BITS F464: Machine Learning

37 Convolution Neural Network (CNN)

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

April 28, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/ml

CNN Motivation

Suppose you have a noisy sensor to have a measurement

How could you estimate the actual reading

1. Average of some readings
2. in local neighborhood
3. What about weighted average?

$$s_t = \sum_{i=0}^{\infty} x_{t-i} w_{-i}$$

	w_{-6}	w_{-5}	w_{-4}	w_{-3}	w_{-2}	w_{-1}	w_0
W	0.01	0.01	0.02	0.02	0.04	0.4	0.5

X	1.00	1.10	1.20	1.40	1.70	1.80	1.90	2.10	2.20	2.40	2.50	2.70
S							1.80					

$$s_6 = x_6 w_0 + x_5 w_{-1} + x_4 w_{-2} + x_3 w_{-3} + x_2 w_{-4} + x_1 w_{-5} + x_0 w_{-6}$$

Here things are in one dimension only.

Image is a 2D signal

$$S_{ij} = (I * K)_{ij} = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} I_{i+a,j+b} K_{a,b}$$

Input

a	b	c	d
e	f	g	h
i	j	k	ℓ

Kernel

w	x
y	z

Output

aw+bx+ey+fz	bw+cx+fy+gz	cw+dx+gy+hz
ew+fx+iy+jz	fw+gx+jy+kz	gw+hx+ky+ℓz

General formula

$$S_{ij} = (I * K)_{ij} = \sum_{a=\lfloor -\frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b=\lfloor -\frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a,j-b} K_{\frac{m}{2}+a,\frac{n}{2}+b}$$

Image is a 2D signal

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

blurs the image

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$$

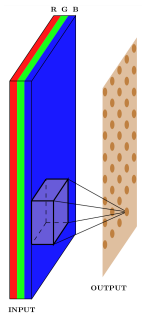
sharpens the image

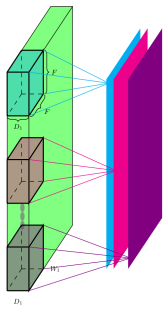
$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

detects the edges

Filter in 3D

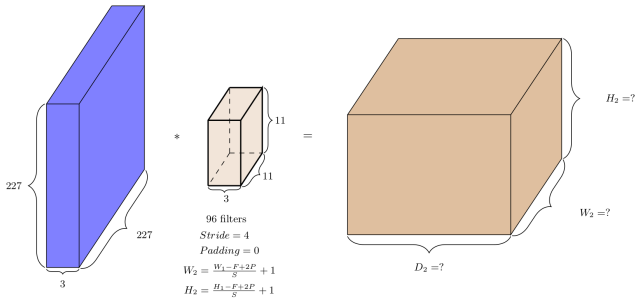
- It would refer to volume. Assume the filter extends to the depth
- Output is 2D





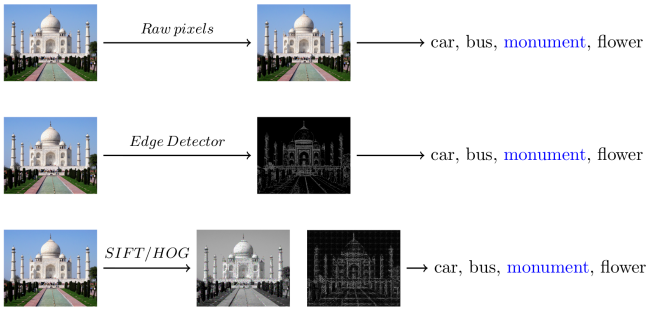
- Apply multiple filters to get multiple feature maps

Filters, Padding and Stride



$$H, W = \frac{H - F + 2P}{S} + 1, \frac{S - F + 2P}{S} + 1$$

Classification Pipeline

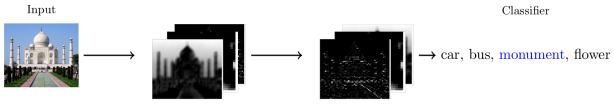


- Where is learning? (hand craft features, then learn weights for classification). One can see feature as a convolution.

Automate feature kernel discovery

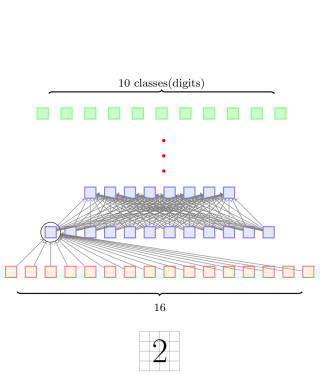
Instead of handcrafted kernels, learn filters

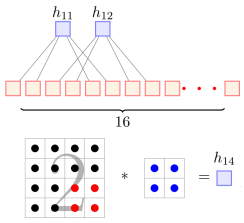
(Why not multiple?) learn multiple layers of kernels/filters



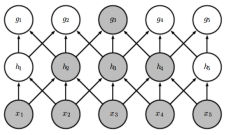
Treating these kernels as parameters and learning them.

