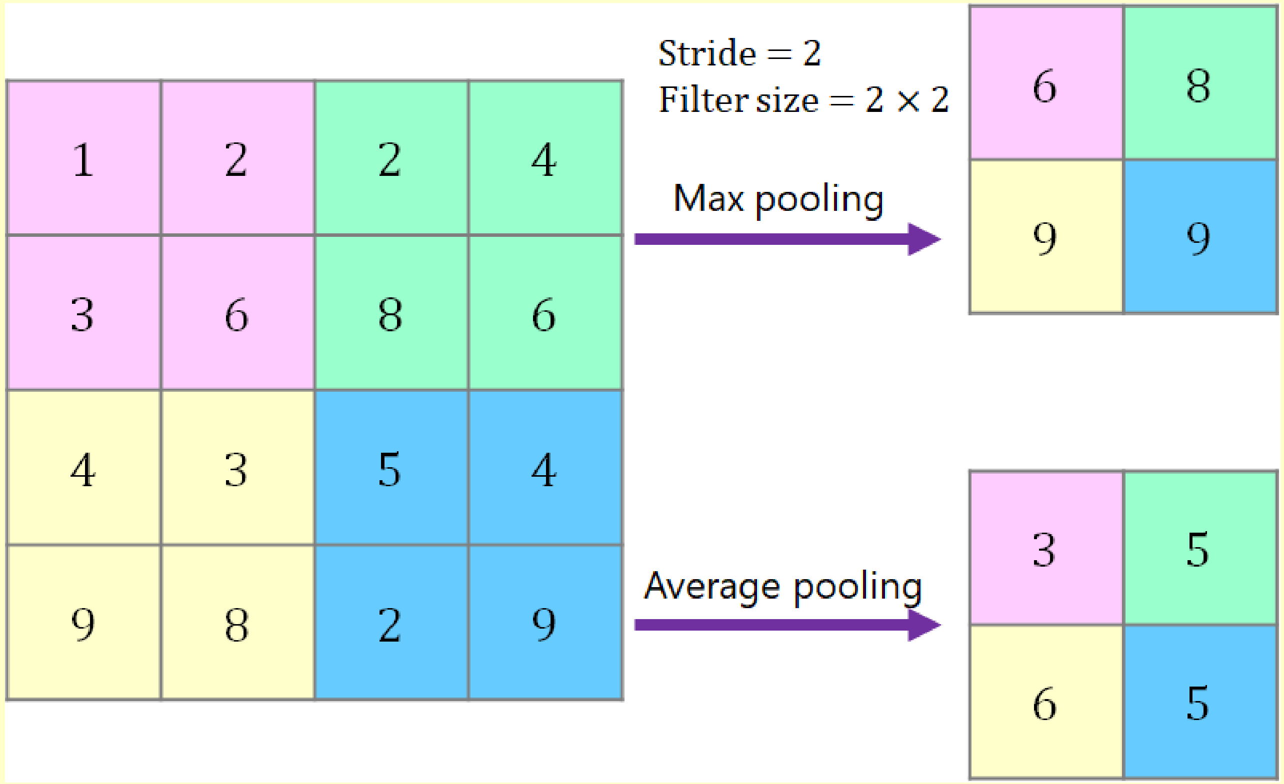
Pooling Layer

- Max pooling

$$\begin{aligned}&\text{max-pooling}(f[i-1], f[i], f[i+1])\\=\;&\max(f[i-1], f[i], f[i+1])\end{aligned}$$

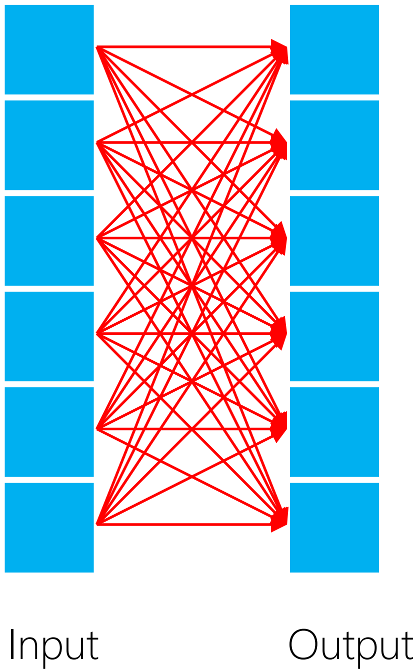
- Average pooling

$$\begin{aligned}&\text{ave-pooling}(f[i-1], f[i], f[i+1])\\=\;&\frac{1}{3}(f[i-1]+f[i]+f[i+1])\end{aligned}$$



