CMSC 5743
Efficient Computing of Deep Neural Networks

Lecture 01: Introduction

Bei Yu
CSE Department, CUHK
byu@cse.cuhk.edu.hk

(Latest update: September 2, 2023)

2023 Fall

What We Focus on?

What you expect to Learn?

How About the Workload?

Grading System?

Overview

Overview

Object Detection: PASCAL VOC mean Average Precision (mAP)

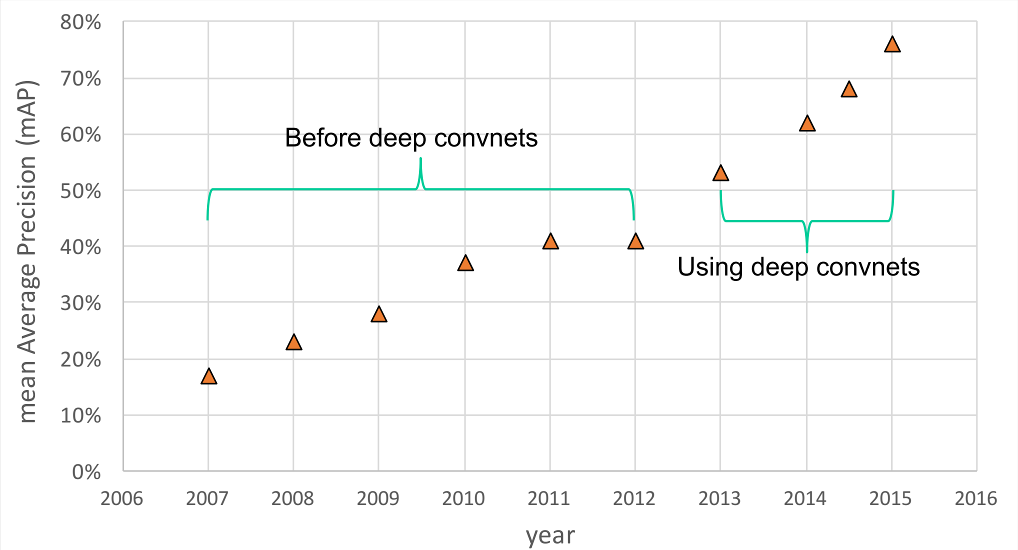
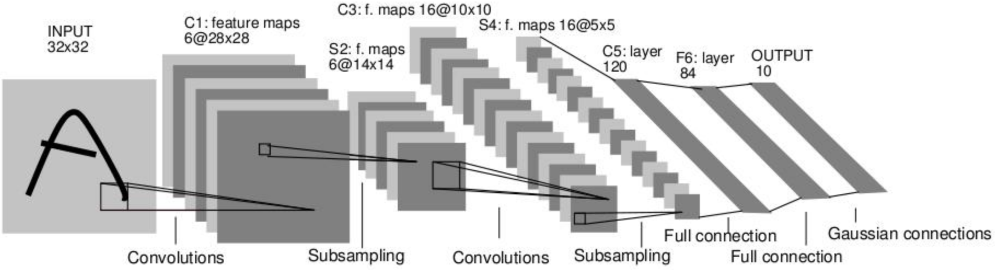


