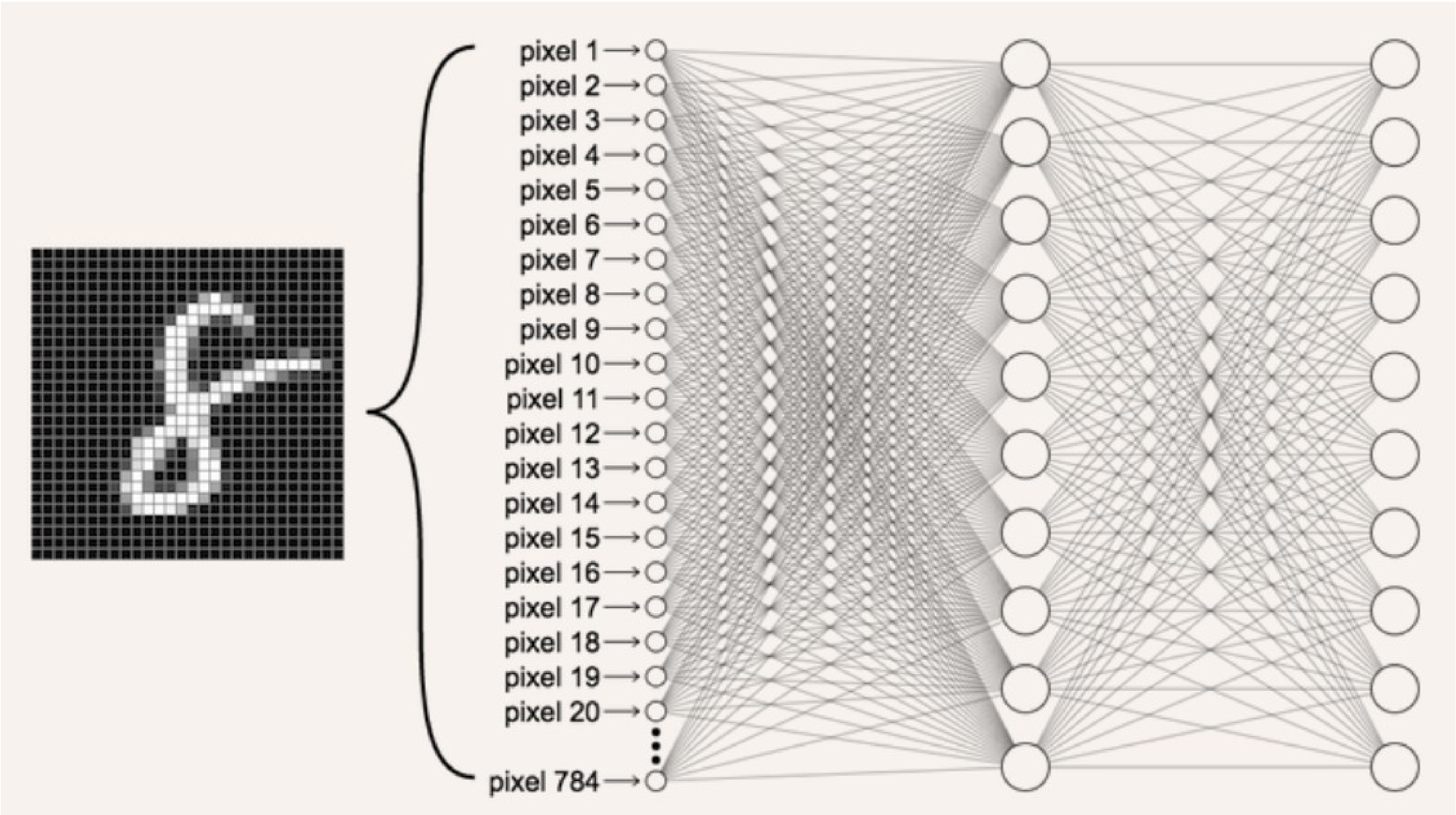
