9, 10, 11

CNN + RNN

Convolutional Neural Network

} Classification
Regression

Signal → Image - 2D data x-y plane

Convolutional Neural Networks (CNNs) are Artificial Intelligence algorithms based on multi-layer neural networks that learns relevant features from images, being capable of performing several tasks like object classification, detection, and segmentation.

Image Classification

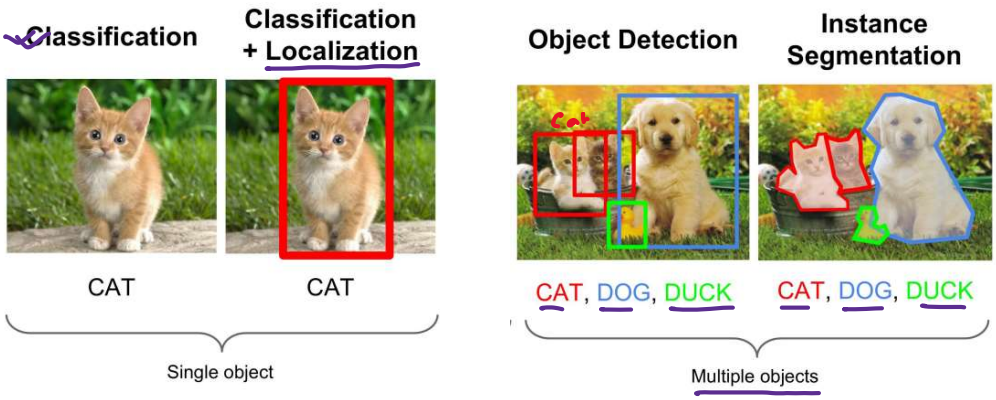


Image - Signal

$f(x) * g(x') \rightarrow$ addition & multiplication

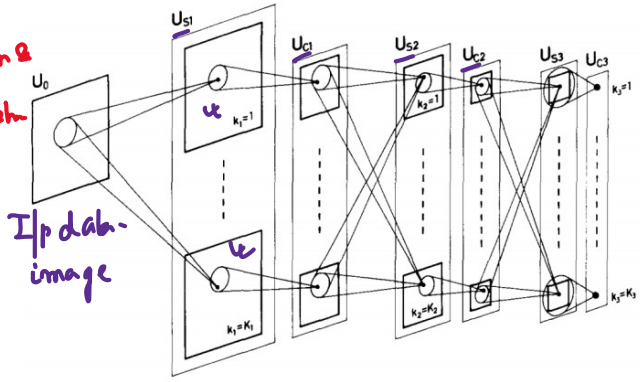
1) U_S = Sampling
2) U_C = Convolutional

Signal processing

Neorecognitron

1980

Kunihiko Fukushima



} Mag, Phase, Energy

1) Time Domain
2) Frequency Domain

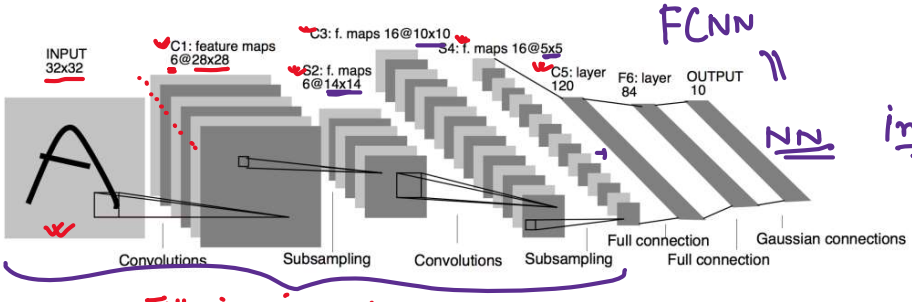
The modern concept of Convolutional Neural Networks comes from the work of Yann LeCun published in 1998. LeCun proposed a CNN called LeNet for hand-write recognition.

0 to 9

FCNN

0, 1, 2, 3, ... 9

called LeNet for hand-write recognition.



