香港中文大學
The Chinese University of Hong Kong

CENG5030

Part 2-1: Introduction to Convolutional Nueral Network

Bei Yu

(Latest update: March 4, 2019)

Spring 2019

Overview

CNN Architecture Overview

CNN Energy Efficiency

CNN on Embedded Platform

Overview

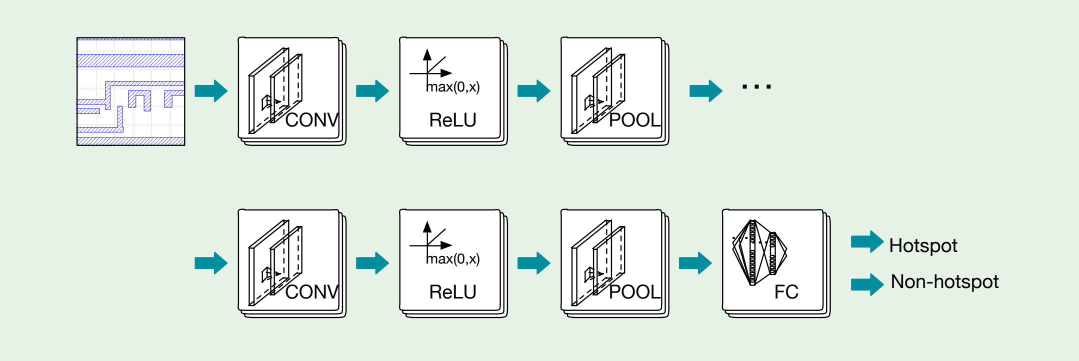
CNN Architecture Overview

CNN Energy Efficiency

CNN on Embedded Platform

CNN Architecture Overview

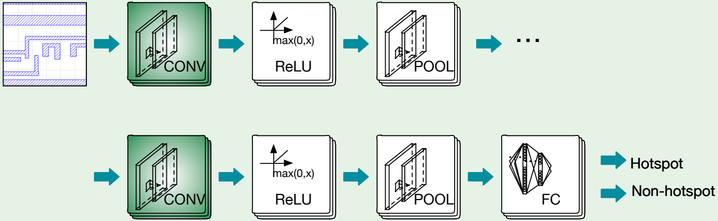
- Convolution Layer
- Rectified Linear Unit (ReLU)
- Pooling Layer
- Fully Connected Layer



Convolution Layer

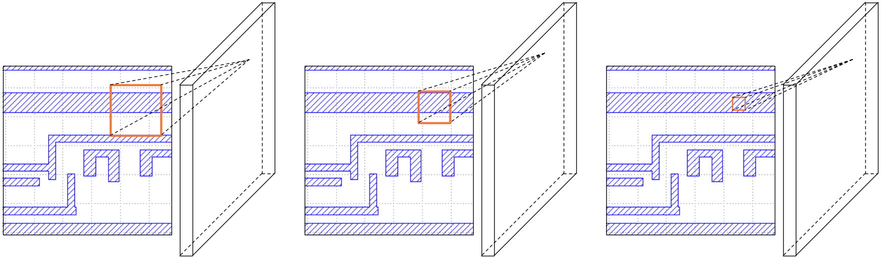
Convolution Operation:

$$\mathbf{I} \otimes \mathbf{K}(x, y) = \sum_{i=1}^{c} \sum_{j=1}^{m} \sum_{k=1}^{m} \mathbf{I}(i, x - j, y - k)\mathbf{K}(j, k)$$



Convolution Layer (cont.)

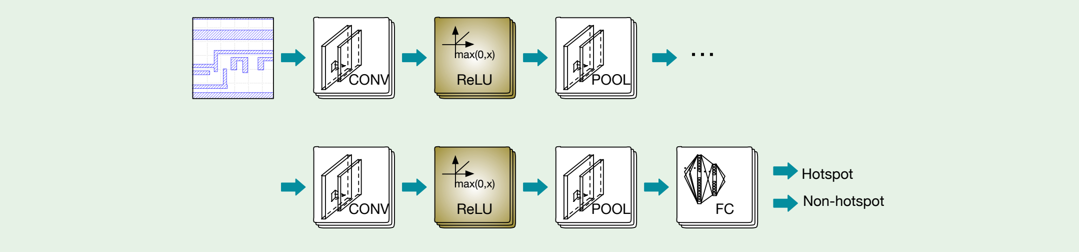
Effect of different convolution kernel sizes:



(a) 7×7 (b) 5×5 (c) 3×3

Kernel Size	Padding	Test Accuracy
7×7	3	87.50%
5×5	2	93.75%
3×3	1	**96.25%**

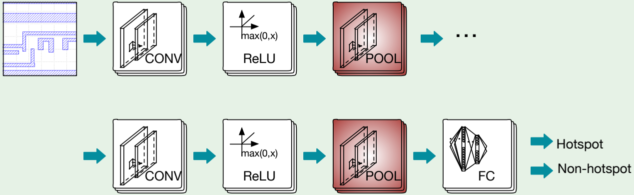
Rectified Linear Unit



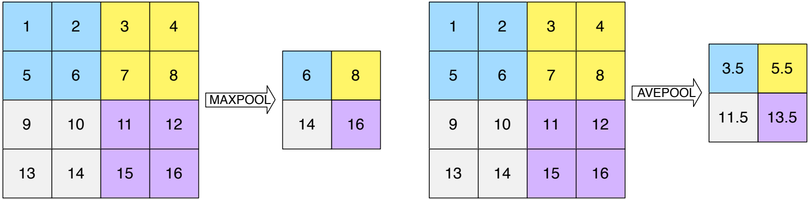
- Alleviate overfitting with sparse feature map
- Avoid gradient vanishing problem

Activation Function	Expression	Validation Loss
ReLU	$\max\{x, 0\}$	**0.16**
Sigmoid	$\frac{1}{1+\exp(-x)}$	87.0
TanH	$\frac{\exp(2x)-1}{\exp(2x)+1}$	0.32
BNLL	$\log(1+\exp(x))$	87.0
WOAF	NULL	87.0

Pooling Layer



- Extracts the local region statistical attributes in the feature map



(a) max pooling (b) avg pooling

Pooling Layer (cont.)

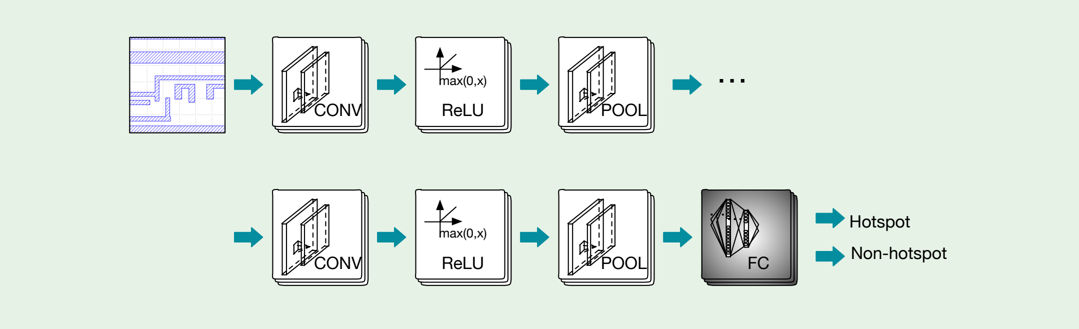
- Translation invarient
- Dimension reduction

Effect of pooling methods:

Pooling Method	Kernel	Test Accuracy
Max	2×2	**96.25%**
Ave	2×2	**96.25%**
Stochastic	2×2	90.00%

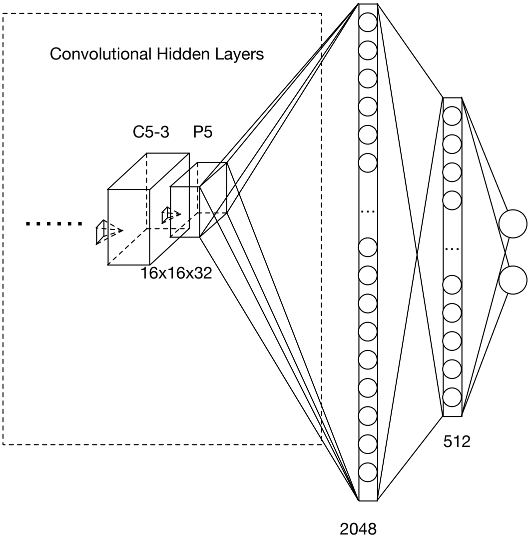
Fully Connected Layer

- Fully connected layer transforms high dimension feature maps into flattened vector.