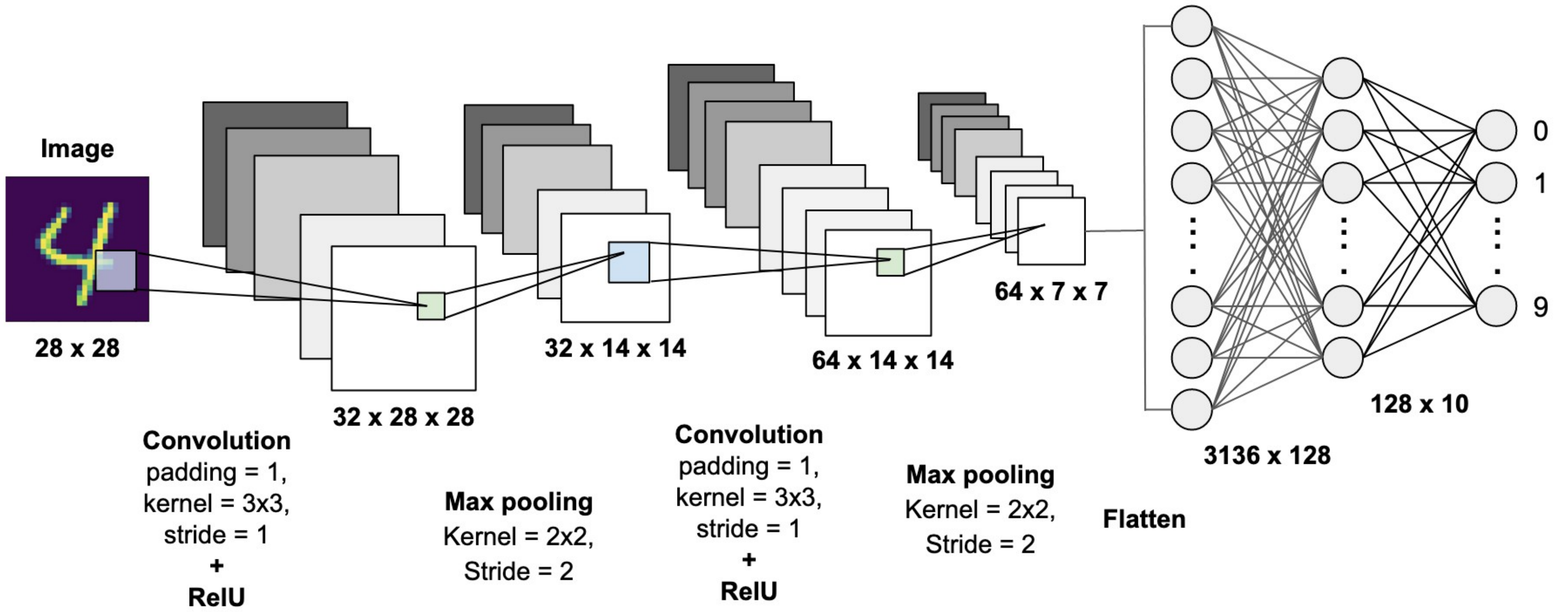
CNN

