Convolutional Neural Network

Tuan Nguyen - AI4E

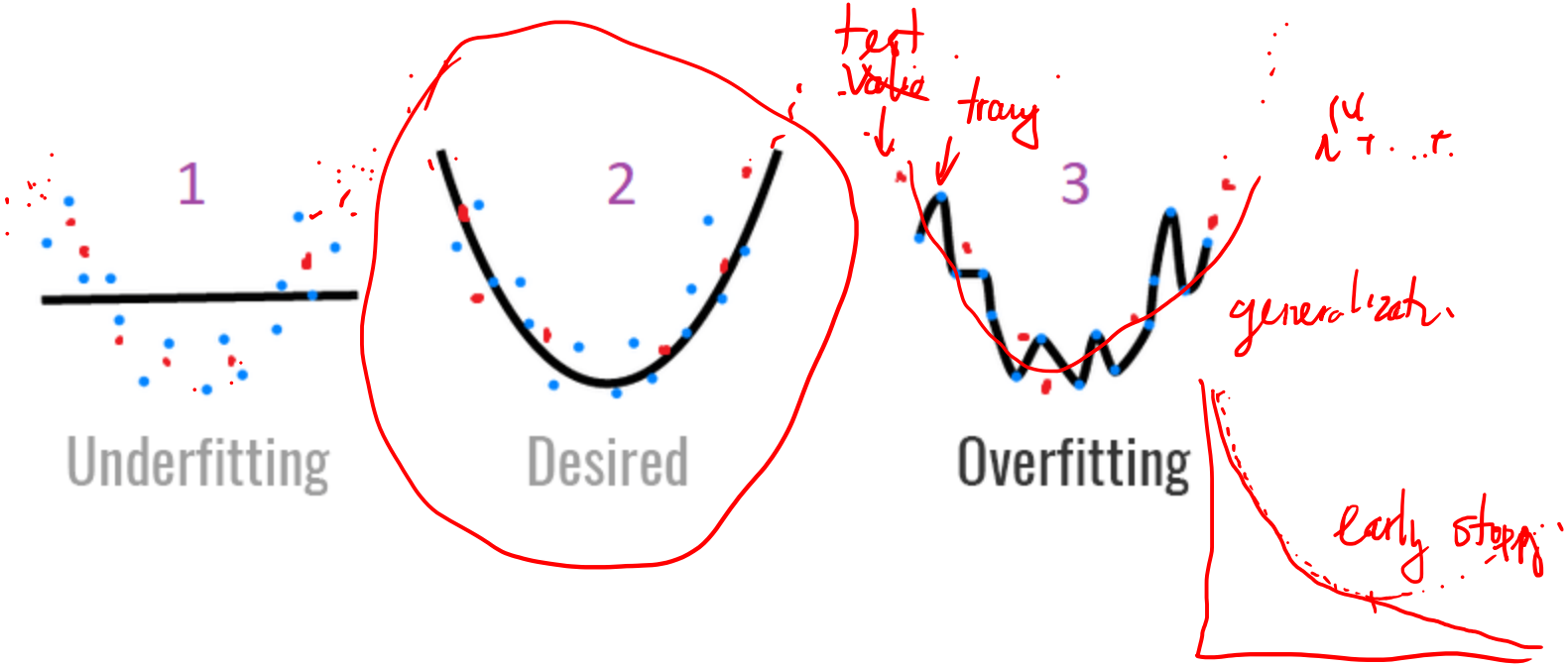
Outline

- Underfitting/Overfitting
- What problem with Neural Network?
- Image processing
- Convolutional operation
- Building blocks (Convolutional layer, pooling layer)
- ImageNet challenge
- VGG 16
- CNN applications

Underfitting/Overfitting

parabol

$y = x^2 + N(0,1) \cdot \alpha$



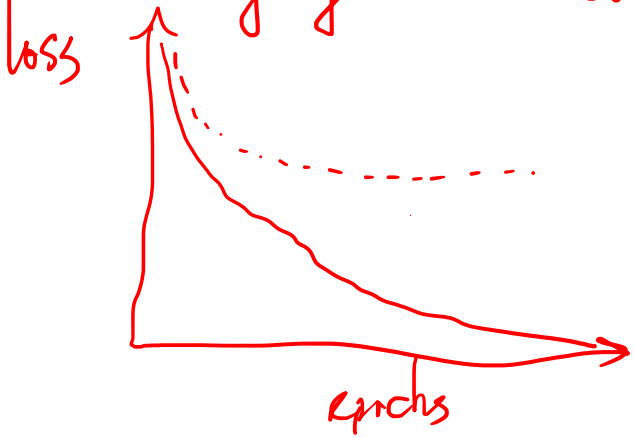
Signal?

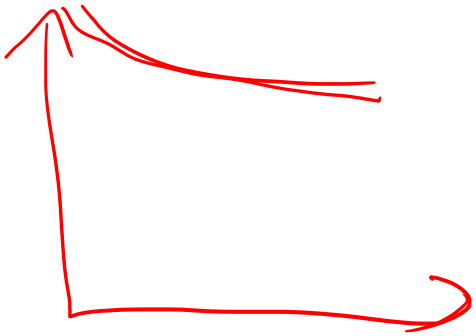
Train set error	**1%**	15%	**15%**	0.5%
Val set error	**11%**	16%	**30%**	1%

overfitting

underfit

diverg.





Solution

- Giải quyết underfitting: Ta cần tăng độ phức tạp của model
  - Tăng số lượng hidden layer và số node trong mỗi hidden layer.
- Giải quyết overfitting:
  - Thu thập thêm dữ liệu hoặc dùng data augmentation
  - Dùng regularization như: L1, L2, Dropout