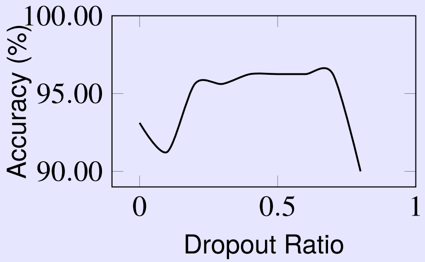


Fully Connected Layer (cont.)

- A percentage of nodes are dropped out (i.e. set to zero)
- avoid overfitting

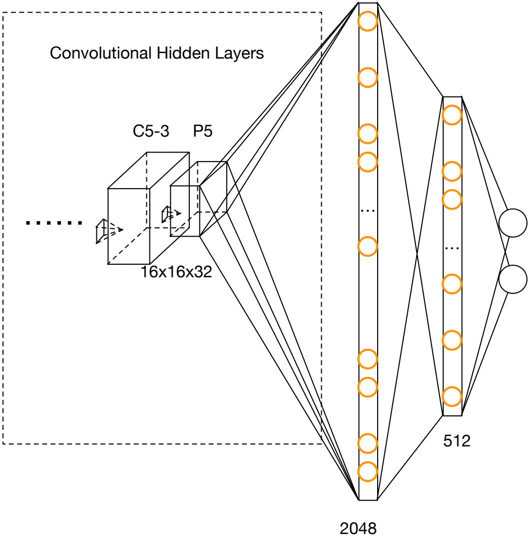


Effect of dropout ratio:

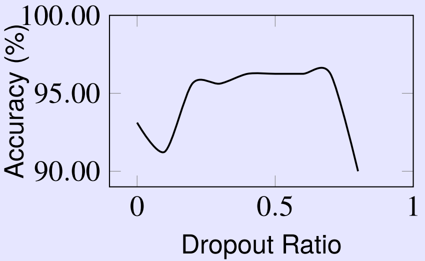


Fully Connected Layer (cont.)

- A percentage of nodes are dropped out (i.e. set to zero)
- avoid overfitting



Effect of dropout ratio:



Overview

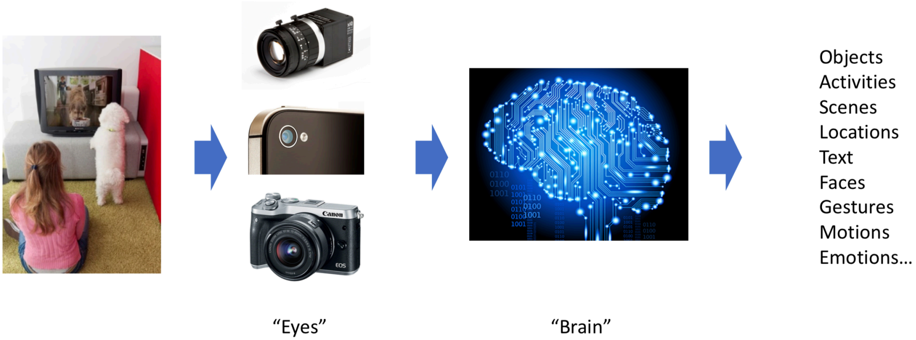
CNN Architecture Overview

CNN Energy Efficiency

CNN on Embedded Platform

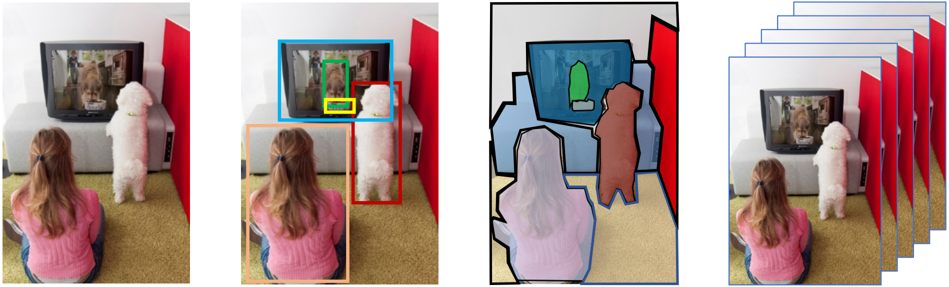
Computer Vision

- Humans use their eyes and their brains to visually sense the world.
- Computers user their cameras and computation to visually sense the world



Jian Sun, "Introduction to Computer Vision and Deep Learning".

