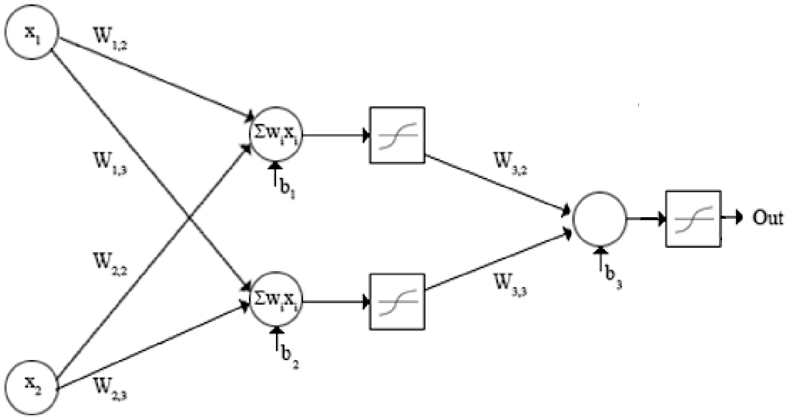
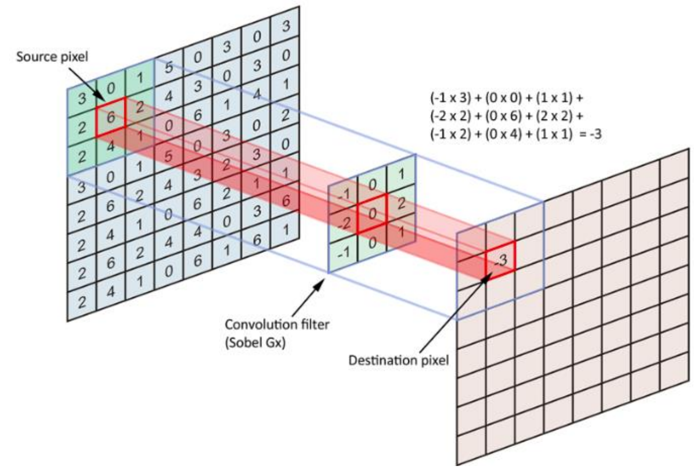
Course Outline

- TOPICS
  1. What is Machine Learning and Image Processing
  2. Traditional Features, K-NN classifier
  3. Linear Classification
  4. Perceptron Algorithm, Sigmoid Activation Function, Gradient Descent
  5. Stochastic Gradient Descent, Back-Propagation
  6. Multi-Layer Neural Network
  7. Convolution and Pooling
  8. Mid-Term Examination
  9. Mid-Term Examination
  10. Convolutional Neural Networks.
  11. Training Convolutional Neural Networks: Hyper-Parameters, Activation functions, initialization, dropout, batch normalization
  12. Recurrent Neural Networks
  13. Applications of Convolutional Neural Networks for Image Segmentation and Object Classification
  14. Project Presentations

Difference Between NN and CNN





NN:

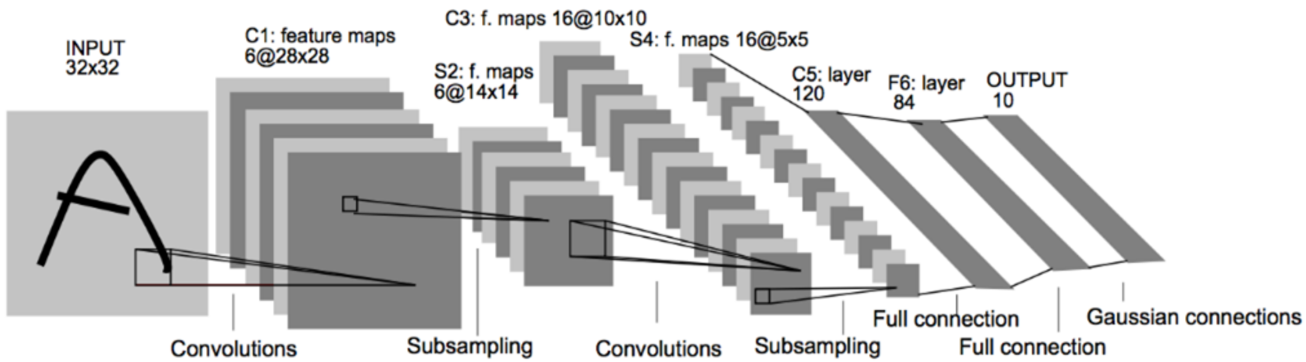
- Resemble to human brain: interconnected neurons, transmit signal.
- Input Layer, Hidden Layers, Output Layer.
- **Inputs** are in the **Array** form.
- **Filters** are in the **Array** form