FCNN

0, 1, 2, 3, ... 9

NN

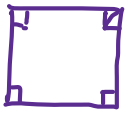
Filtering input image

Image processing

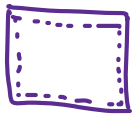
MATLAB

Manually selected filter

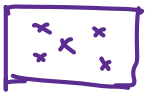
image



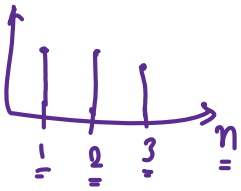
C_1



C_2



C_3



1D (continuous, discrete) :

Input

Kernel

$$f * g\ (x) = \int_{\alpha=-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$

$$= \sum_{\alpha=0}^{N-1} f(\alpha) g(x-\alpha)$$

↳ matrix filtering

Output is sometimes called Feature map

2D (continuous, discrete) :

$$f * g\ (x,y) = \int_{\alpha=-\infty}^{\infty} \int_{\beta=-\infty}^{\infty} f(\alpha,\beta) g(x-\alpha, y-\beta) d\alpha d\beta$$

$$= \sum_{\alpha=0}^{N-1} \sum_{\beta=0}^{N-1} f(\alpha,\beta) g(x-\alpha, y-\beta)$$

Signal processing.

Convolution Properties

- ✓ Commutative:

  f*g = g*f

  $f * g = g * f$

- ✓ Associative:

  (f*g)*h = f*(g*h)

  $f(\alpha)\, g(x-\alpha) = g(\alpha)\, f(x-\alpha)$

- ✓ Homogeneous:

  f*(λg)= λ f*g

- ✓ Additive (Distributive):

  f*(g+h)= f*g+f*h

face recognition

ML 7/10 SVM

DL 9/10 CNN

Why CNN is better?

The advantage of CNNs over others classification algorithms (SVM, K-NN, Random-Forest, and others) is that the CNNs learns the best features to represent the objects in the images and has a high generalization capacity, being able to precisely classify new examples

K-NN, Random-Forest, and others) is that the CNNs learns the best features to represent the objects in the images and has a high generalization capacity, being able to precisely classify new examples with just a few examples in the training set.

A CNN is topically composed by four types of layers:

- Convolutional
- Pooling → downsample the f-map ↓ reduces the overfitting
- Relu (Activation layer)
- Fully Connected (FCNN)

- A convolutional layer is composed by a set of filters, also called kernels, that slides over the input data.
- Each kernel has a width, a height and (width × height) weights utilized to extract features from the input data.
- In the training step, the weights in the kernel starts with random values, and will be learning based on the training set.

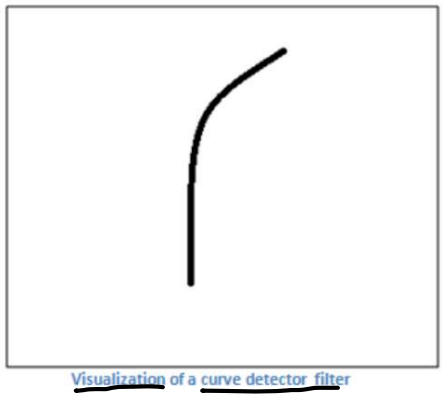
5 x 5 – Image Matrix

3 x 3 – Filter Matrix

Each filter in the convolutional layer represents a feature.

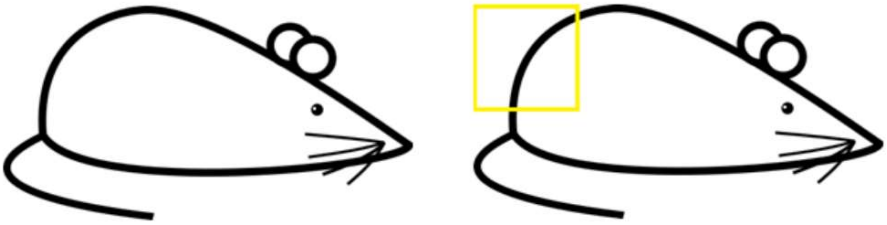
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



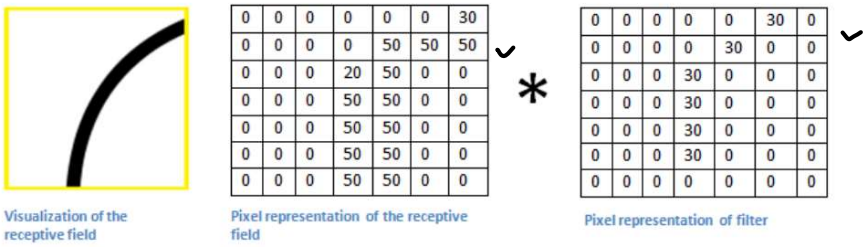
Visualization of a curve detector filter

When the filter slides over the image and finds a match...