Dropout

0.2 – 0.5

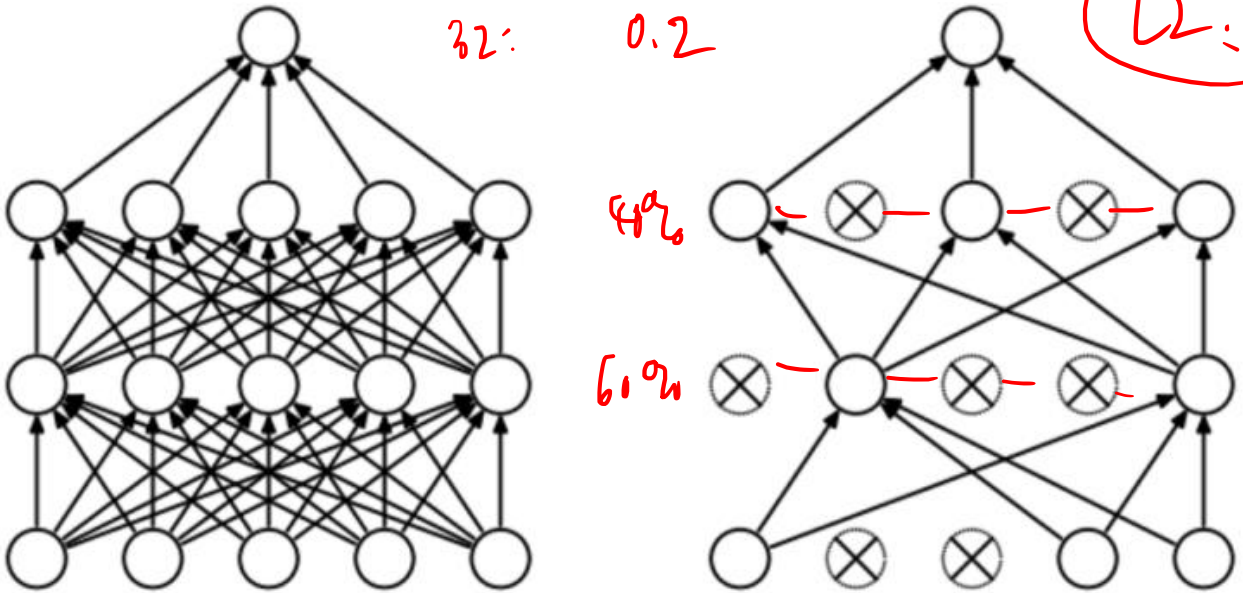
124 .. 0.5

↓

32: 0.2

ensemble

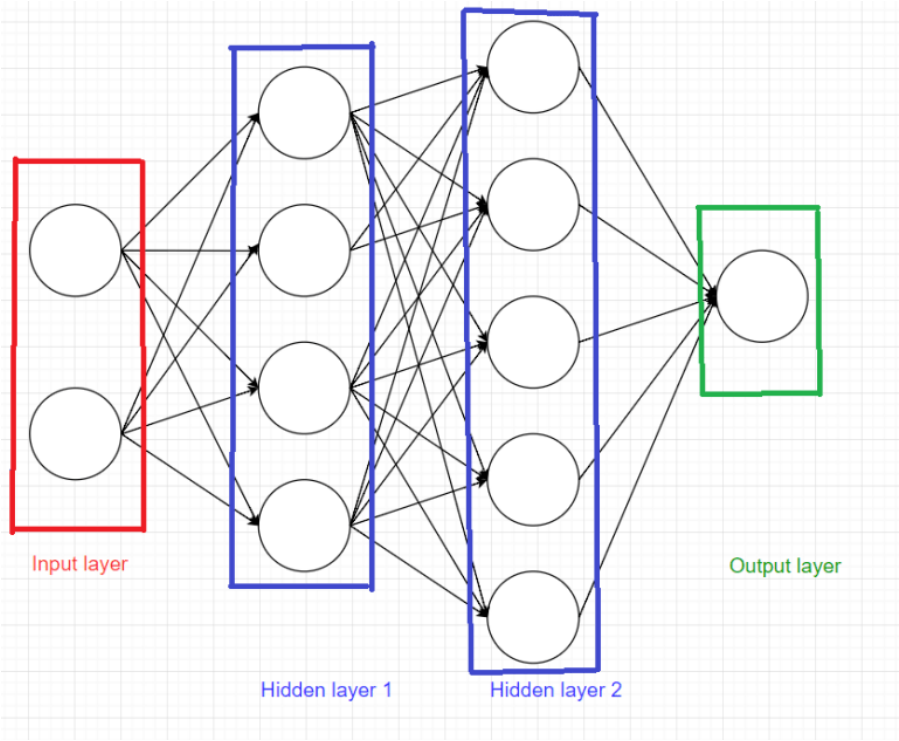
L2:



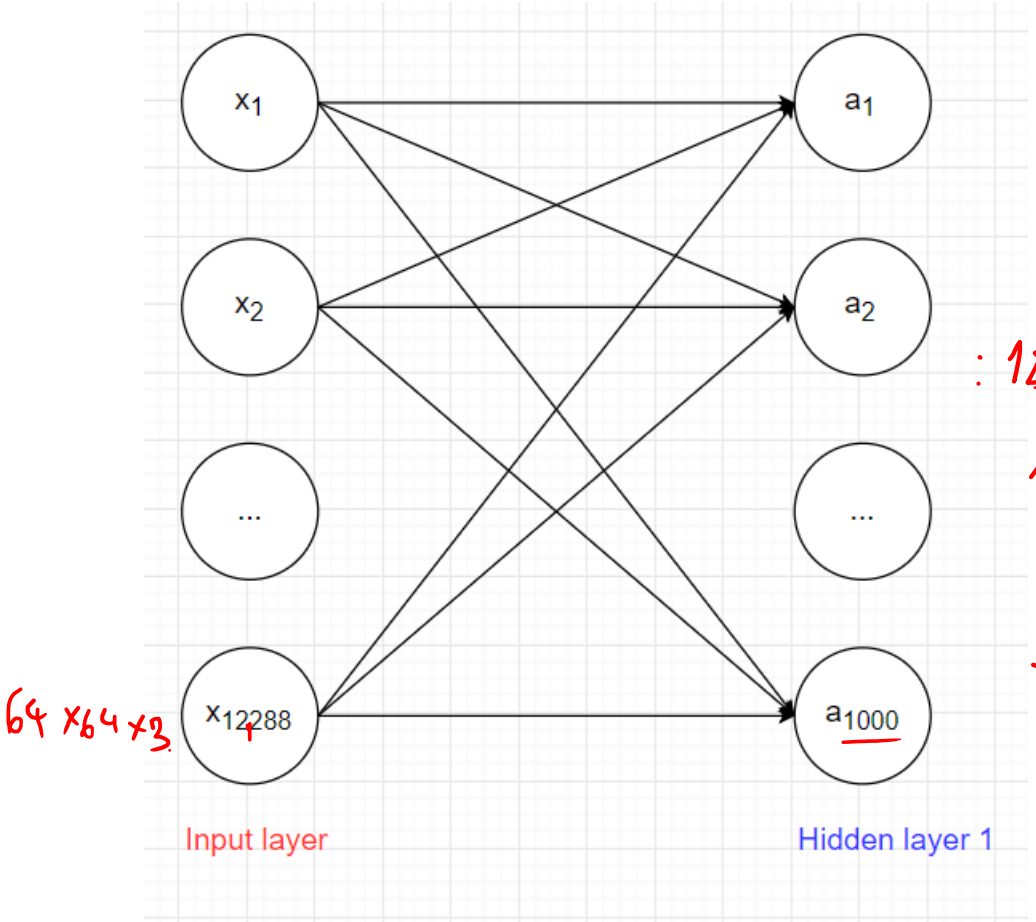
(a) Standard Neural Net

(b) After applying dropout.