CNN has sparse connectivity with respect to NN



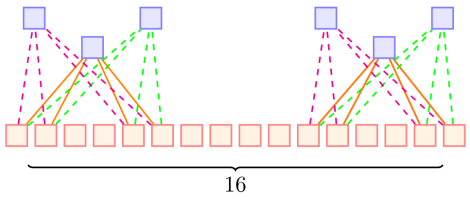


Interactions are preserved, even with reduced model parameters,



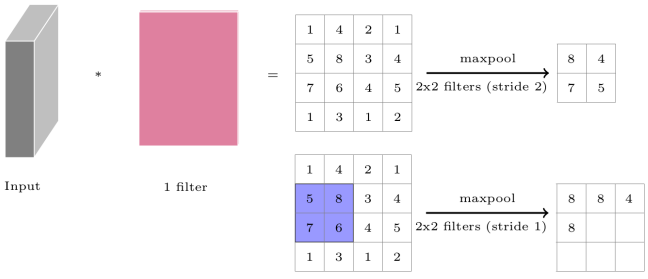
Weight sharing in CNN

One place you are extracting edge other place something else? So we do not want the kernel to be different for different portions of the image.



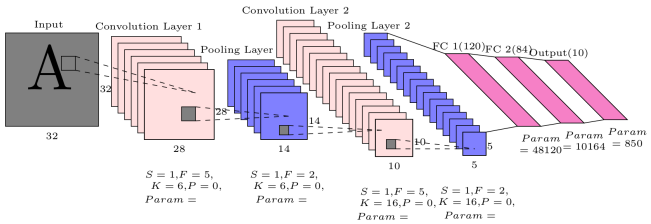
- Weight sharing in CNN makes the job of learning weights easier
- Multiple kernels help get different feature at the same level

Pooling (max, min, average)



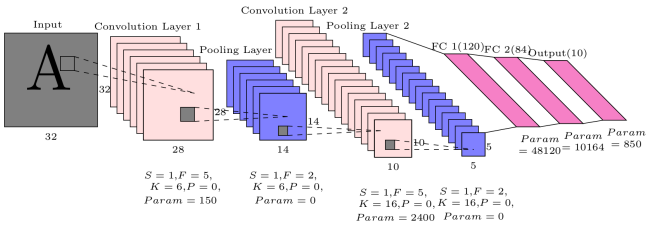
Training CNN? backpropagation!

Case-Study: LeNet-5 for handwritten character recognition [1]



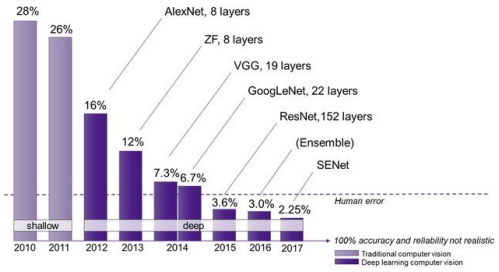
[1] Yann Lecun and Leon Bottou and Yoshua Bengio and Patrick Haffner, *Gradient-based learning applied to document recognition*, pp 2278–2324, IEEE-1998

LeNet-5 for handwritten character recognition [2]



[2] Yann Lecun and Leon Bottou and Yoshua Bengio and Patrick Haffner, *Gradient-based learning applied to document recognition*, pp 2278–2324, IEEE-1998

ImageNet ILSVRC[3]



- (2009) 22K category, 14M images
- Challenge 1000 class, 1431167 images
- HoG, LBP, SVM ...

[3] Imagenet large scale visual recognition challenge http://www.image-net.org/challenges/LSVRC/

AlexNet [4]

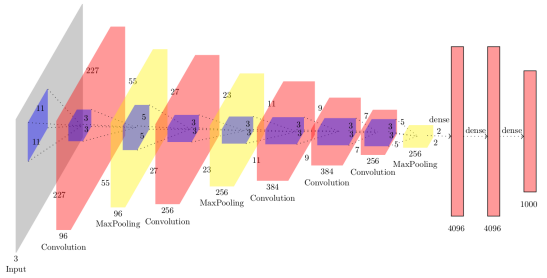


Image size 227 × 227 × 3 → [96 F=11, S=4, P=0] → [MaxPool F=3, S=2] → [256 F=5, S=1, P=0] → [MaxPool F=3, S=2] → [384 F=3, S=1, P=0] → [384 F=3, S=1, P=0] → [256 F=3, S=1, P=0] → [MaxPool F=3, S=2] → FC 4096 → FC 4096 → FC 1000