CNN:

- Similar to NN but convolution exist.
- Convolutional Layer, ReLU Layer (called Activation), Pooling Layer, Fully Connected Layer.
- **Inputs** are in the **Matrix** form.
- **Filters** are in the **Matrix** form

https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050

LeNet5

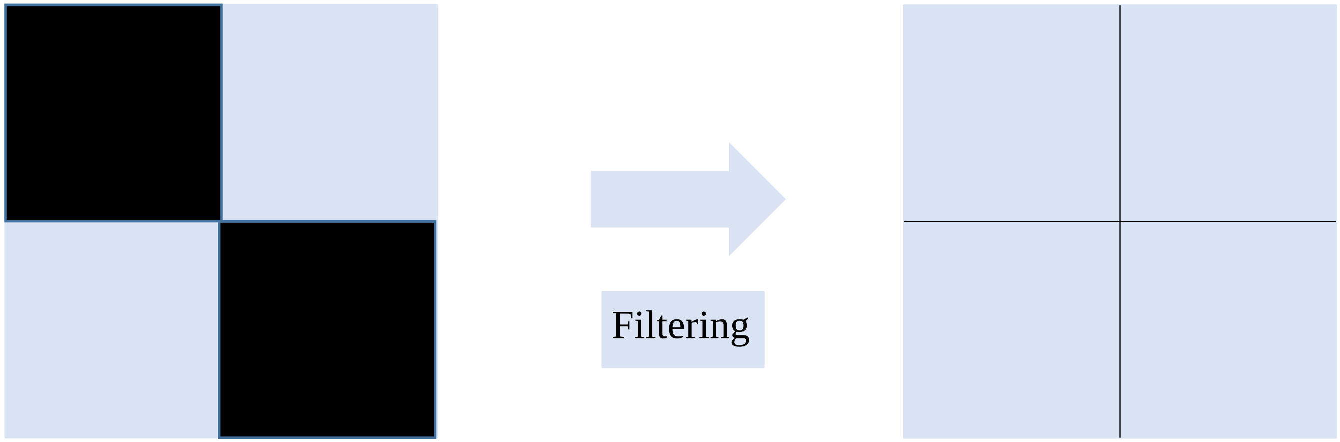


CNN called LeNet by Yann LeCun (1998)

Lenet5: The first CNN structure was proposed by Yann LeCun in 1998.

- Applied on MNIST and ASCII characters for document classification.
- Specifically, characters and digits are segmented and forwarded to CNN to determine its label.

Edge Detection



- How can we find the edges in images?
- Answer: Using derivative operation to remove redundant information (homogenous regions).
- Perform derivative operation through –x axis and –y axis.