Neural Network



Problem?



: ~~12288 1~~

12288 . 1000

+ 1000

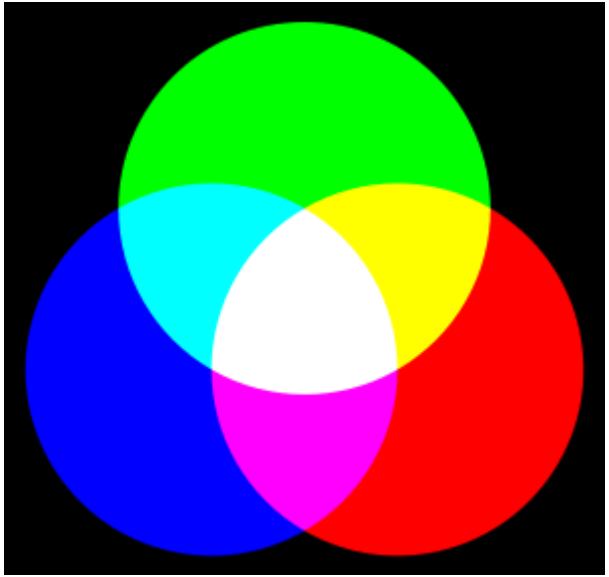
= 12,289.000

Hệ màu rgb

$(r, g, b) : 256^3$

$\downarrow$

$0 \to 255$



Ảnh màu

800.

600

Biểu diễn ảnh màu

1d : vector

2d : matrix

≥3d : tensor 3d , tensor 4d, ...

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,800} \\ w_{2,1} & w_{2,2} & \dots & w_{2,800} \\ \dots & \dots & \dots & \dots \\ w_{600,1} & w_{600,2} & \dots & w_{600,800} \end{bmatrix}$$

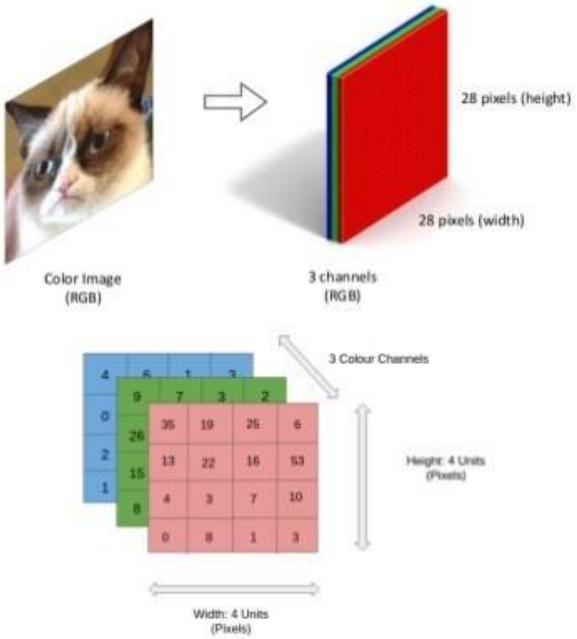
$$\begin{bmatrix} (1, 100, 10) & \dots \\ \vdots & \ddots \end{bmatrix}$$

$$\begin{bmatrix} (100, 100, 50) & (101, 112, 3) & \dots \\ & \ddots & \end{bmatrix}$$

$$\underset{R}{\begin{bmatrix} 100 & 101 & 131 \\ 150 & 10 & 111 \\ 10 & 200 & 100 \end{bmatrix}}, \underset{G}{\begin{bmatrix} 100 & 112 & 20 \\ 210 & 120 & 120 \\ 260 & 20 & 20 \end{bmatrix}}, \underset{B}{\begin{bmatrix} 50 & 3 & 80 \\ 130 & 130 & 130 \\ 30 & 30 & 3 \end{bmatrix}} \rightarrow \text{tensor 3d.}$$

Ảnh màu

color image is 3rd-order tensor



→ 32x32x3

Ảnh xám

(0-255)

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,800} \\ w_{2,1} & w_{2,2} & \dots & w_{2,800} \\ \dots & \dots & \dots & \dots \\ w_{600,1} & w_{600,2} & \dots & w_{600,800} \end{bmatrix}$$

Element-wise multiplication matrix

Ma trận A và B cùng kích thước m*n thì phép tính này cho ra ma trận C cùng kích thước m*n và C[i,j] = A[i,j] * B[i,j]. Hay là mỗi phần tử ở ma trận C bằng tích 2 phần tử tương ứng ở A và B.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} * b_{11} & a_{12} * b_{12} \\ a_{21} * b_{21} & a_{22} * b_{22} \end{bmatrix}$$

Convolutional operation

$$\otimes = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

kernel.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

X

$\otimes$

1	0	1
0	1	0
1	0	1

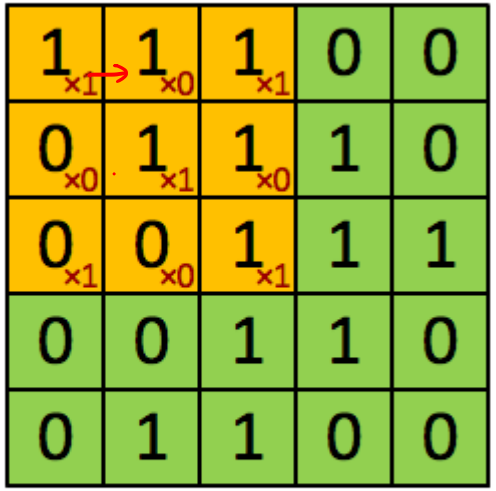
W

=

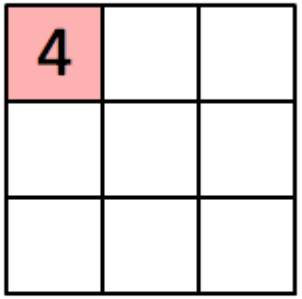
4	.	.
.		

Y

Convolutional operation

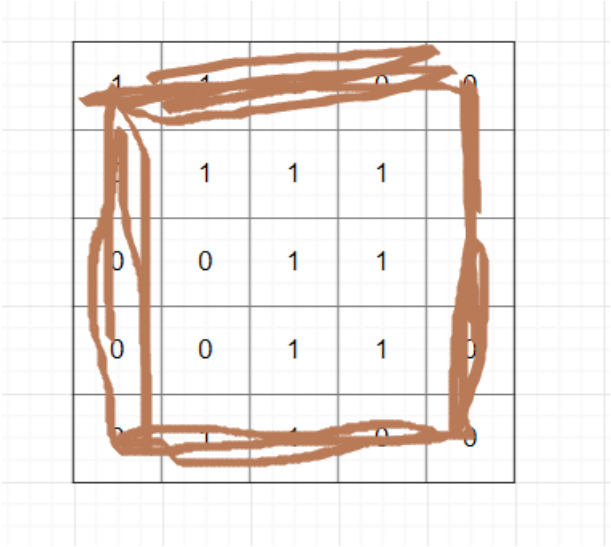


Image



Convolved Feature

Padding



Padding

size Input = size output.