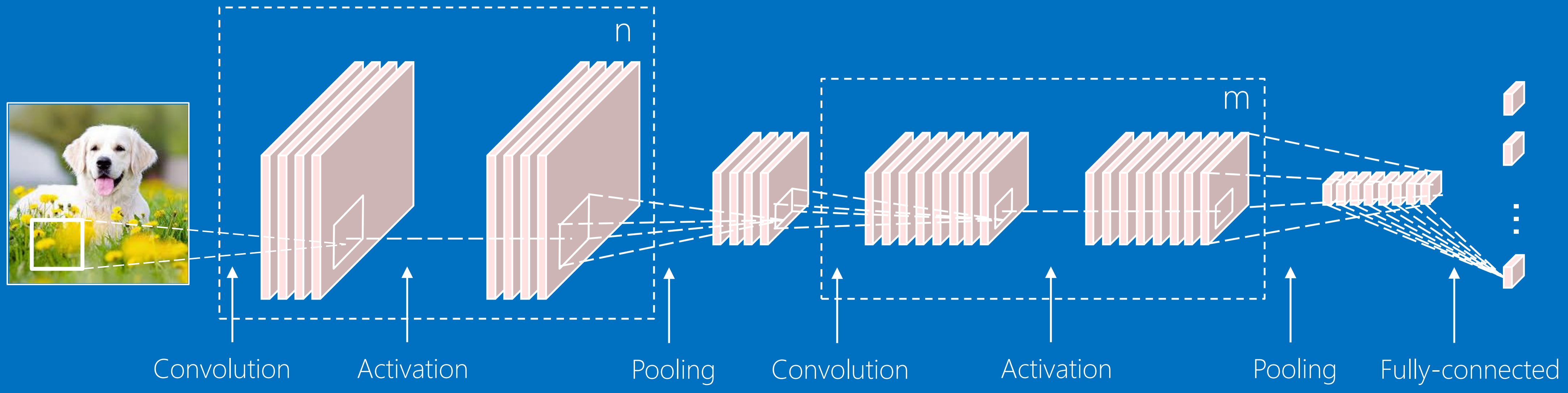
Fully-Connected Layer

- Each output neuron is connected to each input neuron

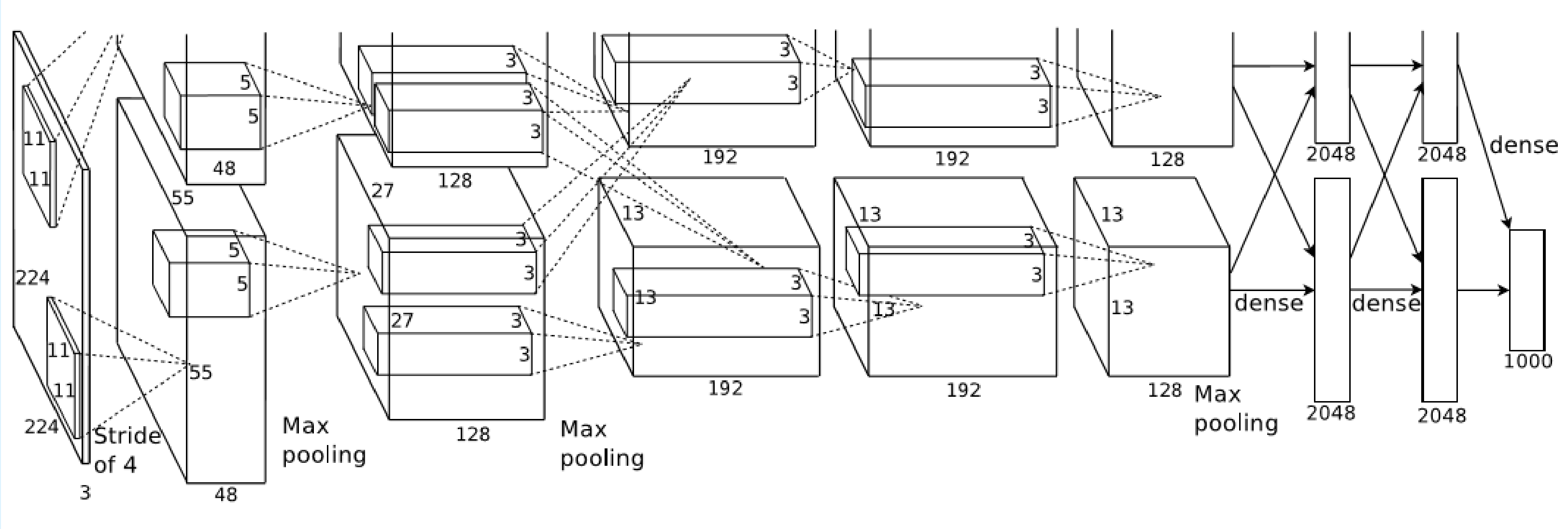


Deep Convolutional Neural Networks

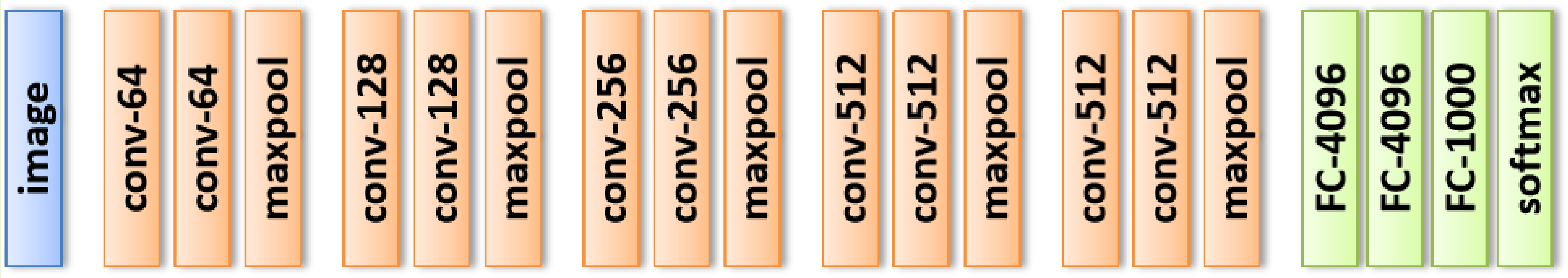
Many convolutional (+activation) layers ➡ Deep



Examples of CNNs



AlexNet



VGGNet