Edge Detection

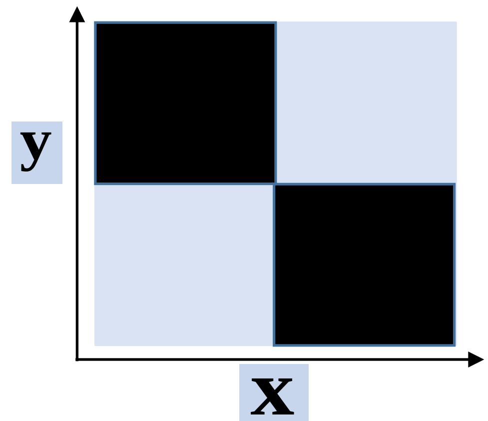


Image: I

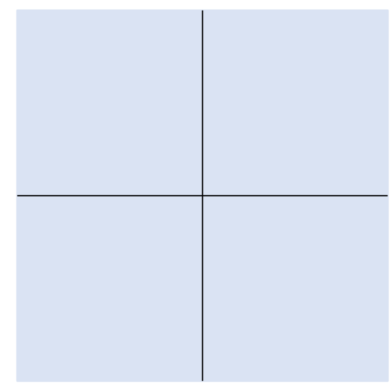
Horizantal Edges : $\dfrac{dI}{dy} \cong \dfrac{f(y_i) - f(y_{i-1})}{y_i - y_{i-1}}$

Image: I

Vertical Edges : $\dfrac{dI}{dx} \cong \dfrac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$

Image: I

EdgeMap $= \left|\dfrac{dI}{dx}\right| + \left|\dfrac{dI}{dy}\right|$



Edge Detection

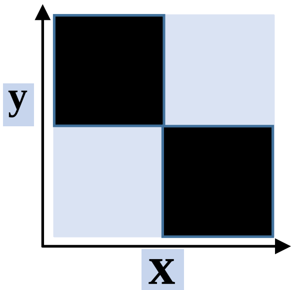


Image: I

As an alternative way,

we can convolve the image with filters.

$$\textbf{Horizantal Filter}: F_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\textbf{Horizantal Edges}: \frac{dI}{dx} \cong I * F_x \qquad *: \text{refers convolution}$$

$$\textbf{Vertical Filter}: F_y = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\textbf{Vertical Edges}: \frac{dI}{dy} \cong I * F_y \qquad *: \text{refers convolution}$$

Edge Detection

1	1	1
0	0	0
–1	–1	–1

convolve

0	0	0	20	20	20
0	0	0	20	20	20
0	0	0	20	20	20
20	20	20	0	0	0
20	20	20	0	0	0
20	20	20	0	0	0

gradient

0	0	0	0
60	20	–20	–60
60	20	–20	–60
0	0	0	0

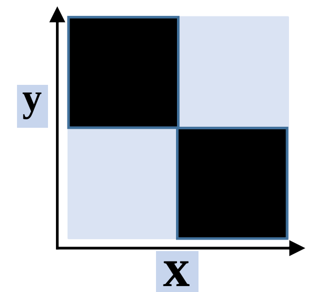
1	0	–1
1	0	–1
1	0	–1

convolve

0	0	0	20	20	20
0	0	0	20	20	20
0	0	0	20	20	20
20	20	20	0	0	0
20	20	20	0	0	0
20	20	20	0	0	0

gradient

0	60	60	0
0	20	20	0
0	–20	–20	0
0	–60	–60	0



0	0	0	20	20	20
0	0	0	20	20	20
0	0	0	20	20	20
20	20	20	0	0	0
20	20	20	0	0	0
20	20	20	0	0	0

Gradient Strengthness $\Delta I \cong \left|\frac{dI}{dx}\right| + \left|\frac{dI}{dy}\right|$

Gradient Direction $\theta = \tan^{-1}\left(\frac{dI}{dy} \Big/ \frac{dI}{dx}\right)$

gradient strength

0	60	60	0
60	40	40	60
60	40	40	60
0	60	60	0

Padding

1	1	1
0	0	0
–1	–1	–1

convolve

0	0	0	20	20	20
0	0	0	20	20	20
0	0	0	20	20	20
20	20	20	0	0	0
20	20	20	0	0	0
20	20	20	0	0	0

output

0	60	60	0
60	40	40	60
60	40	40	60
0	60	60	0

valid

padding : 1

1	1	1
0	0	0
–1	–1	–1

convolve

0	0	0	0	0	0	0	0
0	0	0	0	20	20	20	0
0	0	0	0	20	20	20	0
0	0	0	0	20	20	20	0
0	20	20	20	0	0	0	0
0	20	20	20	0	0	0	0
0	20	20	20	0	0	0	0
0	0	0	0	0	0	0	0

output

0	0	20	40	60	40
0	0	0	0	0	0
40	60	20	–20	–60	–40
40	60	20	–20	–60	–40
0	0	0	0	0	0
–40	–60	–40	–20	0	0

same

Convolution

$$o = \frac{n + 2p - f}{s} + 1$$

Ex1:

$f = 3 \times 3$

$n = 6, f = 3, \ p = 0, s = 1$

$$out = \frac{6 + 2 \cdot 0 - 3}{1} + 1 = 4$$

$out = 4 \times 4$

Ex2:

$f = 3 \times 3$

$n = 6 \times 6, f = 3, \ p = 1, s = 1$

$$out = \frac{6 + 2 \cdot 1 - 3}{1} + 1 = 6$$

$out = 6 \times 6$

p : padding

h : height

w : width

f : filter, $f \times f$ or $h \times w$

I : image $n \times n$

s : stride

out : output

Ex3:

$f = 3 \times 4 = h \times w$

$I = 6 \times 6$

$n = 6, h = 3, \ p = 1, s = 1$

$$h_{out} = \frac{6 + 2 \cdot 1 - 3}{1} + 1 = 6$$

$$w_{out} = \frac{6 + 2 \cdot 1 - 4}{1} + 1 = 5$$

$I_{new=} 6 \times 5$

Ex4:

$f = 3 \times 5 = h \times w$

$I = 12 \times 8$

$n = 6, h = 3, \ p = 1, s = 1$

$$h_{out} = \frac{12 + 2 \cdot 1 - 3}{1} + 1 = 12$$

$$w_{out} = \frac{8 + 2 \cdot 1 - 5}{1} + 1 = 6$$

$I_{new=} 12 \times 6$

padding 0
stride 1

1	1	1
0	0	0
–1	–1	–1

convolve

0	0	0	20	20	20
0	0	0	20	20	20
0	0	0	20	20	20
20	20	20	0	0	0
20	20	20	0	0	0
20	20	20	0	0	0

output

0	0	0	40
60	20	–20	–20
60	20	–20	–20
40	40	40	0

valid

padding 1
stride 1

1	1	1
0	0	0
–1	–1	–1

convolve

0	0	0	0	0	0	0	0
0	0	0	0	20	20	20	0
0	0	0	0	20	20	20	0
0	0	0	0	20	20	20	0
0	20	20	20	0	0	0	0
0	20	20	20	0	0	0	0
0	20	20	20	0	0	0	0
0	0	0	0	0	0	0	0

output

0	0	20	40	60	40
0	0	0	0	40	40
40	60	20	–20	–20	0
40	60	20	–20	–20	0
0	20	20	0	0	0
–40	–20	0	20	0	0

same

padding 1
stride 2

1	1	1
0	0	0
–1	–1	–1

convolve

0	0	0	0	0	0	0	0
0	0	0	0	20	20	20	0
0	0	0	0	20	20	20	0
0	0	0	0	20	20	20	0
0	20	20	20	0	0	0	0
0	20	20	20	0	0	0	0
0	20	20	20	0	0	0	0
0	0	0	0	0	0	0	0

output

?

valid

padding 1, stride 2

wrong padding

1	1	1
0	0	0
−1	−1	−1

convolve

0	0	0	0	0	0	0	0
0	0	0	0	20	20	20	0
0	0	0	0	20	20	20	0
0	0	0	0	20	20	20	0
0	20	20	20	0	0	0	0
0	20	20	20	0	0	0	0
0	20	20	20	0	0	0	0
0	0	0	0	0	0	0	0

output

0	60	60	0
60	40	40	60
60	40	40	60
0	60	60	0

padding 3, stride 3

true padding

1	1	1
0	0	0
−1	−1	−1

			0	0	0	20	20	20			
			0	0	0	20	20	20			
			0	0	0	20	20	20			
			20	20	20	0	0	0			
			20	20	20	0	0	0			
			20	20	20	0	0	0			