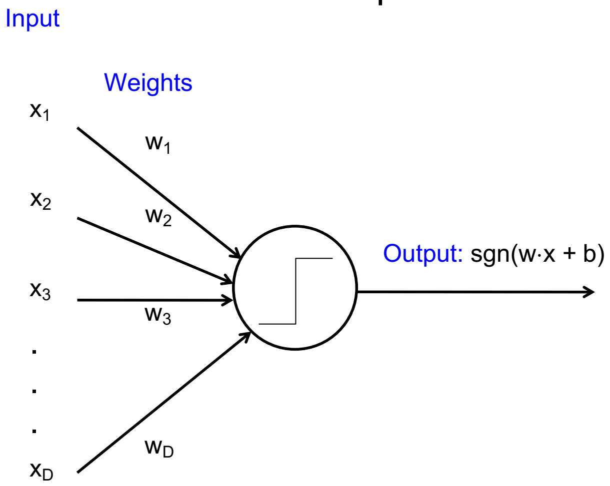
Figure source: Ross Girshick

LeNet 5



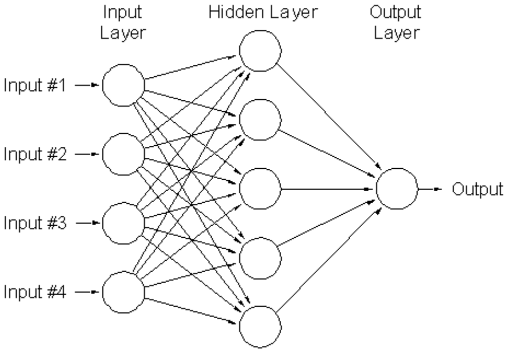
Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278–2324, 1998.

The Perceptron

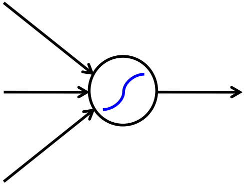


Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386–408.

Two-layer neural network



- Can learn nonlinear functions provided each perceptron has a differentiable nonlinearity

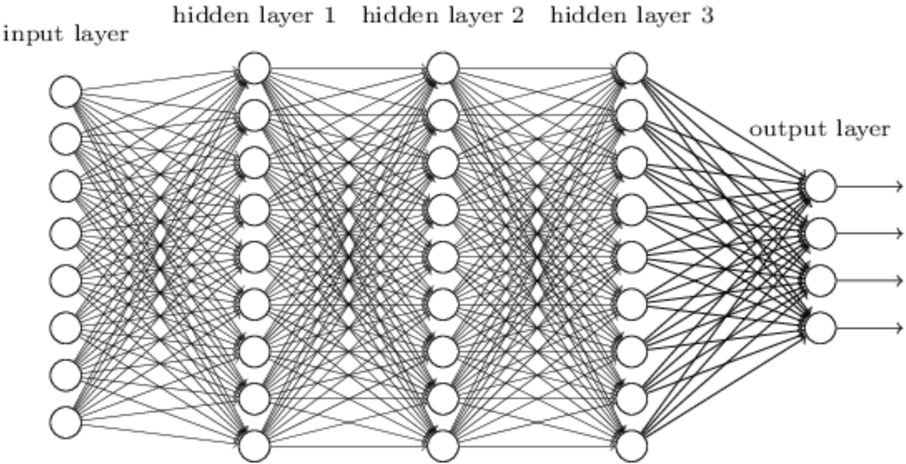


Sigmoid: $g(t) = \dfrac{1}{1+e^{-t}}$

What is the value range of sigmoid activation?

- $[-1, 1]$
- $[-\infty, +\infty]$
- $[0, 1]$
- $[0, +\infty]$

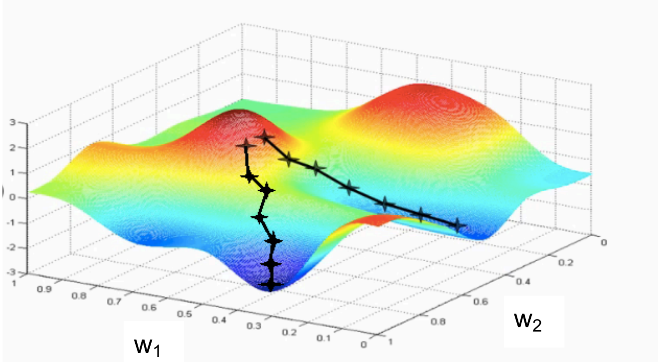
Multi-layer neural network



- Find network weights to minimize the *training error* between true and estimated labels of training examples, e.g.:

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i) \right)^2$$

- Update weights by **gradient descent:** $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial E}{\partial \mathbf{w}}$



Training of multi-layer networks

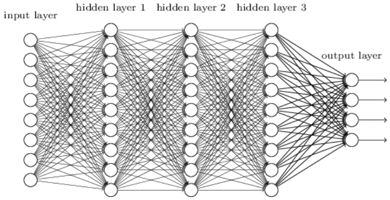
- Find network weights to minimize the *training error* between true and estimated labels of training examples, e.g.:

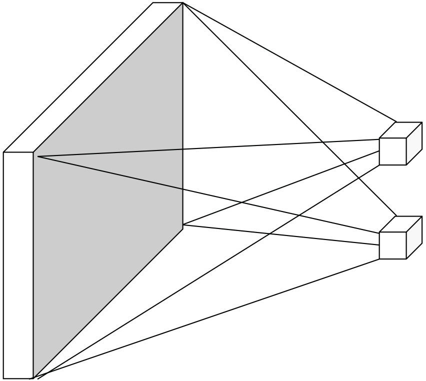
$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i) \right)^2$$

- Update weights by **gradient descent:** $\mathbf{w} \leftarrow \mathbf{w} - \alpha \dfrac{\partial E}{\partial \mathbf{w}}$

- **Back-propagation:** gradients are computed in the direction from output to input layers and combined using chain rule
- **Stochastic gradient descent:** compute the weight update w.r.t. one training example (or a small batch of examples) at a time, cycle through training examples in random order in multiple epochs

From fully connected to convolutional networks

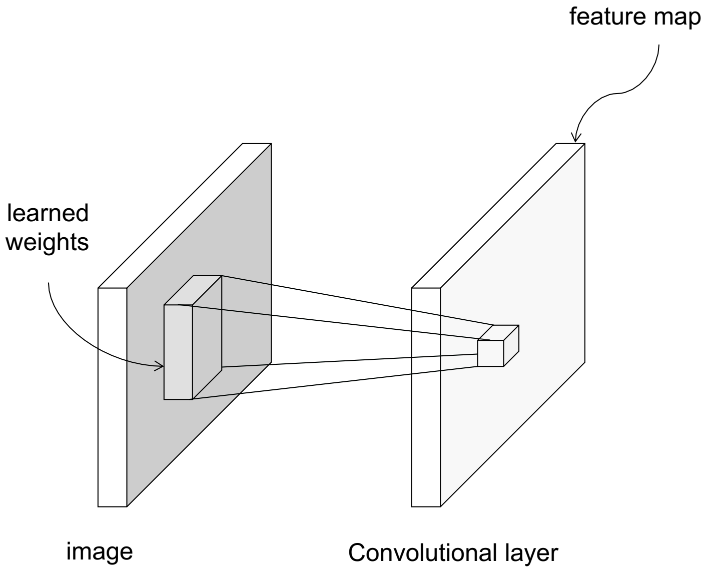




image

Fully connected layer

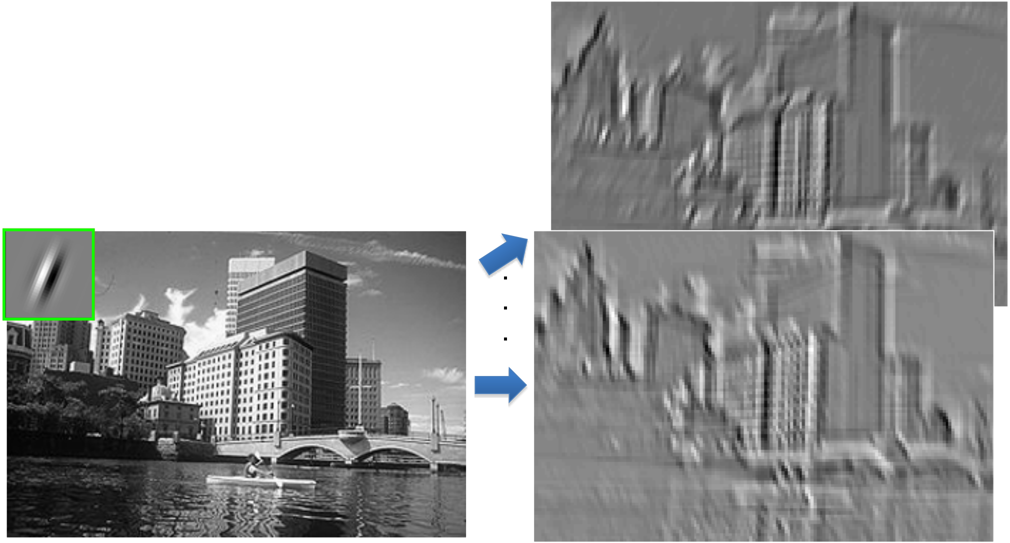
From fully connected to convolutional networks



For a convolution kernel with kernel size 3, stride 1, what is the zero padding number to keep the output feature map size unchanged?