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

$\circledast$ W = $\square$

5x5

Stride = 1

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

Stride = 2

X : W×10

W : 3×3

S : 2

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

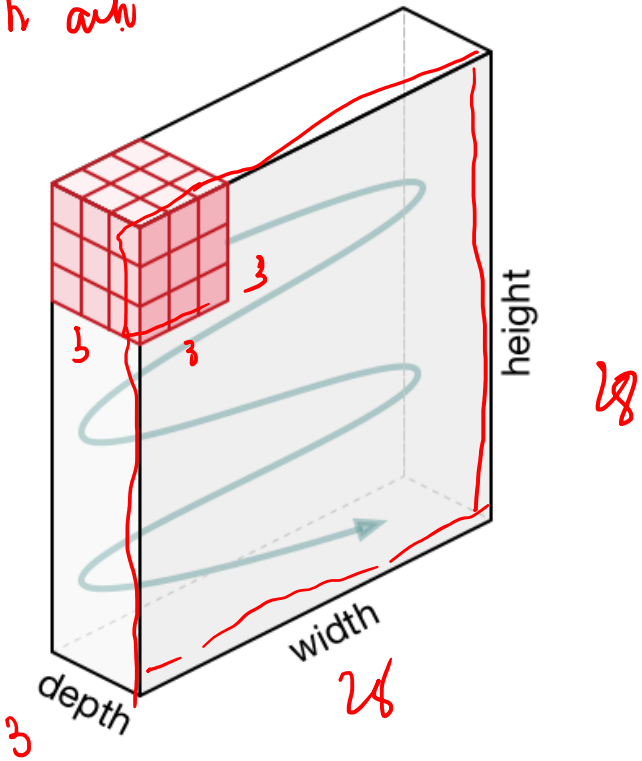
Why convolutional operation?

Operation	Kernel ω	Image result g(x,y)
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ \cdot & \cdot & \cdot \end{bmatrix}$$

Convolutional Layer

depth kernel = depth ảnh



1 kernel 3×3×3: hoặc thì tính đi

Convolutional Layer

k kernel

0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2

0	1	1
0	1	0
1	-1	1

Kernel Channel #3

$$308 + -498 + 164 + 1 = -25$$

Bias = 1

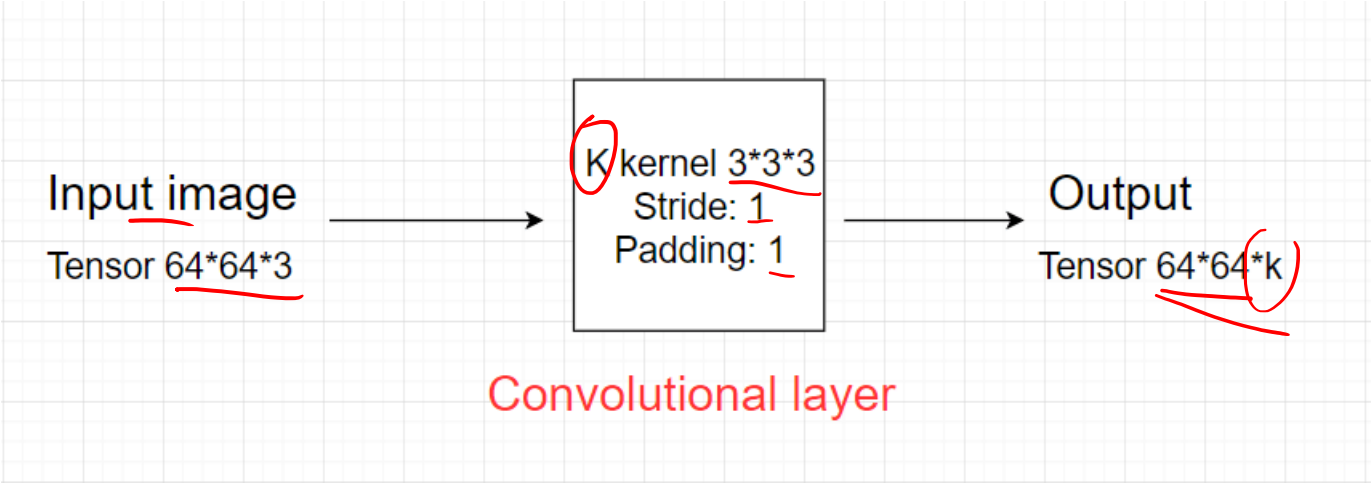
h

Output

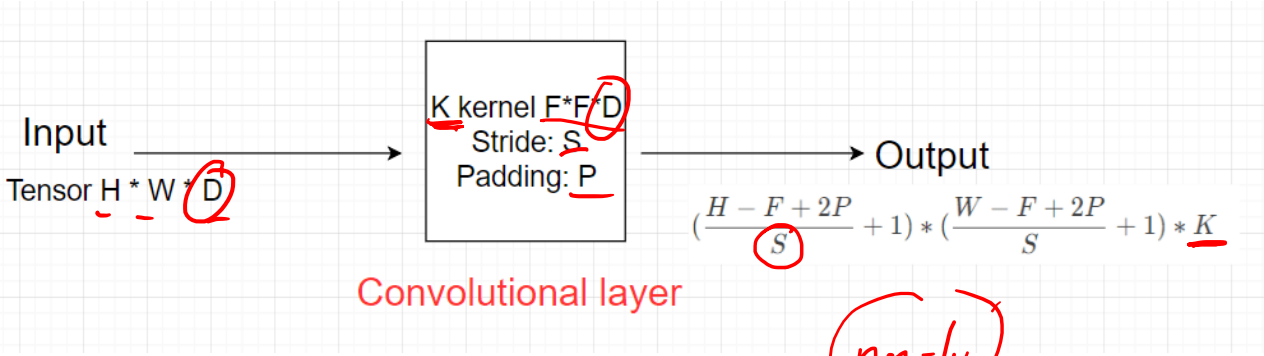
-25				...
				...
				...
				...
...	...	...	...	...