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$$o = \frac{n + 2p - f}{s} + 1$$

$$6 = \frac{6 + 2 \cdot p - 3}{2} + 1$$

$$10 = 6 + 2 \cdot p - 3$$

$$7 = 2 \cdot p$$

$$p = 3.5$$

wrong padding

$$6 = \frac{6 + 2 \cdot p - 3}{3} + 1$$

$$15 = 6 + 2 \cdot p - 3$$

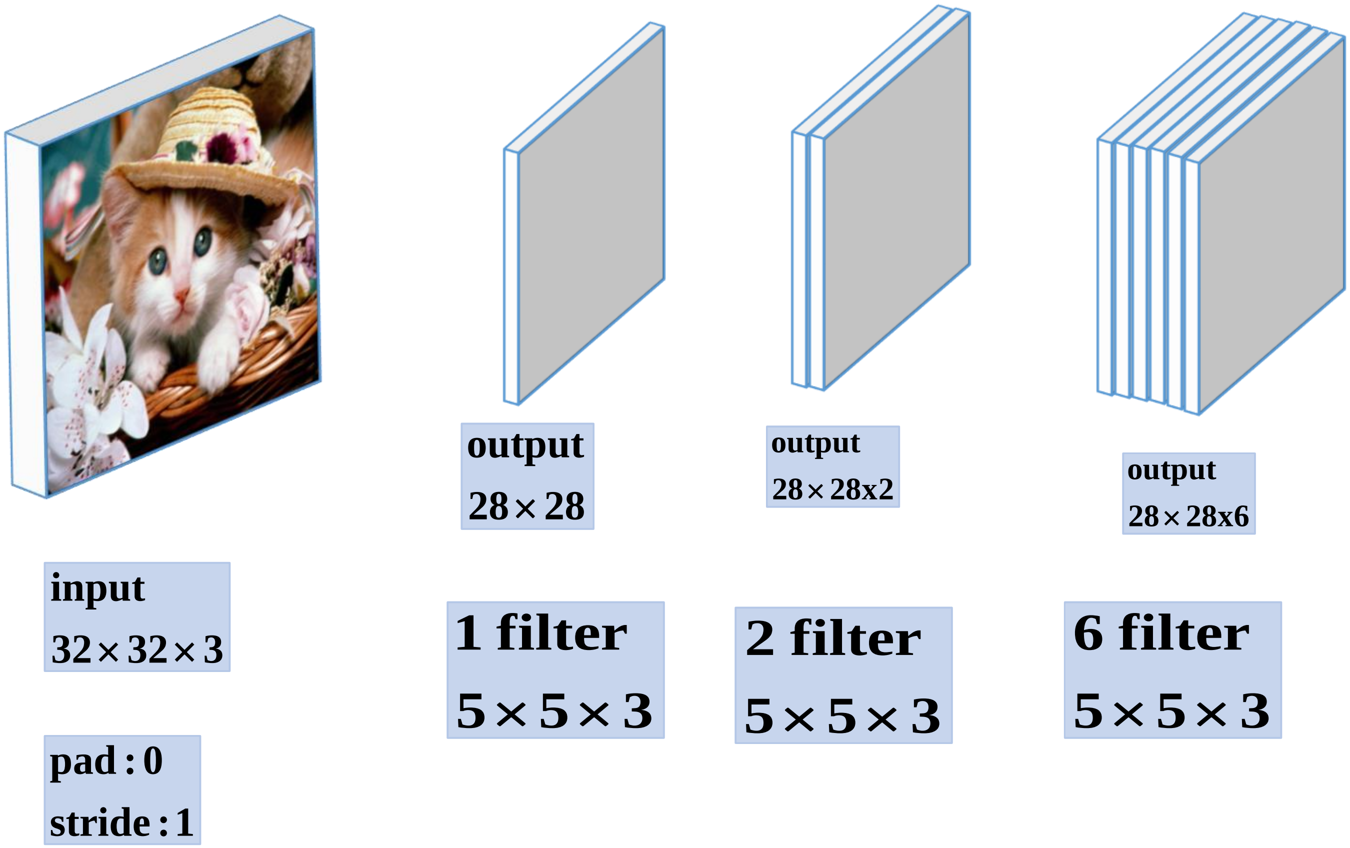
$$12 = 2 \cdot p$$

$$p = 3$$

true padding