By, Joe Johnson, D17024

Classification - MLP



Classification - CNN



Filters

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline\end{array}$$

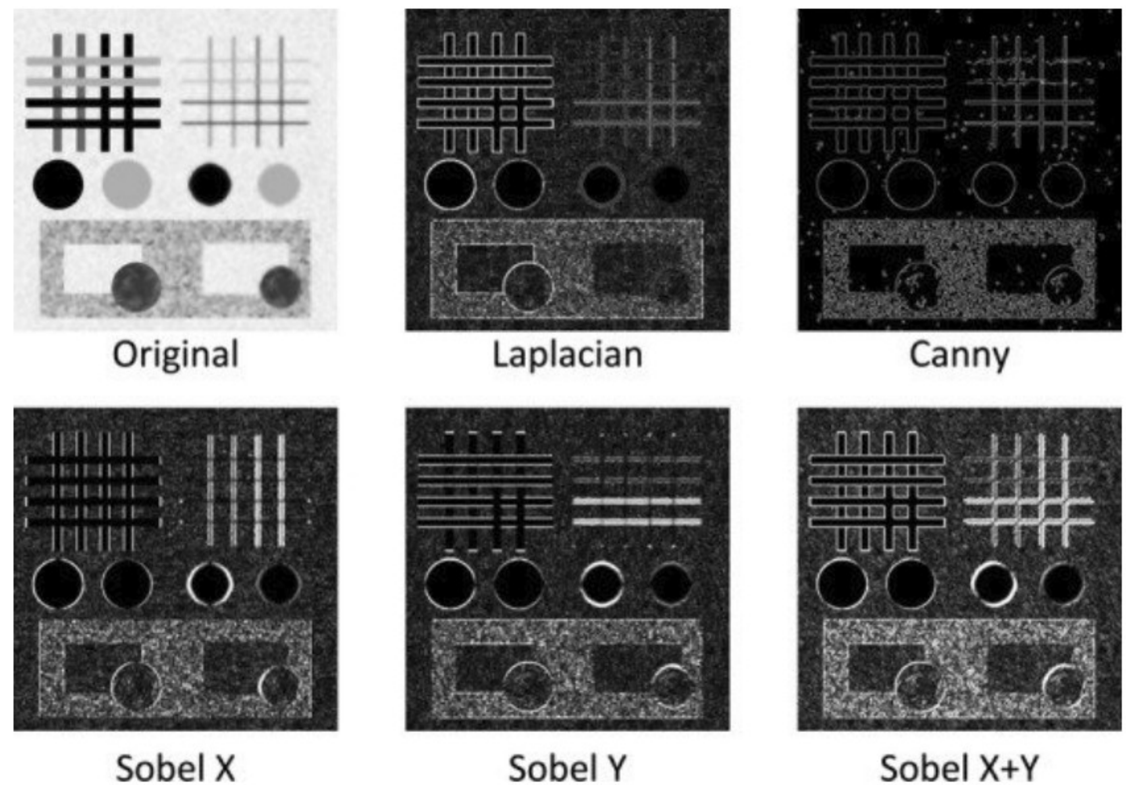
$$\frac{1}{159}\begin{array}{|c|c|c|c|c|}\hline 2 & 4 & 5 & 4 & 2 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 5 & 12 & 15 & 12 & 5 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 2 & 4 & 5 & 4 & 2 \\ \hline\end{array}$$