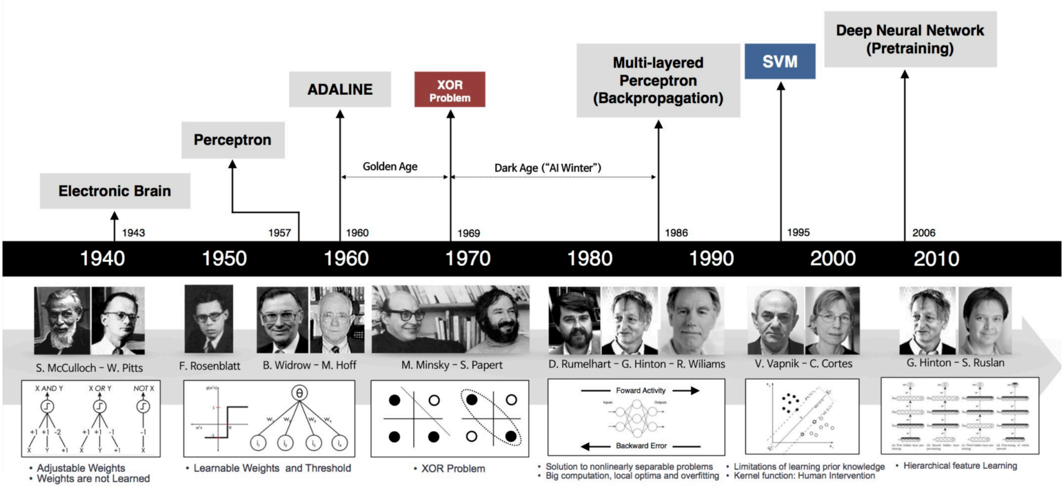
Few More Core Problems



Classification | Detection | Segmentation | Sequence

Image → Region → Pixel | Video

A Bit of History



Jian Sun, "Introduction to Computer Vision and Deep Learning".

Winter of Neural Networks (mid 90' – 2006)

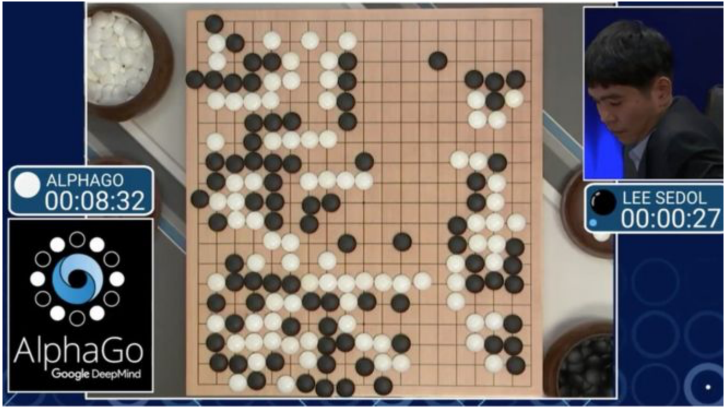
- The rises of SVM, Random forest
- No theory to play
- Lack of training data
- Benchmark is insensitive
- Difficulties in optimization
- Hard to reproduce results

Curse

"Deep neural networks are no good and could never be trained."

