# Module 33 : CNN architectures

→ What kind of tasks are CNNs used for ? CNNs are generally applied to image related tasks.

(1) Classification : Predicting which class the image belongs to.

↓

Popular dataset : Imagenet (1000 categories)

↓

the bigger version (10000 categories)

↓

iconic photos : the class related object covers most space in the picture.

(2) classification and Localization : Predicting the class, along with the location of object of the photo.

↓

The localization problem is a regression problem.

(3) Object detection : multiple objects in the photo.

The classification and localization for each object

(4) Instance segmentation : Contour of the object, rather than a box.

→ The decisions to be made for using CNN for image based tasks

i) No. of layers

ii) No. of filters in each layer

iii) filter size

iv) max pooling → when to use.

↓

Not practical to try all permutations, rather the accepted practice is to use standard architectures which are known to give good results for a wide range of tasks.

→ The imagenet challenge : 1000 classes, with 1000 images each

| | | |
|---|---|---|
| shallow | 2010 | → top 5 error rate : 28.2%. |
| shallow | 2011 | → 25.8%. |
| 8 | 2012 | → Alexnet → 16.4%. (drastic ↓) |
| 8 | 2013 | → ZFNet → 11.7%. |
| 19 | 2014 | → VGG → 7.3%. |
| 22 | 2014 | → Googlenet → 6.7%. |
| 152 | 2015 | → Resnet → 3.57%. |

→ Alexnet : inputs : 227 * 227 * 3 → channels

$\frac{W_I + 2P - F}{S} + 1$ conv. ReLU

↓ K = 96, f = 11, S = 4, P = 0

parameters = 11 * 11 * 3 * 96 (34848)

↓ benefit of taking many filters initially is the ability to capture different dynamics of the image.

55 * 55 * 96

↓ maxpooling

f = 3, S = 2,
parameters = 0

27 * 27 * 96

conv ↓ K = 256, f = 5, S = 1, P = 0, parameters = 96 * 5 * 5 * 256

23 * 23 * 256

f=3, s=2 ↓ maxpooling

11 * 11 * 256

maxpooling are not counted as layer, no parameters.

conv ↓ K=384, f=3, s=1, p=0,
parameters = 384 * 3 * 3 * 256

9 * 9 * 384

conv ↓ K=384, f=3, s=1, p=0
parameters = 384 * 3 * 3 * 384

7 * 7 * 384

conv ↓ K=256, f=3, s=2, p=0
parameters = 384 * 3 * 3 * 256

5 * 5 * 256

maxpooling ↓ f=3, s=2

2 * 2 * 256

flatten ↓

4096

dense ↓

4096

dense ↓

1000 (softmax)

total parameters = 27.55M

→ ZFNet : 8 layers again, can be understood using AlexNet, but differs at no. of filters, sizes, etc at some points

↓ (fewer parameters)

Layer 1 : f = 11 → 7, Difference in parameters = 20.7K

$W_0$ = 56, scaled to 55 (55 * 55 * 96, same as AlexNet)

↓

Layer 2 : No difference (Maxpooling)

↓

Layer 3 : No difference

↓

Layer 4 : No difference

↓

Layer 5 : K = 384 → 512

Difference in parameters = 0.29 M

(Increase in parameters)

↓

Layer 6 : K = 384 → 1024

Difference in parameters = 0.8 M (+)

↓

Layer 7 : K = 256 → 512

Difference in parameters = 0.36 M (+)

↓

Layer 8 : No difference

↓

Layer 9 : No difference

↓

Layer 10 : No difference

↓

Layer 11 : No difference

→ VGGNet (Replaced the notion of needing alternate conv and pooling layers, rather combining conv. layers and pooling layers)

↓

filters : 3 * 3

1) Back to back conv of : 64 * 3 * 3 (maintaining the dimensions along length and width)

2) maxpooling (1/2)

3) Back to back conv of : 128 * 3 * 3

4) maxpooling

5) 3 conv layers : 256 * 3 * 3

6) maxpooling

7) 3 conv layers : 512 * 3 * 3

8) maxpooling

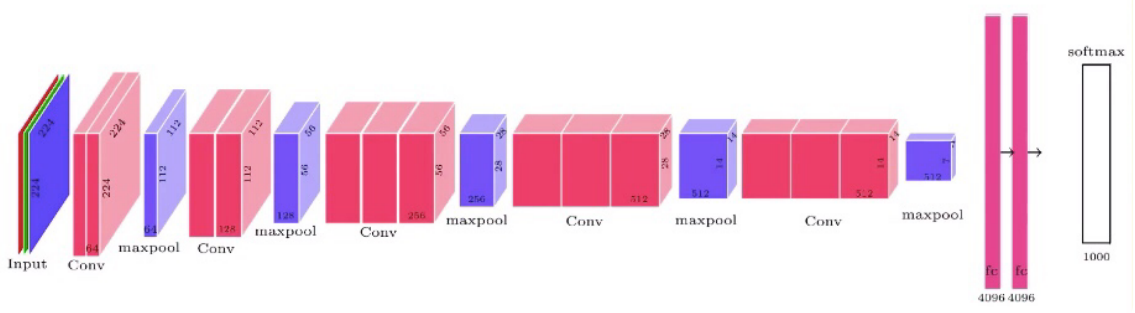
9) 3 conv layers : 512 * 3 * 3

10) maxpooling

11) 2 fully connected layers (4096)

12) O/p layer (softmax)

VGG16




total parameters in non fc layers = ~16 M

total parameters in fc layers = ~122 M

most parameters are in the first fc layer (~102 M)