Original image | Visualization of the filter on the image

The convolution operation generates a large number, activating the filter to that characteristic.



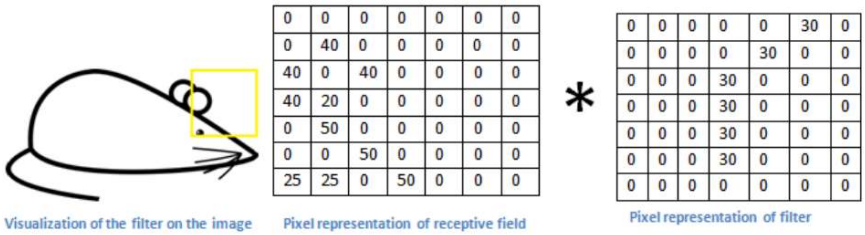
0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Visualization of the receptive field | Pixel representation of the receptive field | Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

When the filter slides over the image and finds no match, the filter does not activate.
The CNN uses this process to learn the best filters to describe the objects.



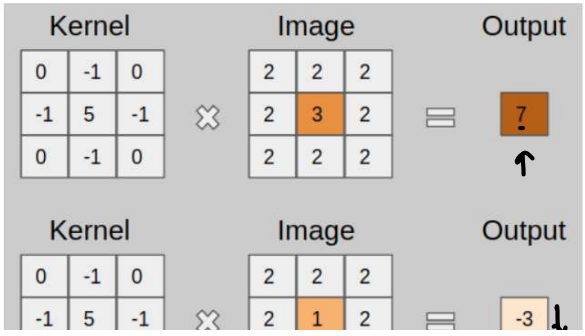
0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Visualization of the filter on the image | Pixel representation of receptive field | Pixel representation of filter

Multiplication and Summation = 0

Convolution is using a 'kernel' to extract certain 'features' from an input image.



5x3 + (2x -1x 4)

15 - 8 = 7

(5x1) + (2x -1x 4)

A kernel is a matrix, slide across the image and multiplied w/ the i/p such that the o/p is enhanced in a certain desirable

$(5\times1)+(2\times-1\times4)$

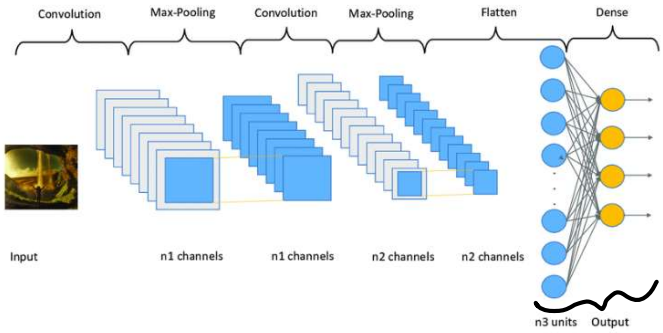
$=-3$

Kernel				Image			Output
0	-1	0	⊛	2	2	2	= -3
-1	5	-1		2	1	2	
0	-1	0		2	2	2	

enhanced in a certain desirable manner.

A convolutional neural network is a specific kind of neural network with multiple layers. It processes data that has a grid-like arrangement then extracts important features. One huge advantage of using CNNs is that you don't need to do a lot of pre-processing on images.

CNN



Different types of CNNs

1D CNN: With these, the CNN kernel moves in one direction. 1D CNNs are usually used on time-series data.

2D CNN: These kinds of CNN kernels move in two directions. You'll see these used with image labelling and processing.

3D CNN: This kind of CNN has a kernel that moves in three directions. With this type of CNN, researchers use them on 3D images like CT scans and MRIs.

- Recognize images with little preprocessing
- Recognize different hand-writing
- Computer vision applications
- Used in banking to read digits on checks
- Used in postal services to read zip codes on an envelope