w

Convolutional Layer



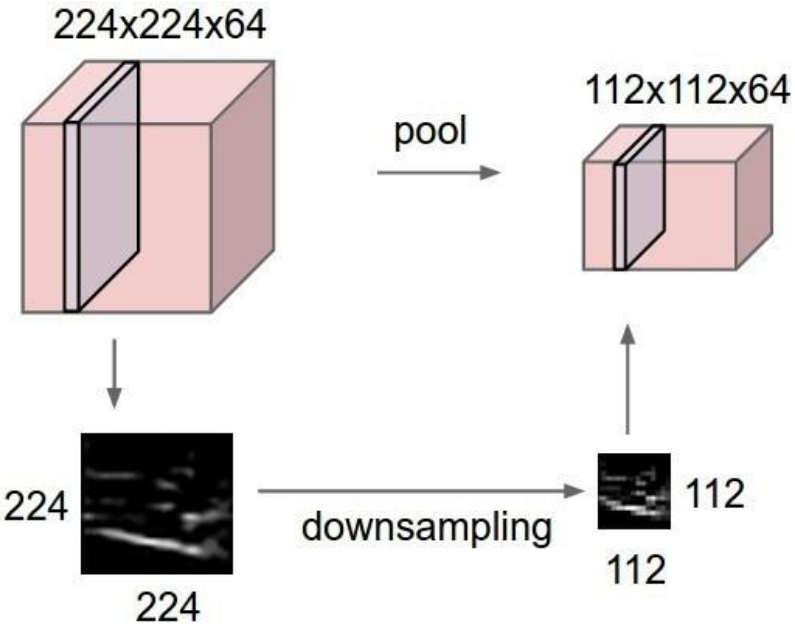
Convolutional Layer



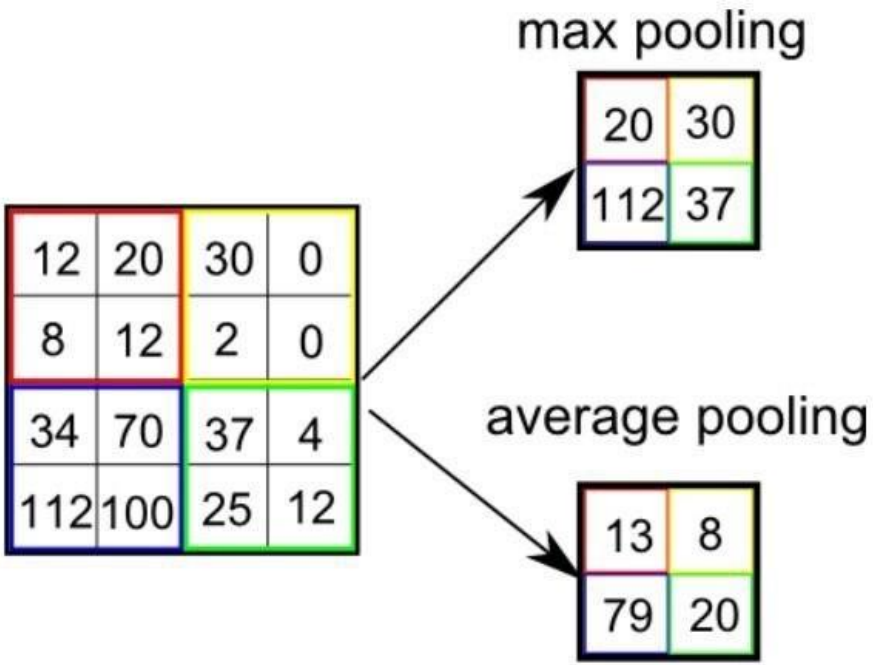
non-linear

- Output của convolutional layer sẽ qua hàm activation function trước khi trở thành input của convolutional layer tiếp theo.
- Tổng số parameter của layer: Mỗi kernel có kích thước F*F*D và có 1 hệ số bias, nên tổng parameter của 1 kernel là F*F*D + 1. Mà convolutional layer áp dụng K kernel => Tổng số parameter trong layer này là K * (F*F*D + 1).

Pooling Layer

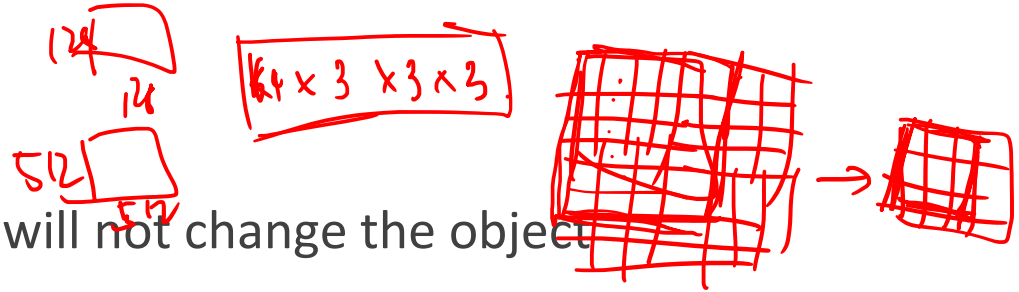


Pooling Layer



Pooling Layer

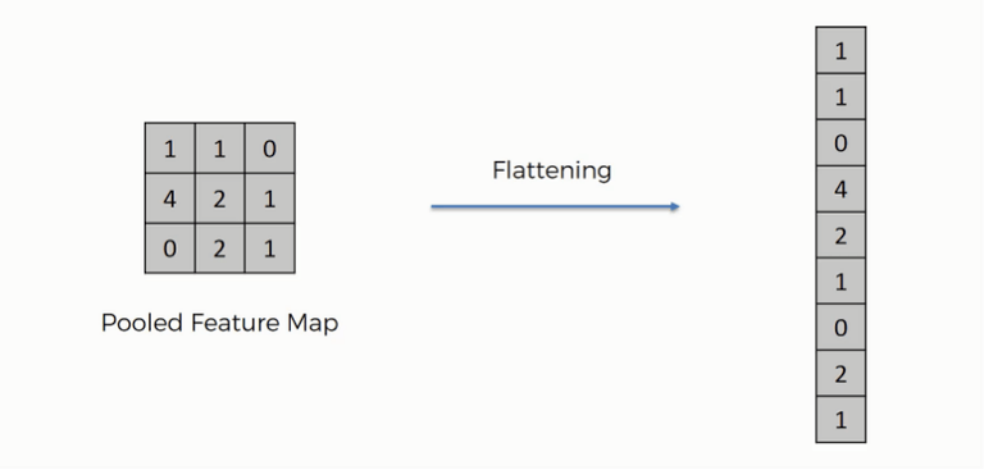
- Subsampling the pixels will not change the object