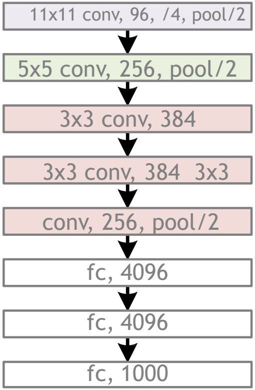
Renaissance of Deep Learning (2006 –)

- A fast learning algorithm for deep belief nets. [Hinton et.al 1996]
- Data + Computing + Industry Competition
- NVidia's GPU, Google Brain (16,000 CPUs)
- Speech: Microsoft [2010], Google [2011], IBM
- Image: AlexNet, 8 layers [Krizhevsky et.al 2012] (26.2% -> 15.3%)



Revolution of Depth

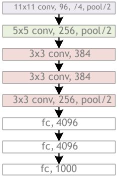
AlexNet, 8 layers (ILSVRC 2012)



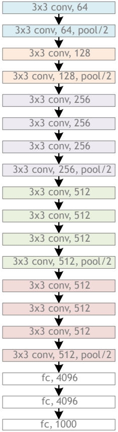
Slide Credit: He et al. (MSRA)

Revolution of Depth

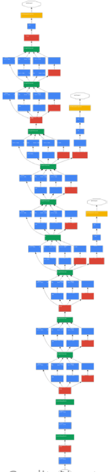
AlexNet, 8 layers (ILSVRC 2012)



VGG, 19 layers (ILSVRC 2014)



GoogleNet, 22 layers (ILSVRC 2014)



Slide Credit: He et al. (MSRA)

Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)

VGG, 19 layers (ILSVRC 2014)

ResNet, 152 layers (ILSVRC 2015)

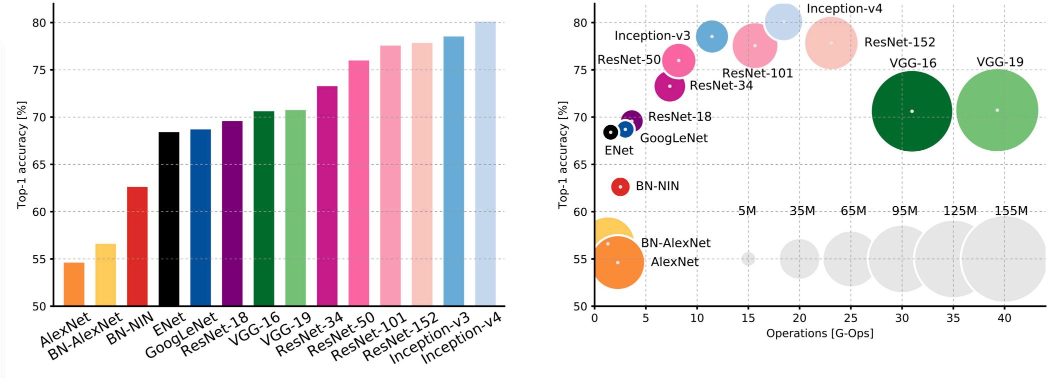
Slide Credit: He et al. (MSRA)

Some Recent Classification Architectures

- AlexNet (Krizhevsky, Sutskever, and E. Hinton 2012) 233MB
- Network in Network (Lin, Chen, and Yan 2013) 29MB
- VGG (Simonyan and Zisserman 2015) 549MB
- GoogleNet (Szegedy, Liu, et al. 2015) 51MB
- ResNet (He et al. 2016) 215MB
- Inception-ResNet (Szegedy, Vanhoucke, et al. 2016)
- DenseNet (Huang et al. 2017)
- Xception (Chollet 2017)
- MobileNetV2 (Sandler et al. 2018)
- ShuffleNet (Zhang et al. 2018)

Some Recent Classification Architectures

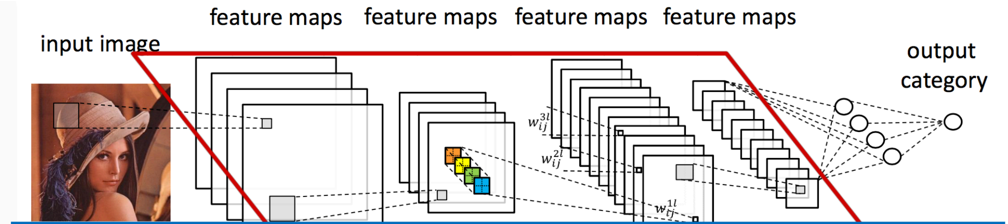
- AlexNet (Krizhevsky, Sutskever, and E. Hinton 2012) 233MB
- Network in Network (Lin, Chen, and Yan 2013) 29MB
- VGG (Simonyan and Zisserman 2015) 549MB
- GoogleNet (Szegedy, Liu, et al. 2015) 51MB
- ResNet (He et al. 2016) 215MB
- Inception-ResNet (Szegedy, Vanhoucke, et al. 2016) 23MB
- DenseNet (Huang et al. 2017) 80MB
- Xception (Chollet 2017) 22MB
- MobileNetV2 (Sandler et al. 2018) 14MB
- ShuffleNet (Zhang et al. 2018) 22MB