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton: ImageNet Classification with Deep Convolutional Neural Networks. NIPS 2012: 1106-1114
Karen Simonyan, Andrew Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR abs/1409.1556 (2014)

Microsoft Virtual Academy

03. Learning

Microsoft

Loss Function

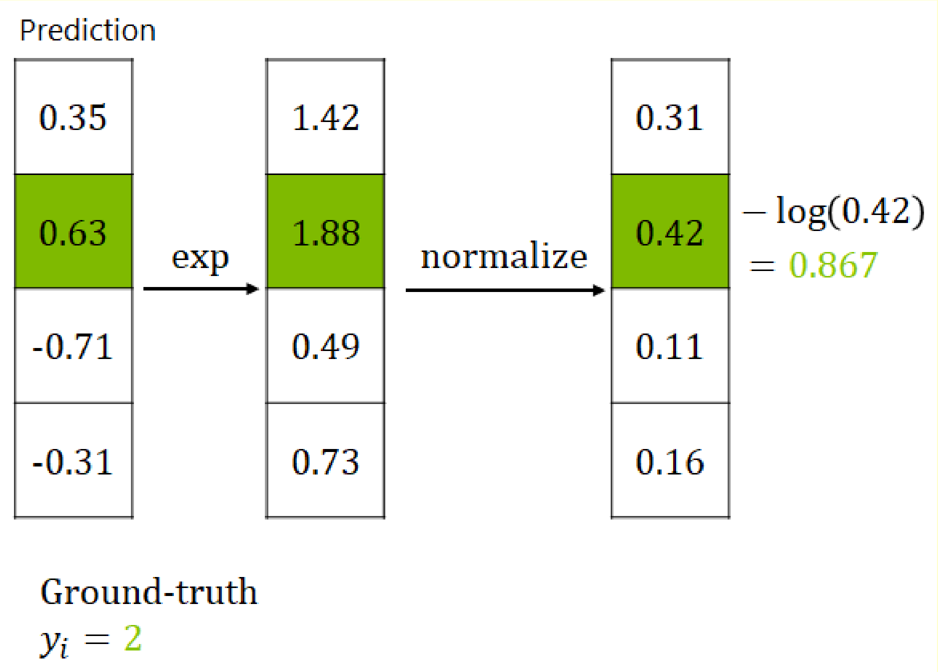
- Classification loss function
  - SoftMax loss
  - …