$$p = \frac{f - 1}{2}$$

Convolution over Volume

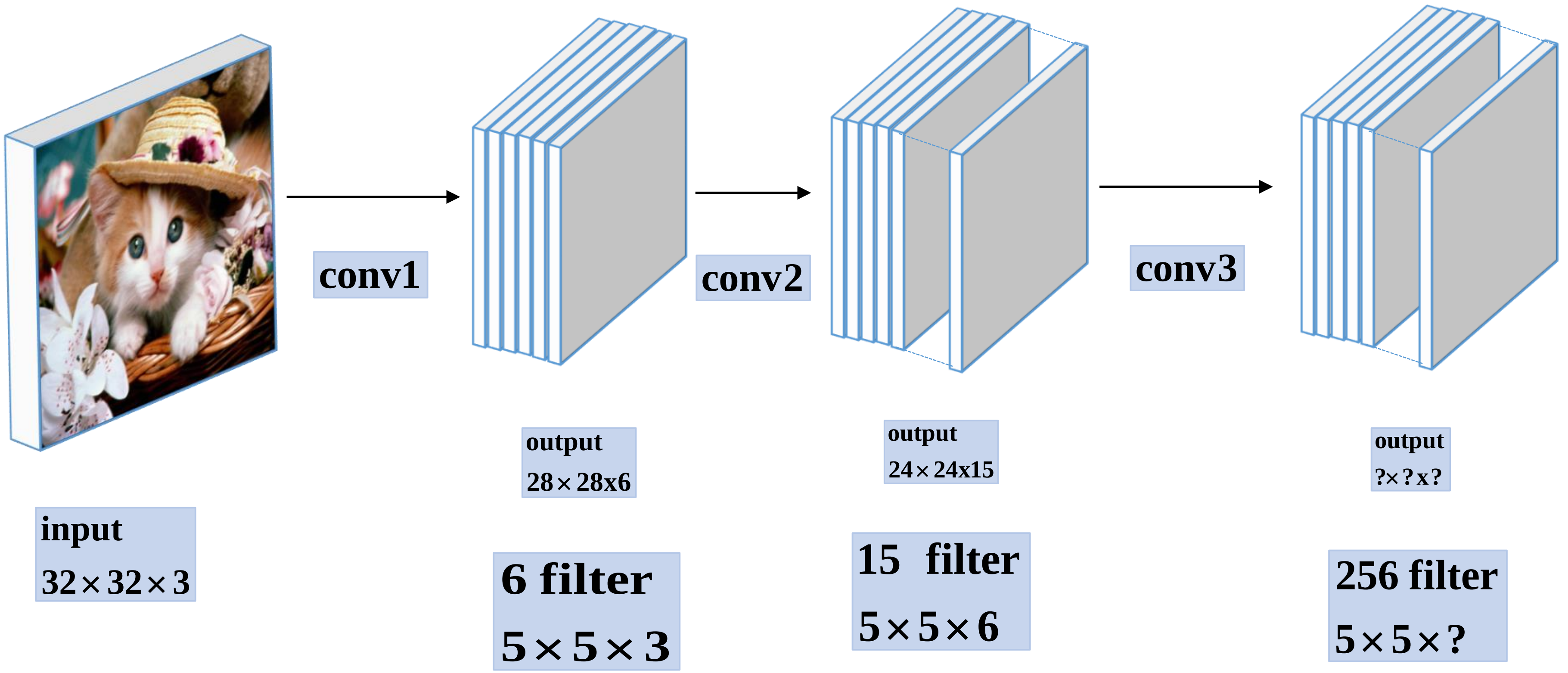


Convolution over Volume

input

$32 \times 32 \times 3$

Convolution over Volume



Max Pooling

10	25	12	3	67	23	0	77
8	1	4	5	12	1	44	3
11	3	12	12	3	5	11	5
2	46	12	12	2	23	21	34
1	5	8	99	12	11	1	3
0	5	22	2	45	1	23	3
1	89	22	23	13	9	5	7
12	1	2	34	1	5	5	3