X – Direction Kernel

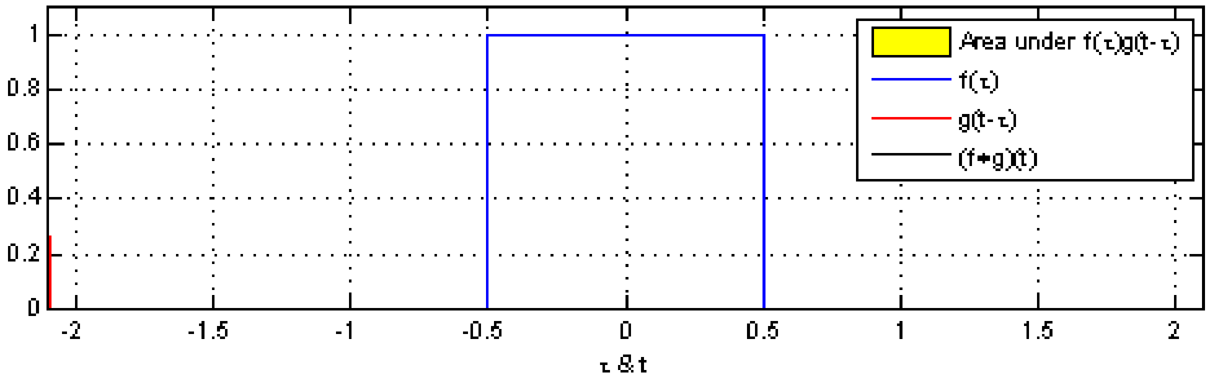
-1	0	1
-2	0	2
-1	0	1

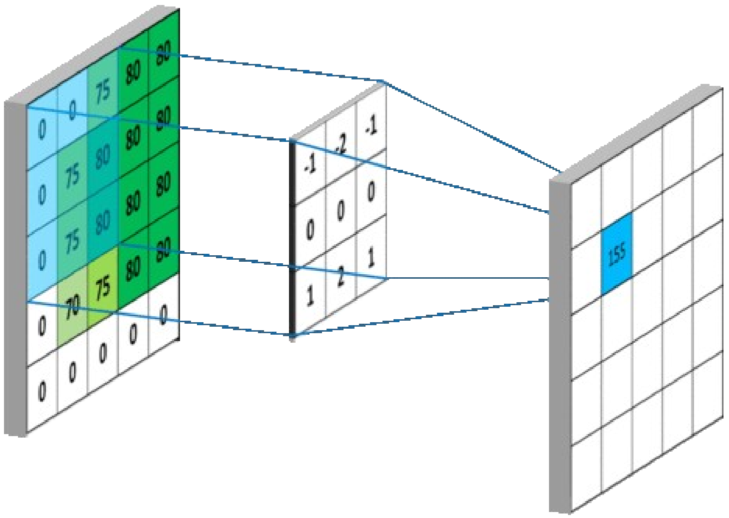
Y – Direction Kernel

-1	-2	-1
0	0	0
1	2	1

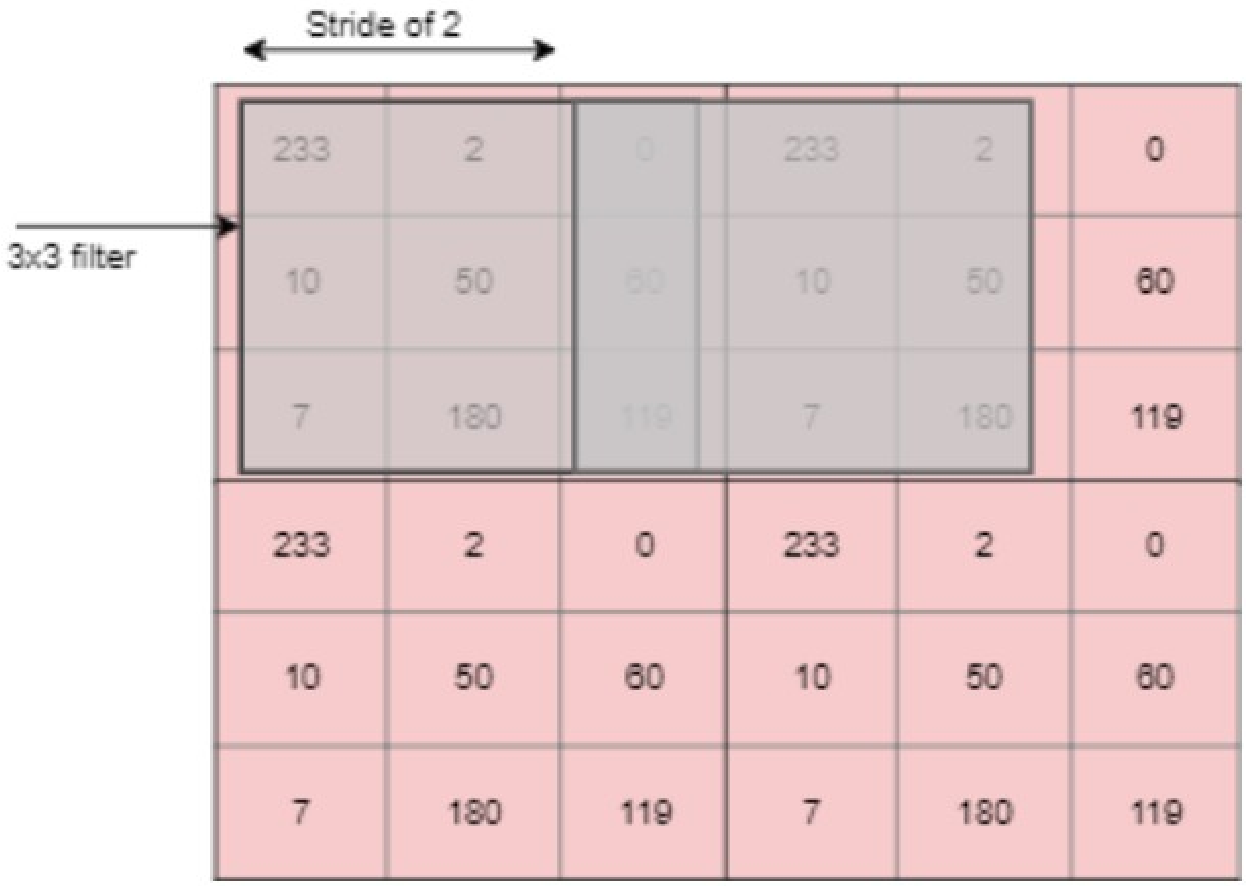