$$L(i) = -\log\left(\frac{e^{f_{y^i}}}{\sum_j e^{f_j}}\right)$$

- Regression loss function
  - Euclidean loss
  - …

$$L(i) = \sum_j (f_j^i - \bar{f}_j^i)^2$$

SoftMax loss



Optimization - SGD

Stochastic gradient descent (SGD)

In each iteration

Gradient of loss function

1. Sample a minibatch: $\{\mathbf{x}^1, \ldots, \mathbf{x}^i, \ldots, \mathbf{x}^m\}$

Weight decay

2. Compute the gradient over the mini-batch

$$\mathbf{g} \leftarrow \boxed{\frac{1}{m} \Delta_{\boldsymbol{\theta}} \sum_{i=1}^{m} L(f(\mathbf{x}^i; \boldsymbol{\theta}), y^i)} + \boxed{\lambda \boldsymbol{\theta}}$$

3. Update the velocity $\mathbf{v} \leftarrow \alpha \mathbf{v} - \epsilon \mathbf{g}$

Momentum

4. Update the parameters $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{v}$

Gradient Computation by Back Propagation

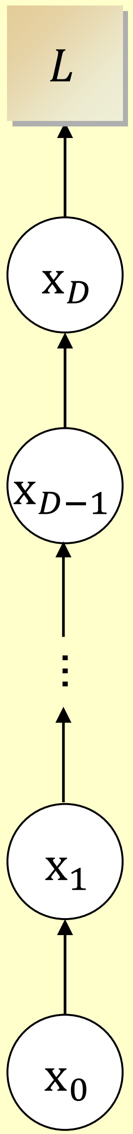
Chain rule of Calculus

- w.r.t hidden responses

$$\boxed{\frac{\partial L}{\partial \mathbf{x}_d} = \left(\frac{\partial \mathbf{x}_{d+1}}{\partial \mathbf{x}_d}\right)^\top \dots \left(\frac{\partial \mathbf{x}_D}{\partial \mathbf{x}_{D-1}}\right)^\top \frac{\partial L}{\partial \mathbf{x}_D}}$$

- w.r.t. model parameters

$$\boxed{\frac{\partial L}{\partial \boldsymbol{\theta}_d} = \left(\frac{\partial \mathbf{x}_d}{\partial \boldsymbol{\theta}_d}\right)^\top \frac{\partial L}{\partial \mathbf{x}_d}}$$



$$\boxed{\mathbf{x}_d = f_d(\mathbf{x}_{d-1}; \boldsymbol{\theta}_d)}$$

Optimization - Weight Decay

- L_2 Regularization

$$\boxed{\mathrm{Reg}(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_2^2}$$

- Overall objective function

$$\boxed{\frac{1}{m}\sum_{i=1}^{m} L(i) + \lambda\|\boldsymbol{\theta}\|_2^2}$$

$$\boxed{\mathbf{g} \leftarrow \frac{1}{m}\Delta_{\boldsymbol{\theta}}\sum_{i=1}^{m} L(f(\mathbf{x}^i;\boldsymbol{\theta}), y^i) + \boxed{\lambda\boldsymbol{\theta}}}$$