27.55 M parameters. Convolution is passes through ReLU activation.

[4] Cite: 50086, **Imagenet classification with deep convolutional neural networks**, Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E, In: Advances in neural information processing systems pages 1097–1105, NIPS-2012

ZFNet [5]

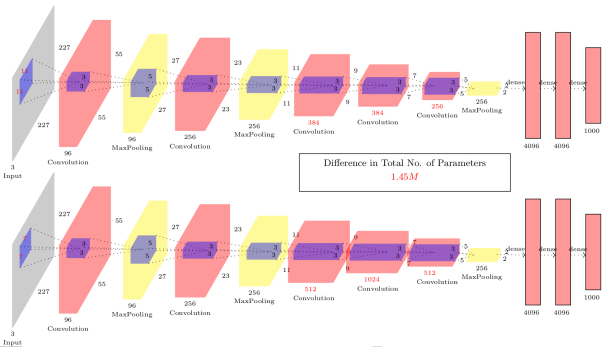
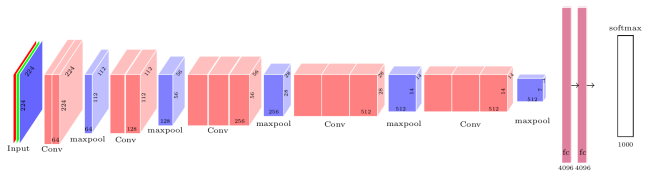


Image size 227 × 227 × 3 → [69 F=7, S=4, P=0] → [MaxPool F=3, S=2] → [256 F=5, S=1, P=0] → [MaxPool F=3, S=2] → [512 F=3, S=1, P=0] → [1024 F=3, S=1, P=0] → [512 F=3, S=1, P=0] → [MaxPool F=3, S=2] → FC 4096 → FC 4096 → FC 1000

[5] Cite 7560 **Visualizing and understanding convolutional networks**, Zeiler, Matthew D and Fergus, Rob, pages 818–833, European conference on computer vision (ECCV) Springer-2014

VGG16 [6]

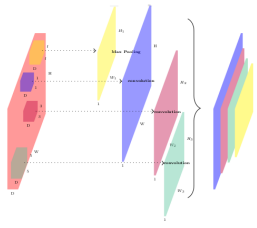


1. Kernel size is always 3 × 3
2. 16M parameters in pre-FC and 122 in FC. First FC layer is huge
3. Layers represents abstract representation and can be reused (FC or Conv)

[6] Cite 29548 **Very deep convolutional networks for large-scale image recognition**, Simonyan, Karen and Zisserman, Andrew, pages 818–833, arXiv preprint arXiv(2014):1409.1556 - In: ICLR-2015

GoogLeNet [7]

- Recall scale invariance in SIFT
- Multiple filters of different size is a good idea
- With $W \times H \times D$ input and $F \times F \times D$ Filter and $S = 1$ and no padding, output is of size $(W - F + 1) \times (H - F + 1)$
- Each value needs $F \times F \times D$ computation

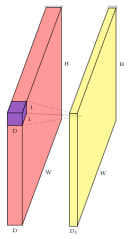


Can we reduce this computation a bit?

Idea is to have 1 × 1 computation

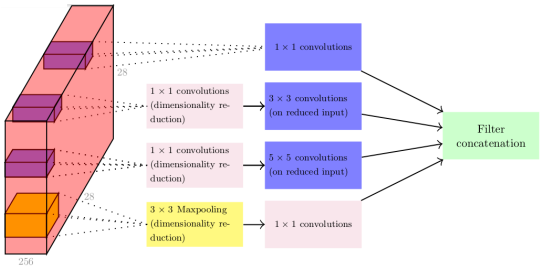
[7] Cite: 17022, **Going deeper with convolutions**, Szegedy, Christian and Liu, Wei and Jia, Yangqing and Sermanet, Pierre and Reed, Scott and Anguelov, Dragomir and Erhan, Dumitru and Vanhoucke, Vincent and Rabinovich, Andrew, In: Proceedings of the IEEE conference on computer vision and pattern recognition pages 1–9, CVPR-2015

1×1 convolution



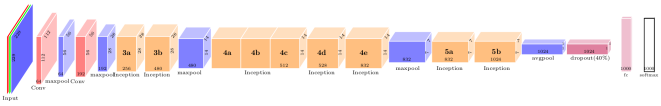
- 1×1 is $1 \times 1 \times D$
- They produce one output place
- By using D_1 such 1×1 convolution output becomes $F \times F \times D_1$
- We have $D_1 < D$