Convolution operation





Stride



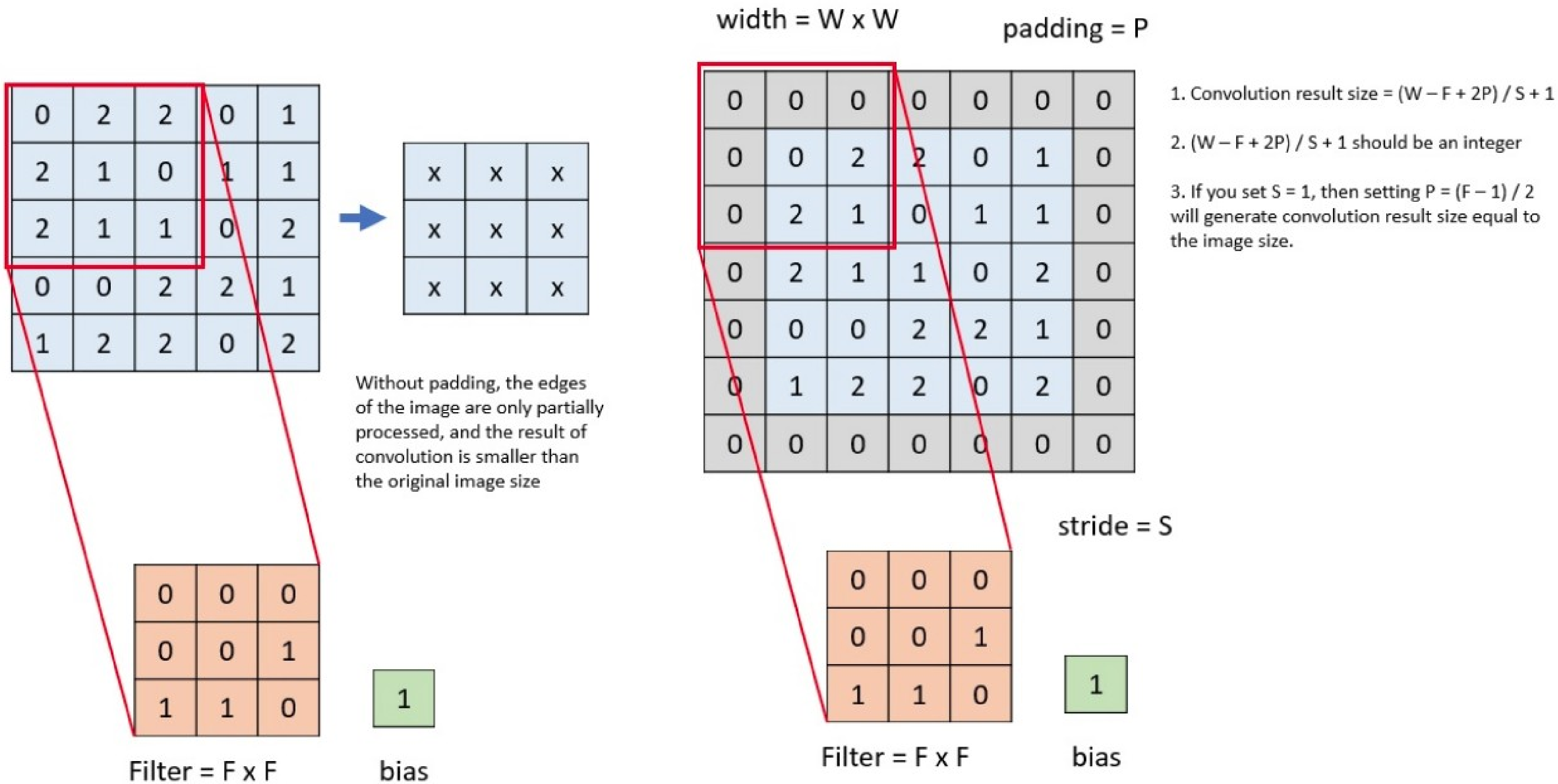
Stride of 2
3x3 filter
233
2
0
233
2
0
10
50
60
10
50
60
7
180
119
7
180
119
233
2
0
233
2
0
10
50
60
10
50
60
7
180
119
7
180
119