Gradient update

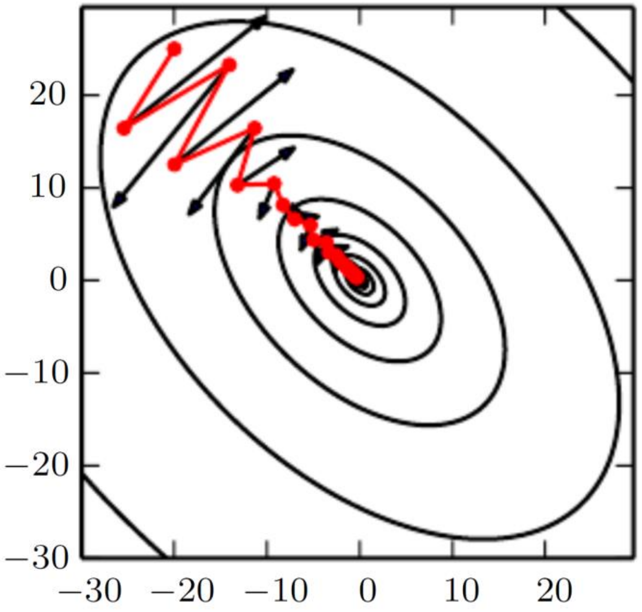
Optimization - Momentum

<u>Handle two problems</u>

- Variance in the stochastic gradient

$$\boxed{\mathbf{v} \leftarrow \alpha \mathbf{v} - \epsilon \mathbf{g}}$$

- Poor conditioning of the Hessian matrix



Optimization ......

- Initialization
  - Gaussian
  - MSRA
- SGD variants
  - AdaGrad (Adaptive gradient algorithm)
  - RMSProp (Root Mean Square Propagation)
  - Adam (Adaptive Moment Estimation)
  - ......
- Second-order optimization
- ......

MVA Microsoft Virtual Academy

04. Recent Advanced Techniques

Microsoft

Batch Normalization – Make Training Easier

Input: response values over a mini-batch $\mathcal{B} = \{x_1, x_2, \ldots, x_m\}$

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

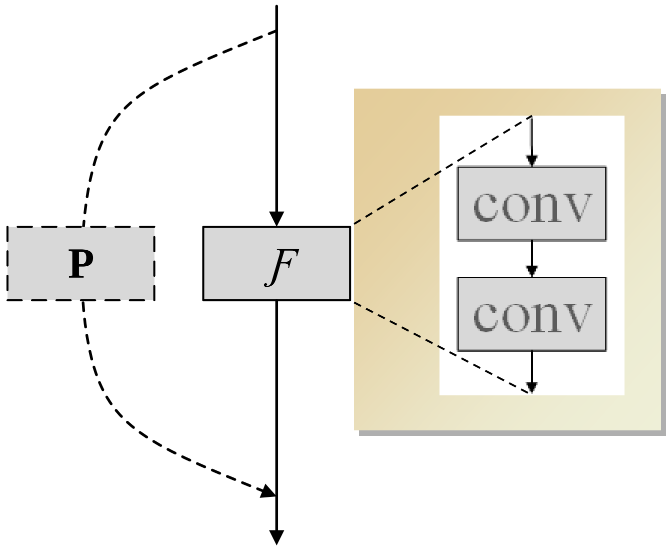
Algorithm:

1. Compute the mean $$\mu \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$
2. Compute the variance $$\sigma^2 \leftarrow \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \mu)^2$$
3. Normalize $$x_i' \leftarrow \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$
4. Scale and shift $$y_i \leftarrow \gamma x_i' + \beta$$

Sergey Ioffe, Christian Szegedy: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ICML 2015: 448-456

Skip Connection – Make Training Easier

Skip connection



$$\boxed{\mathbf{y} = \mathbf{P}\mathbf{x} + \mathcal{F}(\mathbf{x}, \mathcal{W})}$$