Inception Block: Multiple convolutions



1. 1×1 convolution
2. 1×1 convolution followed by 3×3
3. 1×1 convolution followed by 5×5
4. 3×3 maxpool followed by 1×1
5. Appropriate padding is done to make things of same size

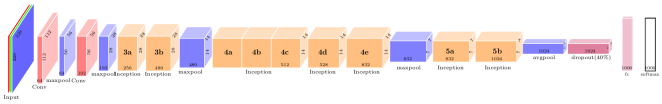
GoogLeNet



1. Input is RGB 229×229
2. Each inception module have very specific configuration.

(3a)	$192 \times 28 \times 28$	64	96	128	16	32	32
(3b)	$256 \times 28 \times 28$	28	128	192	32	96	61
(4a)	$48 \times 14 \times 14$	192	96	208	16	48	96
(4b)	$512 \times 14 \times 14$	160	112	224	24	64	64
(4c)	$512 \times 14 \times 14$	128	128	256	24	64	64
(4d)	$512 \times 14 \times 14$	112	144	228	32	64	64
(4e)	$528 \times 14 \times 14$	256	160	320	32	128	128
(5a)	$832 \times 7 \times 7$	256	160	320	32	128	128
(5b)	$832 \times 7 \times 7$	384	192	384	48	124	128

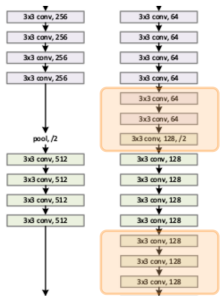
GoogLeNet



- VGGNET has $512 \times 7 \times 7$ size at pre-FC this was an issue to connect with 4096
- GoogLeNet applies a average pool. Gives 49 time reduction. has 1024 values only
- Dropout and connect to 1000
- 12 times less connections as compared to AlexNet
- 2 times more computation as compared to AlexNet
- Very high accuracy. Error reduced from 16% -to- 6.7%

ResNet [8]

If a shallow neural network works well. What would happen if we add more layers?

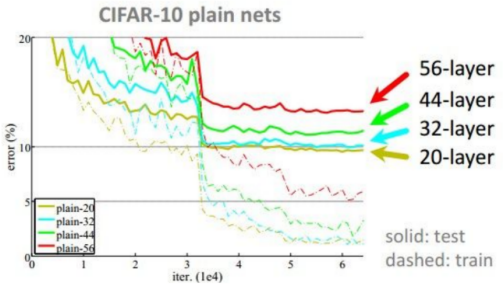


- Deep network should also work well (It would learn identity in new layers)

[8] Cite: 32871, **Deep residual learning for image recognition**, He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, In: IEEE conference on computer vision and pattern recognition, pages 770–778, CVPR-2016

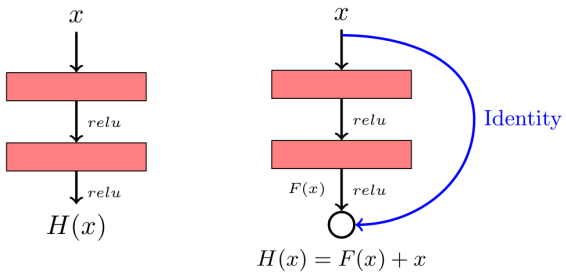
ResNet

But, in practice it was not happening

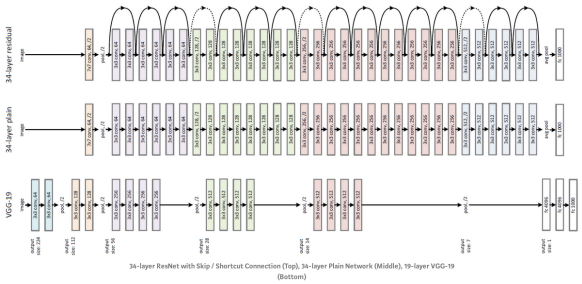


Why? Identity is one of the solution in large domain.

Let me tell this to the network



ResNet Comparison [9]



152-layer deep net was better than human. Only 3.6% error rate ImageNet Classification [a]

[a] Better than the 2nd best system **ImageNet Detection:** 16% **ImageNet Localization:** 27% **COCO Detection:** 11% **COCO Segmentation:** 12%

[9] ResNet Winner of ILSVRC 2015 (Image Classification, Localization, Detection) Sik-Ho Tsang

ResNet Hyper-parameters and Issues

- Training takes huge time
- Batch Normalization
- Zavier/2 initialization
- SGD and momentum
- Small learning rate 0.1
- Mini-batch size 256
- Weight decay
- No Dropout

Thank You!

Thank you very much for your attention!