- 0
- 1
- 2
- 3

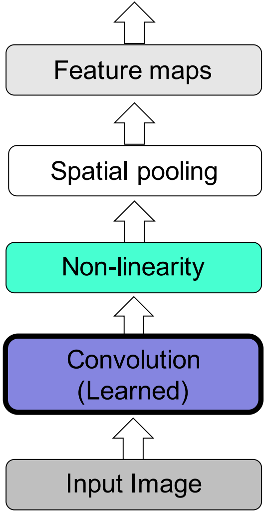
Convolution as feature extraction

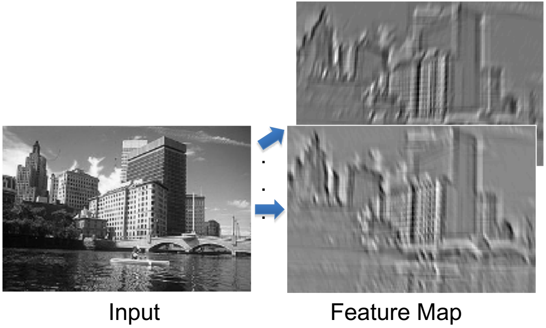


Input

Feature Map

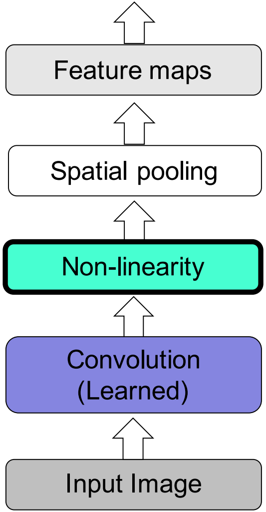
Key operations

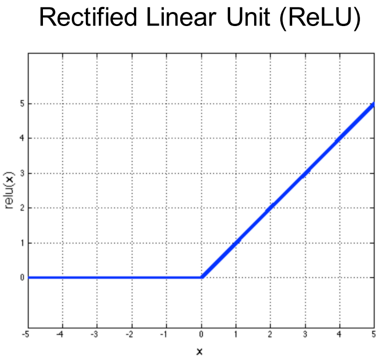




Source: R. Fergus, Y. LeCun

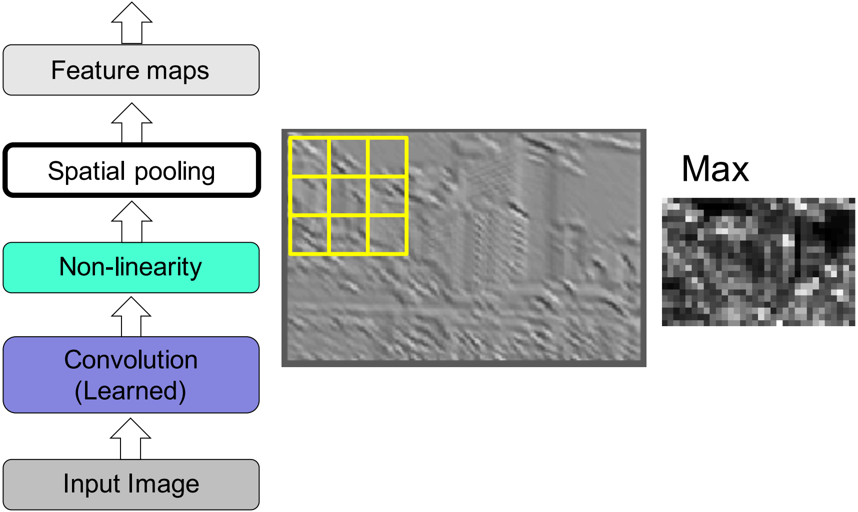
Key operations





Source: R. Fergus, Y. LeCun

Key operations



Source: R. Fergus, Y. LeCun

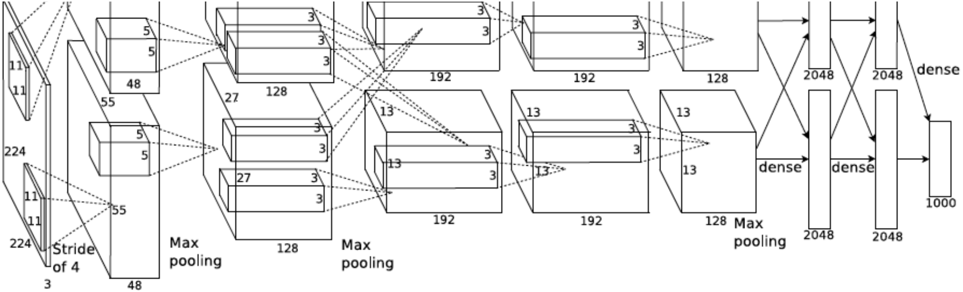
Fast forward to the arrival of big visual data