$$8 \times 8$$

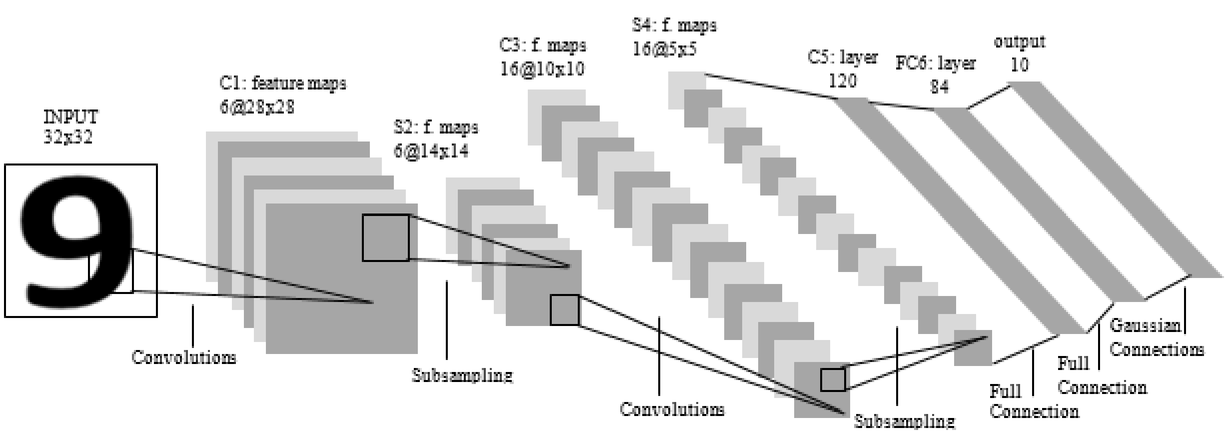
25	12	67	77
46	12	23	34
5	99	45	23
89	34	9	7

$$4 \times 4$$

max – pool : reduce the dimsion by [2,2]

Convolution LeNet5

input		32x32	
conv1	5x5 @6	28x28x6	
pool	2x2	14x14x6	
conv2	5x5x6 @16	10x10x16	
pool	2x2	5x5x16	
FC	5x5x16 @120	1x120	
FC	1x120 @84	1x84	
out	1x84 @10	1x10	predicted
		1x10	ground Truth



ReLU Activation

25	−12	67	77
46	12	−23	34
5	99	45	23
−89	34	9	7

$$\mathbf{relu = max(0, x)}$$

25	0	67	77
46	12	0	34
5	99	45	23
0	34	9	7

Softmax Loss

1	1	3	4

$$p \cong \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}}$$

0.0339	0.0339	0.2507	0.6815

$$e^{x_i} = \begin{bmatrix} 2.7183 & 2.7183 & 20.0855 & 54.5982 \end{bmatrix}$$

$$\sum_{i=1}^{n} e^{x_i} = (2.7183 + 2.7183 + 20.0855 + 54.5982) = 80.1203$$

$$p \cong \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}} = \begin{bmatrix} 0.0339 & 0.0339 & 0.2507 & 0.6815 \end{bmatrix}$$

Convolution LeNet5

input		32x32	
conv1	5x5 @6	28x28x6	
reLU		28x28x6	
pool	2x2	14x14x6	
conv2	5x5x6 @16	10x10x16	
reLU		10x10x16	
pool	2x2	5x5x16	
FC	5x5x16 @120	1x120	
FC	1x120 @84	1x84	
out	1x84 @10	1x10	predicted
		1x10	softmax
		1x10	ground Truth