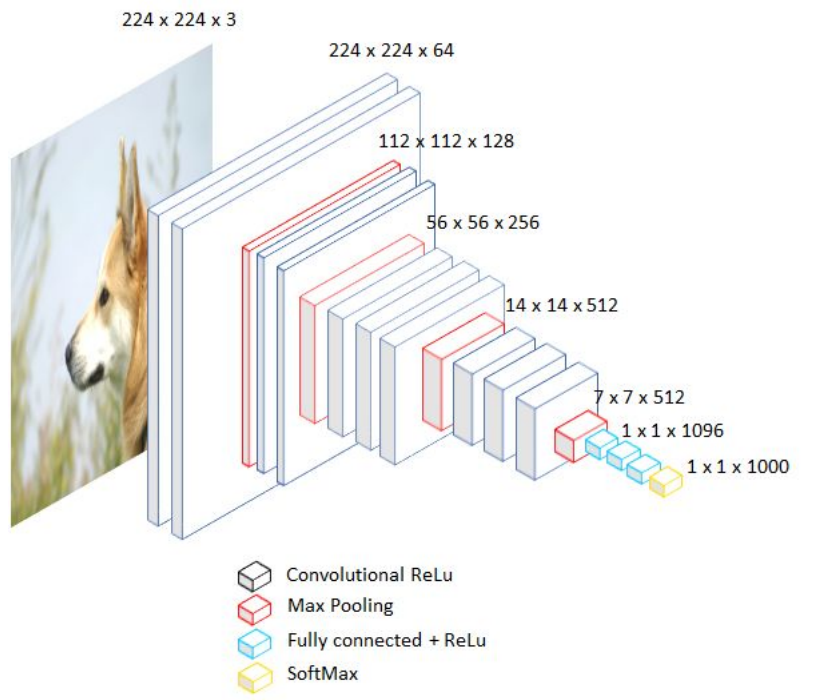


Convolutional Neural Network (Basic) Cheat Sheet

(source: DeepLearning.AI - Andrew Ng)

V2021.01.02

(Dr Yan Xu)

Core Motivations

- ➔ Significantly reduce # of parameters to extract features in hidden layers
- ➔ Sparse localized connections
- ➔ Parameter sharing

Basic Concepts

- ➔ **Feature Extraction Layers:**
  - ○ Convolution Layer
    - ● Stride (default: 1)
    - ● Padding (default: 0)
    - ● Kernel size (or Filter size)
    - ● Kernel count (or Filter count)
  - ○ Non-linear activation function (e.g. ReLU)
  - ○ Pooling Layer
    - ● max pooling

$n \times n$ image $f \times f$ filter

padding p stride s

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

- ➔ **Classification Layers:**
  - ○ Flatten operation
  - ○ Fully Connected Layer
  - ○ Softmax Layer

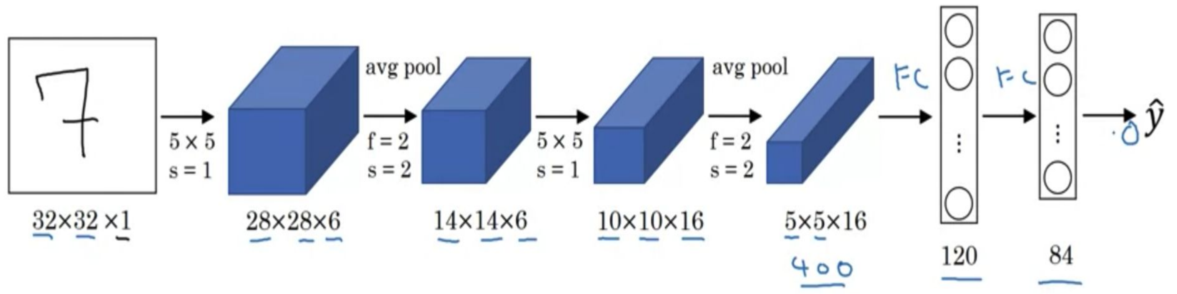
1: The # of parameters in feature extraction layers is small

Parameter Count Example

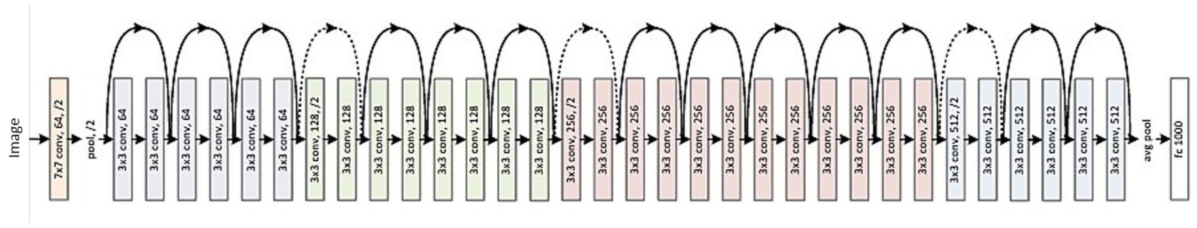
	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	– 3,072 $a^{[0]}$	0
CONV1 (f=5, s=1)	(28,28,8)	6,272	208 ←
POOL1	(14,14,8)	1,568	0 ←
CONV2 (f=5, s=1)	(10,10,16)	1,600	416 ←
POOL2	(5,5,16)	400	0 ←
FC3	(120,1)	120	48,001
FC4	(84,1)	84	10,081
Softmax	(10,1)	10	841