IMAGENET

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon MTurk
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC): 1.2 million training images, 1000 classes

www.image-net.org/challenges/LSVRC/

AlexNet: ILSVRC 2012 winner



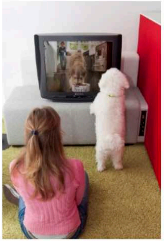
- Similar framework to LeNet but:
  - Max pooling, ReLU nonlinearity
  - More data and bigger model (7 hidden layers, 650K units, 60M params)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week
  - Dropout regularization

A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](), NIPS 2012

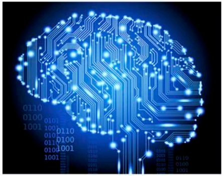
Overview

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

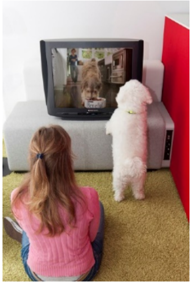


“Eyes”

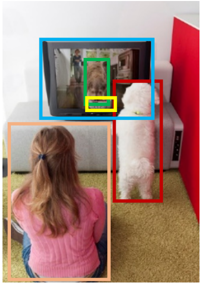


“Brain”

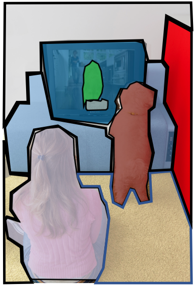
Objects
Activities
Scenes
Locations
Text
Faces
Gestures
Motions
Emotions...



Classification



Detection



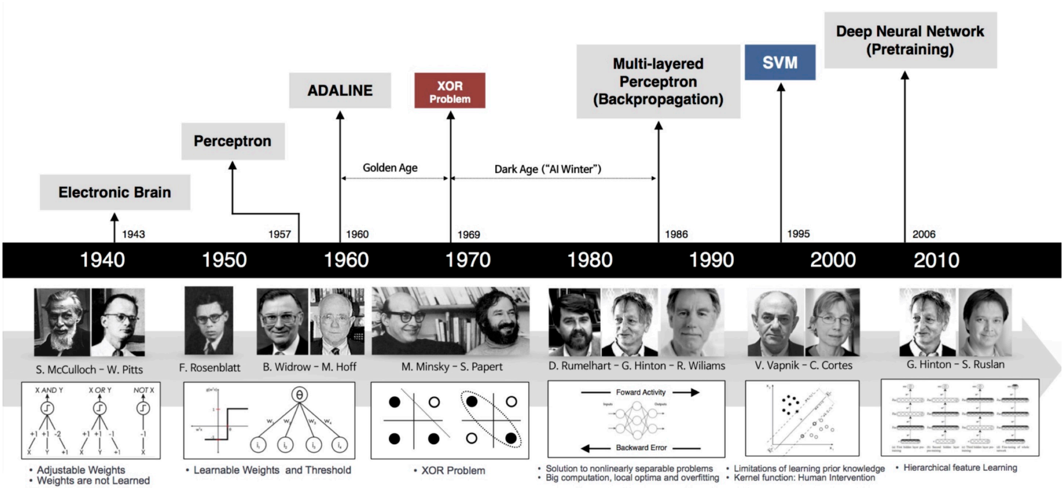
Segmentation

Sequence

Image → Region → Pixel

Video

A Bit of History



Winter of Neural Networks (mid 90′ – 2006)