[1]

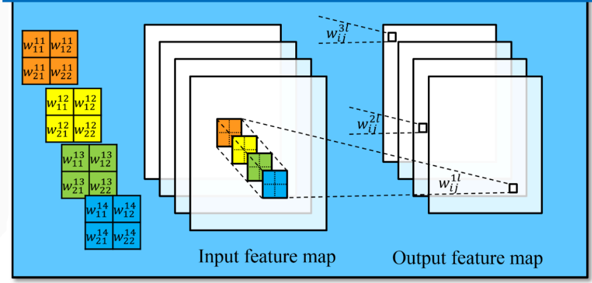
[1] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello (2017). “An analysis of deep neural network models for practical applications”. In: *arXiv preprint*.

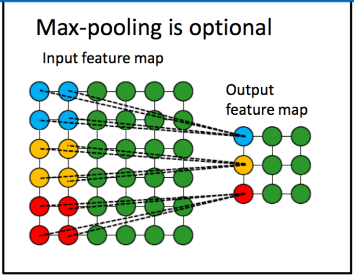
Convolutional Neural Network (CNN)



Convolutional layers account for over 90% computation

[1] A. Krizhevsky, etc. Imagenet classification with deep convolutional neural networks. NIPS 2012.
[2] J. Cong and B. Xiao. Minimizing computation in convolutional neural networks. ICANN 2014





Overview

CNN Architecture Overview

CNN Energy Efficiency

CNN on Embedded Platform

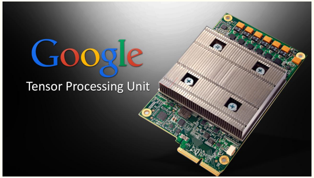
When Machine Learning Meets Hardware

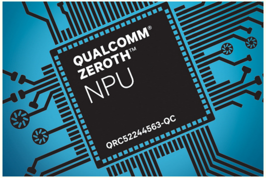
Convolution layer is one of the most expensive layers

- Computation pattern
- Emerging challenges

More and more end-point devices with limited memory

- Cameras
- Smartphone
- Autonomous driving





DEEPHI TECH

MOBILEYE®
An intel company

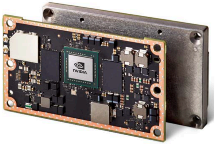
Application Category

Both	Datacenter	Edge
Intel, Nvidia, IBM, Xilinx, HiSilicon, Google, Baidu, Alibaba Group, Cambricon, DeePhi, Bitmain, Wave Computing	AMD, Microsoft, Apple, Tencent Cloud,Aliyun, Baidu Cloud, HUAWEI Cloud, Fujitsu, Nokia, Facebook, HPE, Thinkforce, Cerebras, Graphcore, Groq, SambaNova Systems, Adapteva, PEZY	Qualcomm, Samsung, STMicroelectronics, NXP, MediaTek, Rockchip, Amazon_AWS, ARM, Synopsys, Imagination, CEVA, Cadence, VeriSilicon, Videantis, Horizon Robotics, Chipintelli, Unisound, AISpeech, Rokid, KnuEdge, Tenstorrent, ThinCI, Koniku, Knowm, Mythic, Kalray, BrainChip, AImotive, DeepScale, Leepmind, Krtkl, NovuMind, REM, TERADEEP, DEEP VISION, KAIST DNPU, Kneron, Esperanto Technologies, Gyrfalcon Technology, GreenWaves Technology, Lightelligence, Lightmatter, ThinkSilicon, Innogrit, Kortiq, Hailo,Tachyum

Source: https://basicmi.github.io/Deep-Learning-Processor-List/

Flexibility vs. Efficiency

CPU
(Raspberry Pi3)



GPU
(Jetson TX2)

FPGA
(UltraZed)

ASIC
(Movidius)

Flexibility ⟷ Power/Performance Efficiency