For improving information flow

- Identity transformation

$$\boxed{\mathbf{P} = \mathbf{I}}$$

- Idempotent transformation

$$\boxed{\mathbf{P}^n = \mathbf{P}, n = 1, 2, \ldots}$$

- Orthogonal transformation

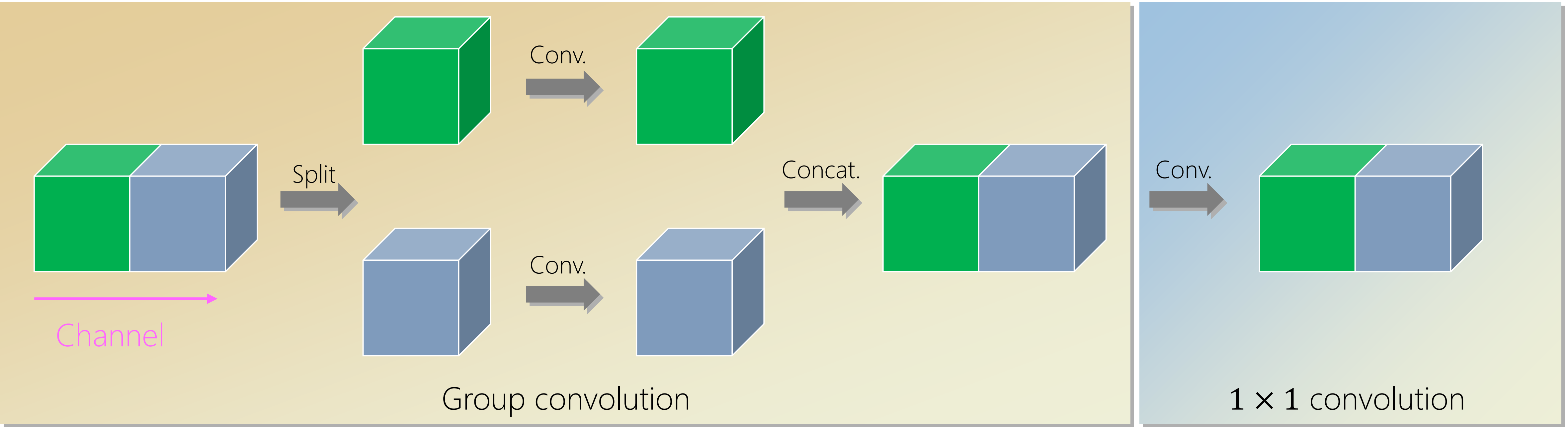
$$\boxed{\mathbf{P}^\top \mathbf{P} = \mathbf{P}\mathbf{P}^\top = \mathbf{I}}$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Deep Residual Learning for Image Recognition. CVPR 2016: 770-778

Jingdong Wang, Yajie Xing, Kexin Zhang, Cha Zhang: Orthogonal and Idempotent Transformations for Learning Deep Neural Networks. CoRR abs/1707.05974 (2017)

Group Convolution – Improve Parameter Efficiency

- Group convolution
  - Split + separate convolution + concatenation
  - Extreme: channel-wise
- Combination with 1×1 convolution



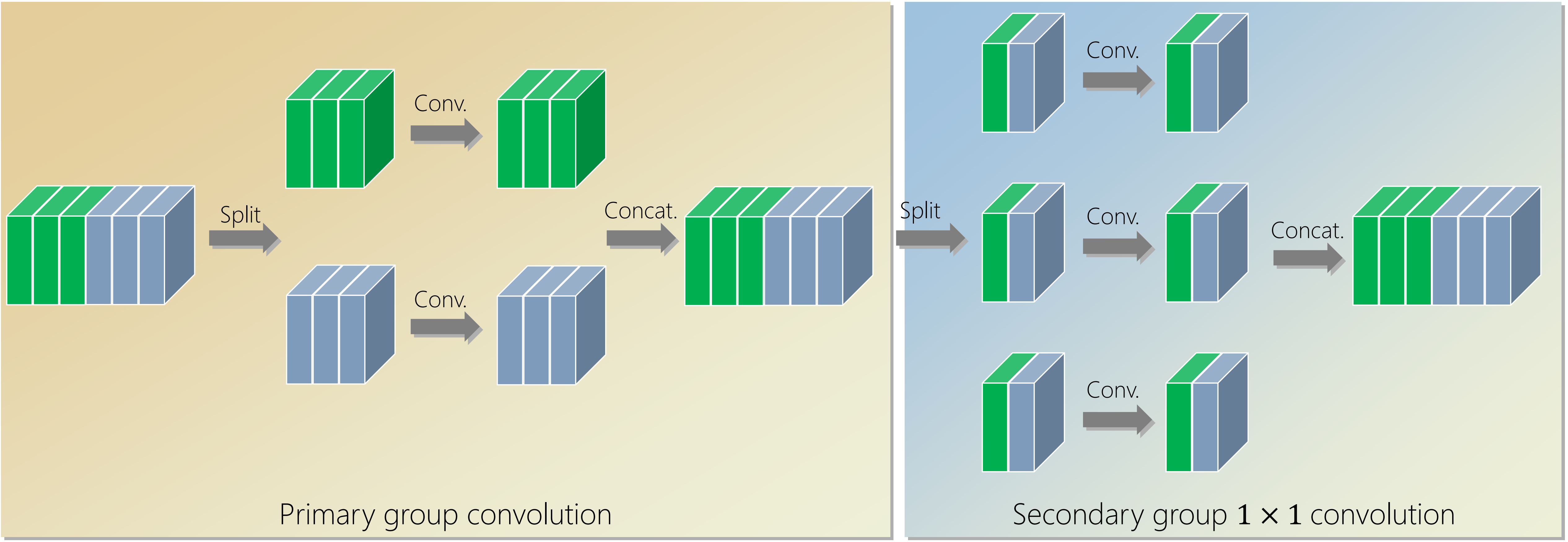
François Chollet: Xception: Deep Learning with Depthwise Separable Convolutions. CoRR abs/1610.02357 (2016)