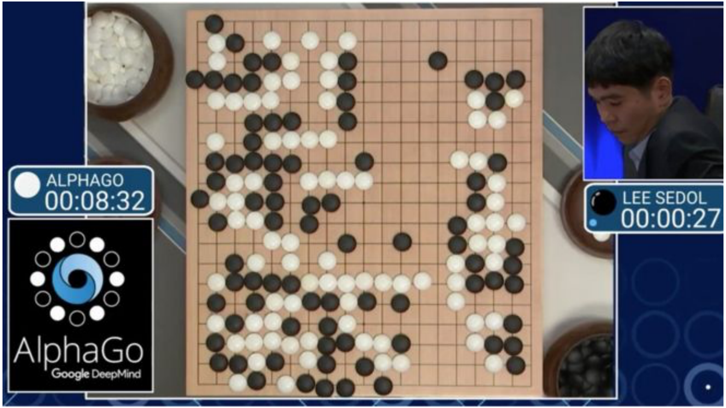
- The rises of SVM, Random forest
- No theory to play
- Lack of training data
- Benchmark is insensitive
- Difficulties in optimization
- Hard to reproduce results

Curse

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

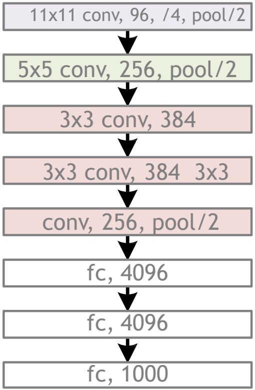
Renaissance of Deep Learning (2006 –)

- 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

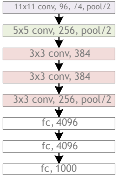
AlexNet, 8 layers (ILSVRC 2012)



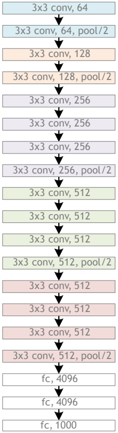
Slide Credit: He et al. (MSRA)

Revolution of Depth

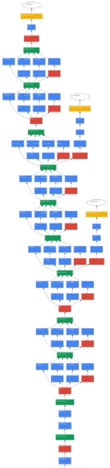
AlexNet, 8 layers (ILSVRC 2012)



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)

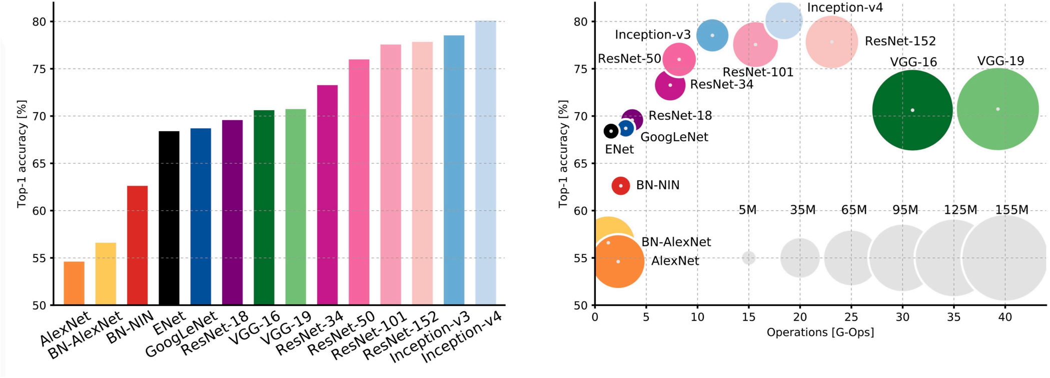
Slide Credit: He et al. (MSRA)

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)

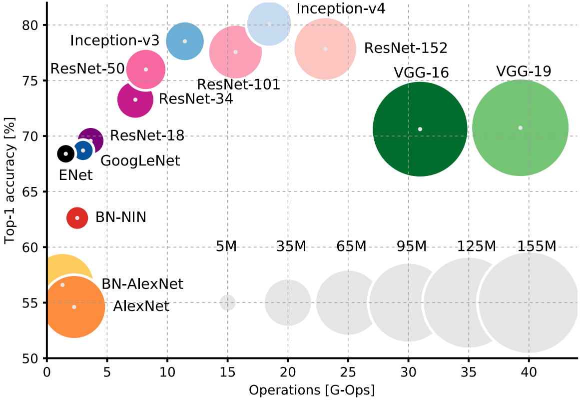
Some Recent Classification Architectures

- 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]