We can subsample the pixels to make image smaller

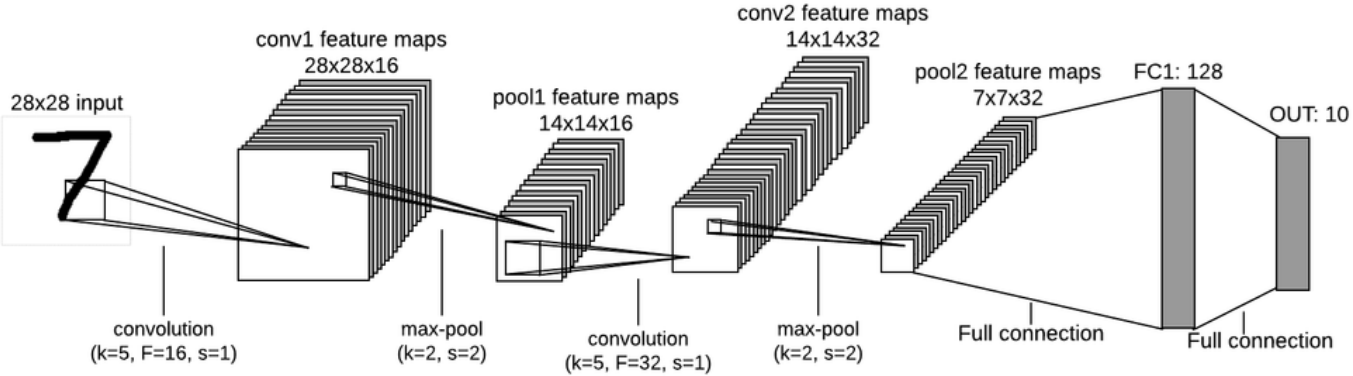
Flatten

Sau khi ảnh được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của ảnh (ví dụ mắt, mũi, khung mặt,...) thì tensor của output của layer cuối cùng, kích thước H*W*D, sẽ được chuyển về 1 vector kích thước (H*W*D)



Convolutional Neural Network

Input image -> Convolutional layer (Conv) + Pooling layer (Pool) -> Fully connected layer (FC) -> Output.