Yani Ioannou, Duncan P. Robertson, Roberto Cipolla, Antonio Criminisi: Deep Roots: Improving CNN Efficiency with Hierarchical Filter Groups. CoRR abs/1605.06489 (2016)

Interleaved Group Convolution – Improve Parameter Efficiency

Group convolutions are complementary

- The channels in the same secondary partition are from different primary partitions



Ting Zhang, Guo-Jun Qi, Bin Xiao, Jingdong Wang: Interleaved Group Convolutions for Deep Neural Networks. ICCV 2017

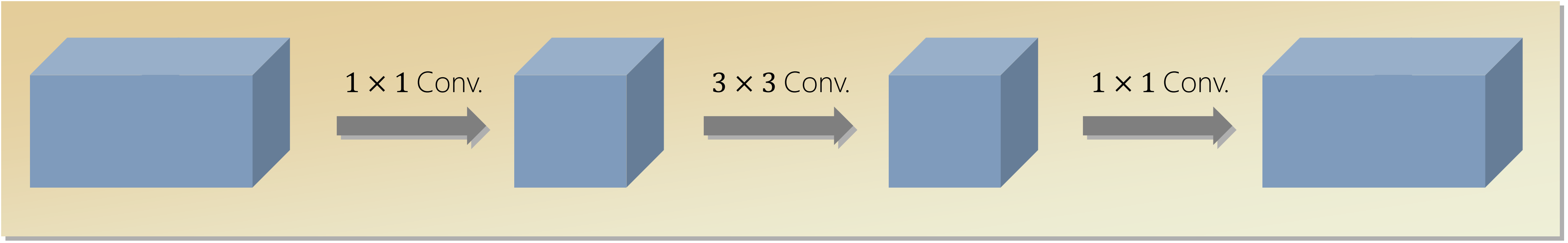
Bottleneck Layer – Improve Parameter Efficiency