CNN Popular Architectures

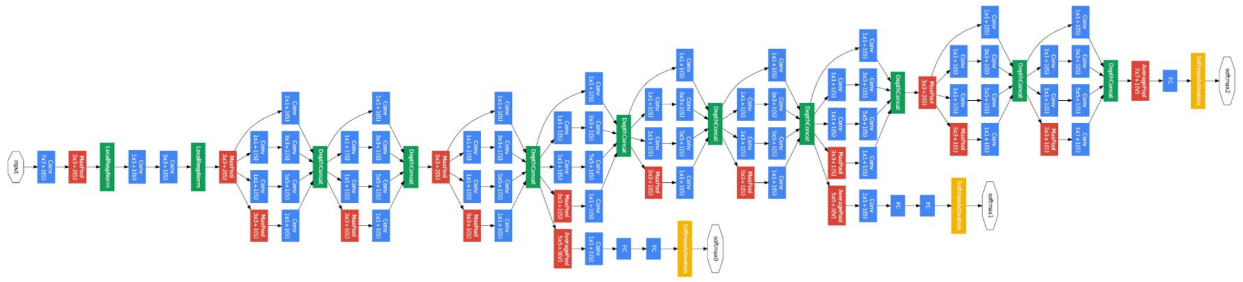
- ➔ Classic Networks
  - ○ LeNet - 5 (*a small network*)
  - ○ AlexNet
  - ○ VGG - 16



- ➔ ResNet (deep networks)
  - ○ ability to learn the identity transformation

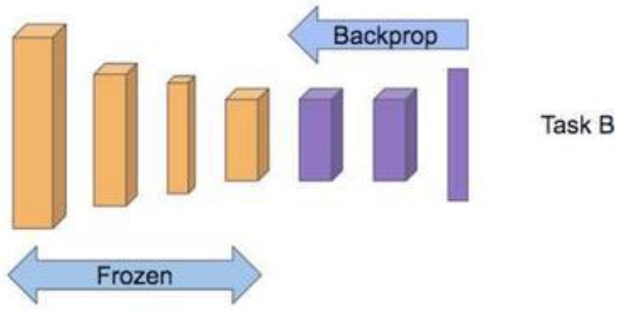


- ➔ Inception Network
  - ○ To concat various sizing filters & polling together
  - ○ Trick: use 1*1 convolution to reduce filters
  - ○ Variant: GoogLeNet



Transfer Learning

- ➔ Frozen layers
- ➔ Initial parameters

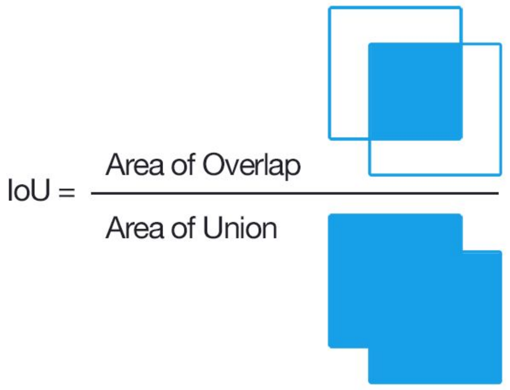


Object Detection (e.g. YOLO)

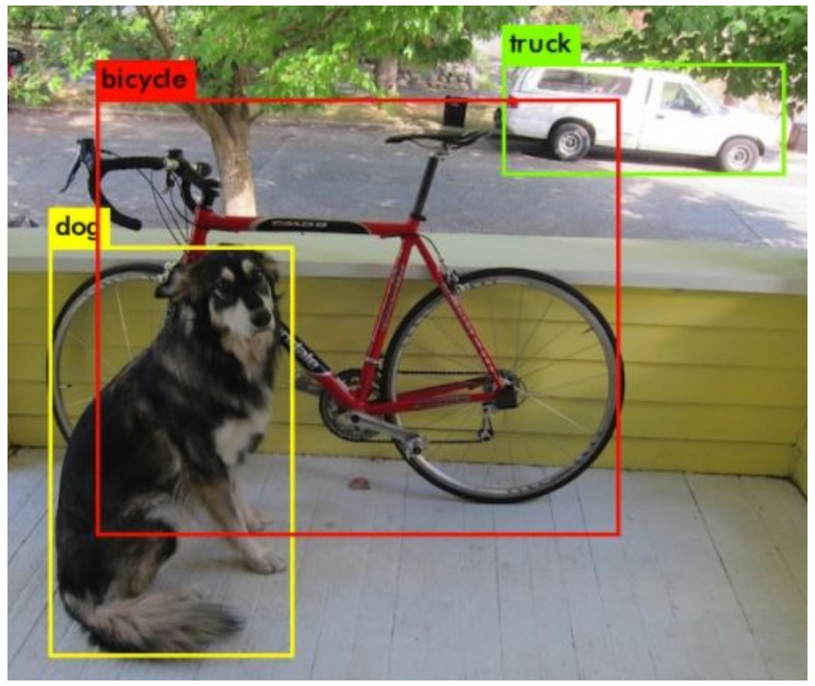
- ➔ Idea 1: Object & Location Learning Together
  - ○ location encoding should be consistent
- ➔ Idea 2: Sliding Window Detection
- ➔ Idea 3: Sliding Window as CNN
  - ○ speed up calculation
- ➔ Idea 4: Object/Cell Ownership
  - ○ divide an image to N * N cells (e.g. n = 19)
  - ○ each object belongs to one cell
  - ○ training / prediction targets for each cell

(note: each cell is learning itself based on nearby cells)

- ➔ Idea 5: Anchor Box
  - ○ Each cell could have 1+ objects
- ➔ Idea 6: Non-Max Suppression
  - ○ Remove predictions which have a higher IOU with a better prediction with the same class
- ➔ Accuracy Measure: IOU



- ➔ Finally, Object Detection: YOLO



(source: https://pjreddie.com/darknet/yolo/)

Other Applications

- ➔ Face Recognition
- ➔ Image Style Transfer
- ➔ Generative Adversarial Networks