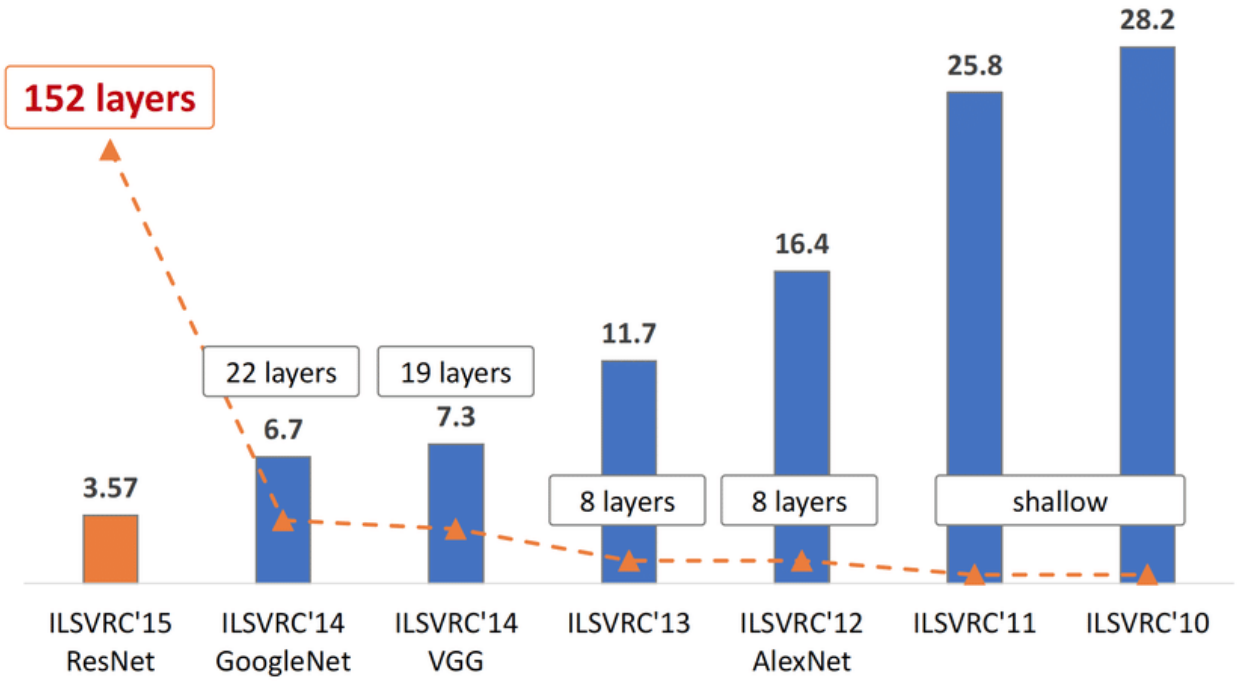


ImageNet Challenge

- ~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/

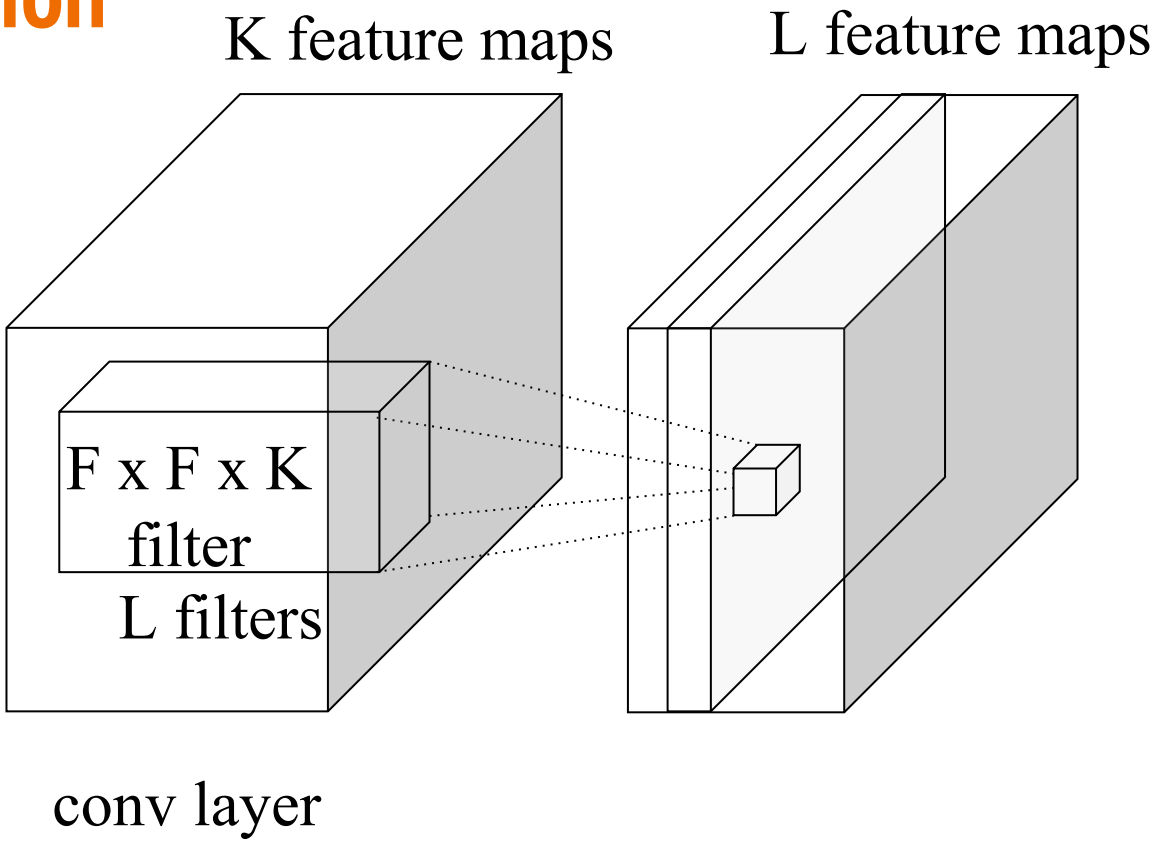
ImageNet winner



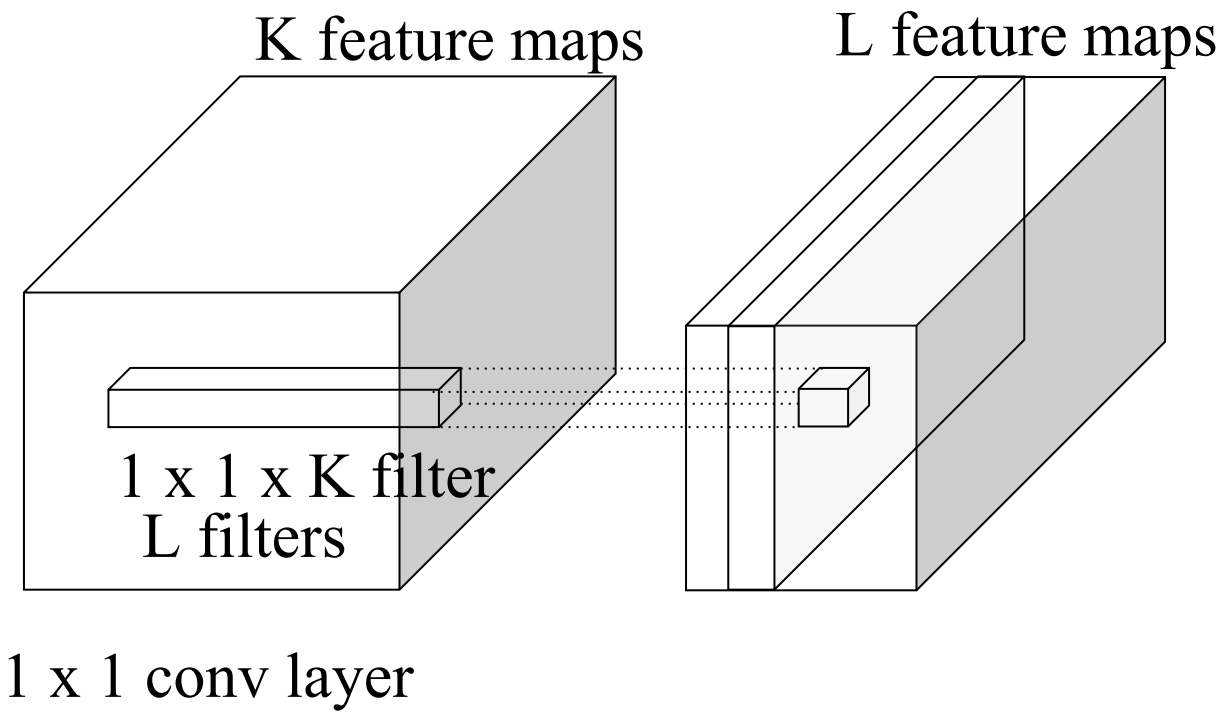
VGG 16


Size:224
3x3 conv, 64
3x3 conv, 64
pool/2
Size:112
3x3 conv, 128
3x3 conv, 128
pool/2
Size:56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool/2
Size:28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
Size:14
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
Size:7
fc 4096
fc 4096
fc 4096