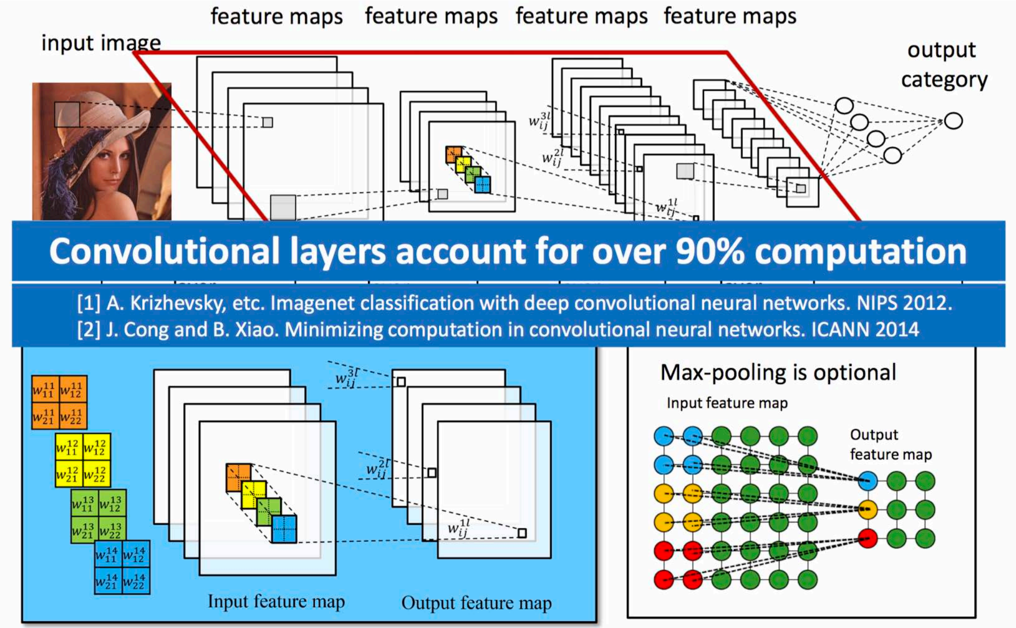
[1] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello (2017). "An analysis of deep neural network models for practical applications". In: *arXiv preprint*.



Why AlexNet is large in size, but small in operations?

- Special FC layers
- Special Conv layers
- More channels
- Some redundant operators

Convolutional Neural Network (CNN)

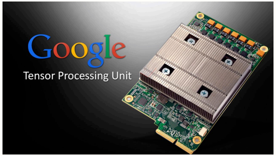


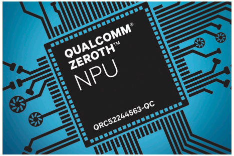
Convolution layer is one of the most expensive layers

- Computation pattern
- Emerging challenges

More and more end-point devices with limited memory

- Cameras
- Smartphone
- Autonomous driving





XILINX

MOBILEYE®
An Intel Company

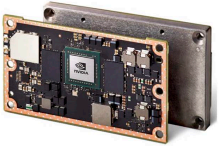
Application Category

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)

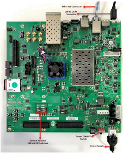
FPGA
(UltraZed)

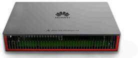
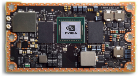
ASIC
(Movidius)

Flexibility

Power/Performance Efficiency

Comparisons: FPGA, ASIC, GPU[2]




	Xilinx ZCU102	Xilinx ZCU104	Huawei Atlas 200	nVIDIA Jetson TX2	Cambricon MLU 270
price	3K RMB	2K RMB	4K RMB	2.8K RMB	12K RMB
MobileNet-V1	1.14 ms	1.37 ms	1.8 ms	12.44 ms	1.85 ms
ResNet50	5.23 ms	6.81 ms	3.6 ms	24.70 ms	2.54 ms
Inception_v2	2.68 ms	3.35 ms	6.0 ms	10.81 ms	5.12 ms
Inception_v3	6.44 ms	8.53 ms	5.7 ms	32.53 ms	4.71 ms
Inception_v4	11.87 ms	17.06 ms	9.3 ms	44.37 ms	11.33 ms

[2]price is NOT accurate – reference purpose.