Reduce the model size and the time complexity

✓1×1 convolution: reduce the width

✓convolution

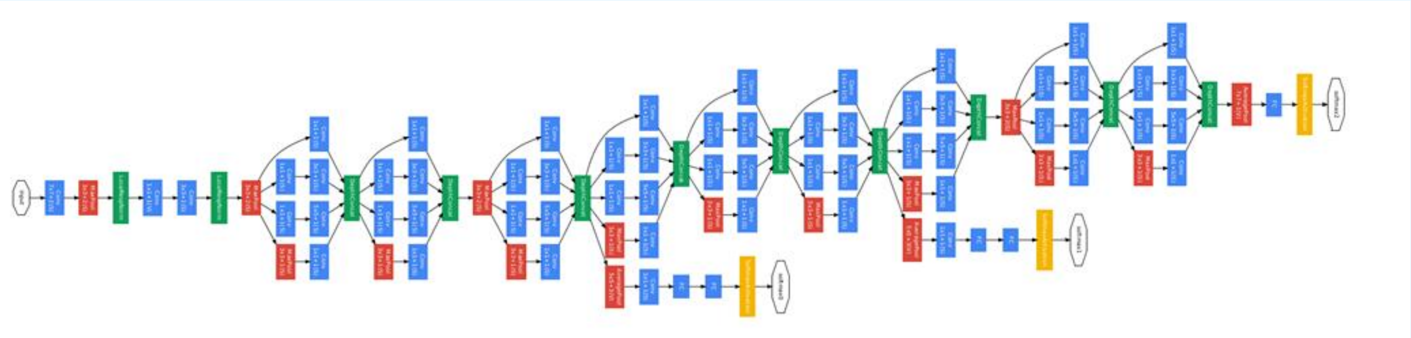
✓1×1 convolution: increase the width



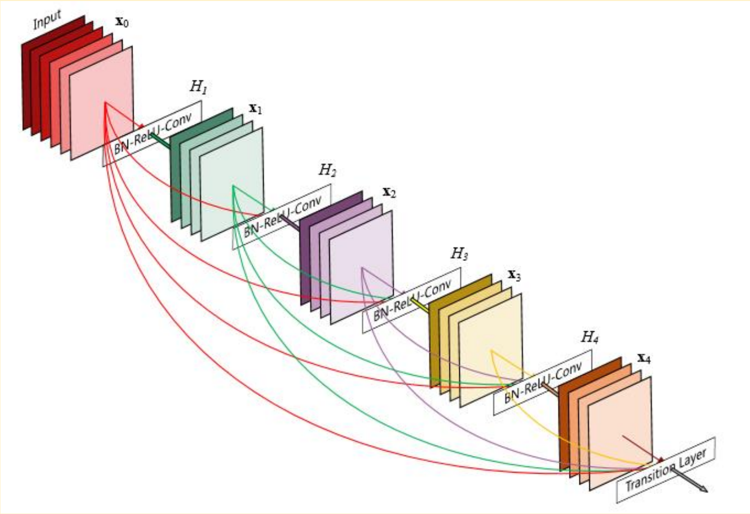
Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich: Going deeper with convolutions. CVPR 2015: 1-9

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Deep Residual Learning for Image Recognition. CVPR 2016: 770-778

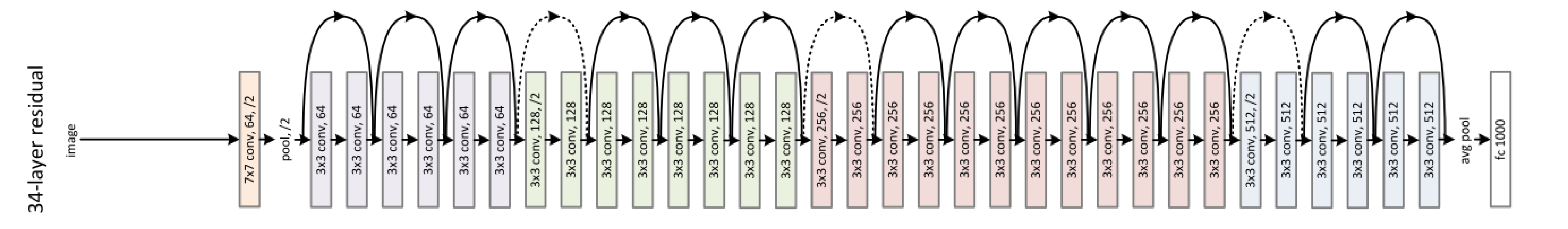
Examples of Advanced CNNs



GoogleNet



DenseNet



ResNet

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich: Going deeper with convolutions. CVPR 2015: 1-9

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Deep Residual Learning for Image Recognition. CVPR 2016: 770-778

Gao Huang, Zhuang Liu, Kilian Q. Weinberger, Laurens van der Maaten: Densely Connected Convolutional Networks. CVPR 2017

References

- Ian Goodfellow and Yoshua Bengio and Aaron Courville: Deep Learning. MIT Press, 2016
- The Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition http://cs231n.github.io/