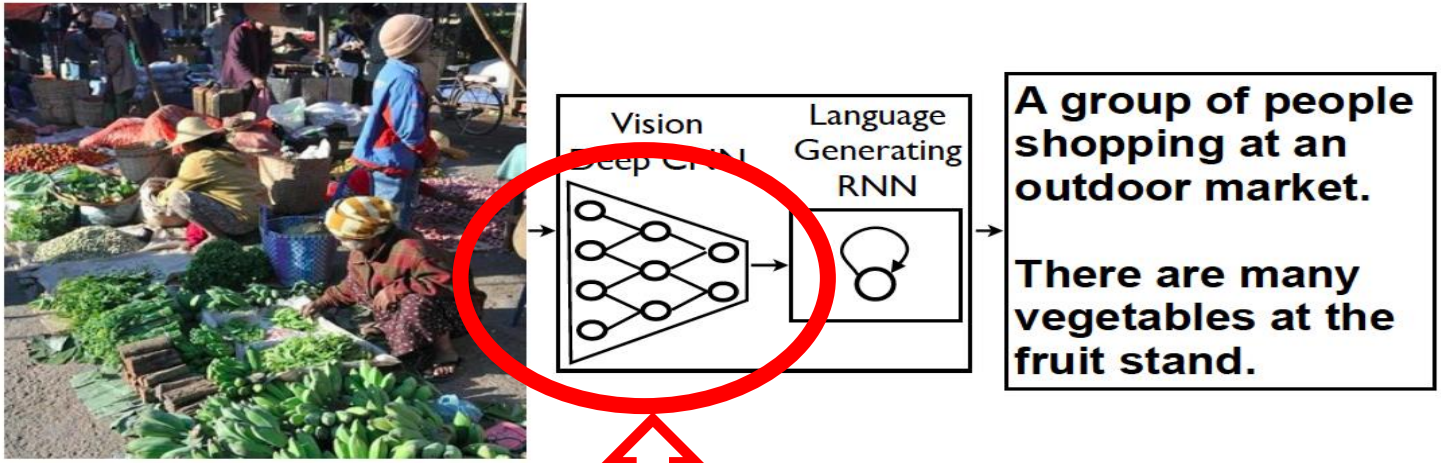
1x1 convolution



1x1 convolution



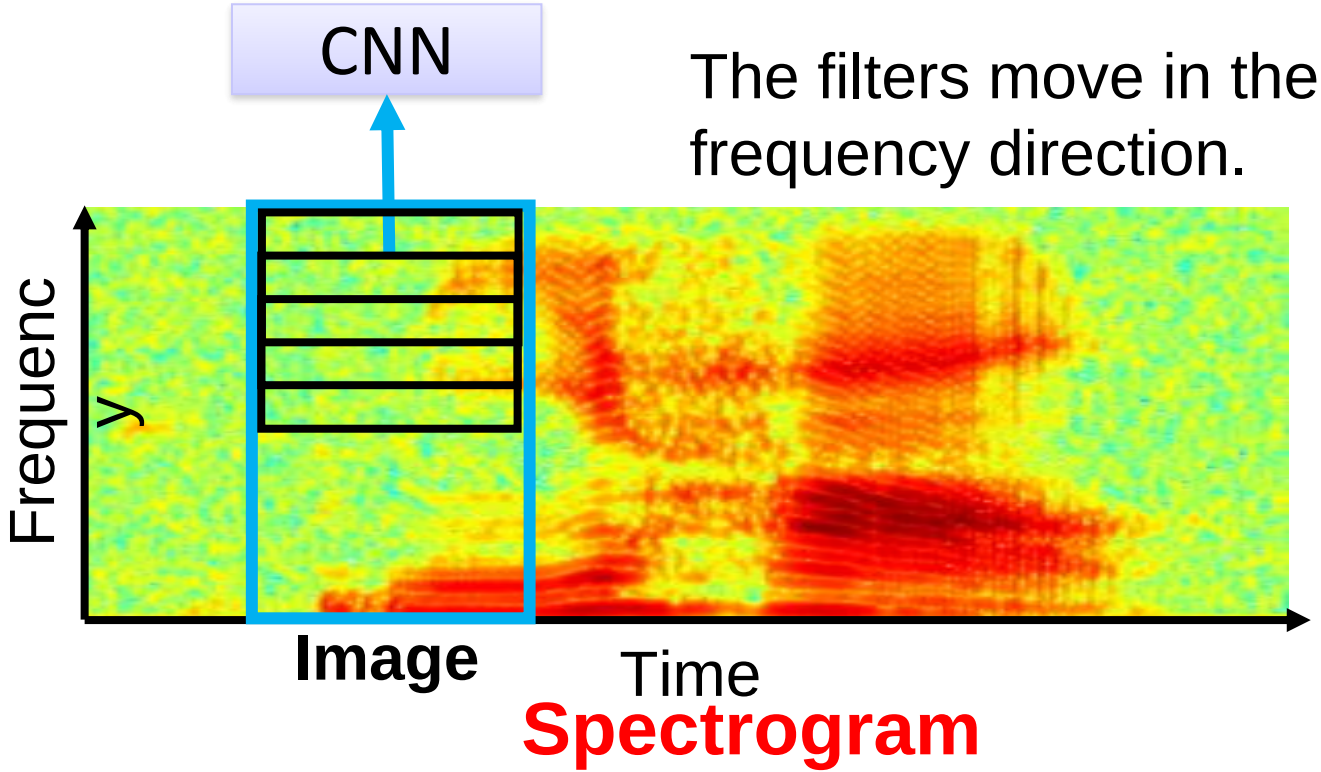
CNNs for image captioning



FC vectors from pre-trained network

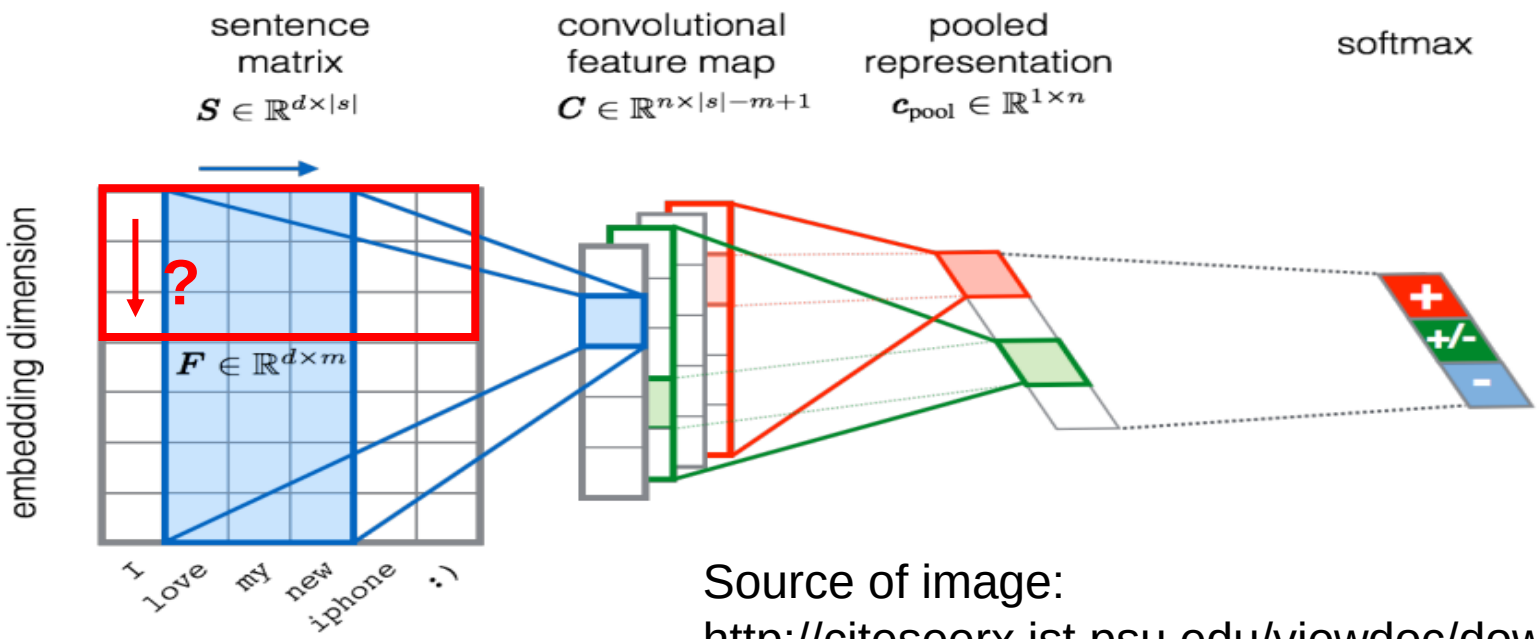
O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. CVPR 2015

CNN in speech recognition



The filters move in the frequency direction.

CNN in text classification



Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

Q&A


@LateNightSeth
THANK YOU!