Padding

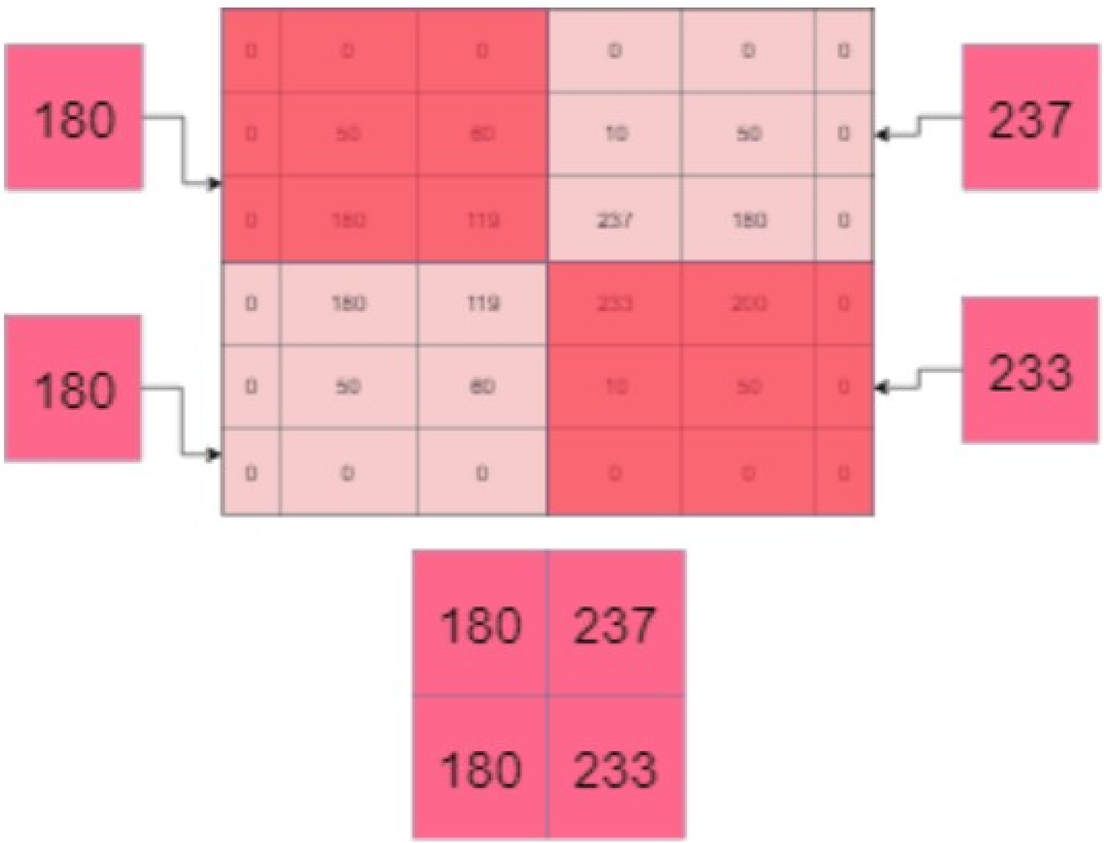
0	0	0	0	0	0
0	50	60	10	50	0
0	180	119	237	180	0
0	180	119	233	200	0
0	50	60	10	50	0
0	0	0	0	0	0

Convolution Operation

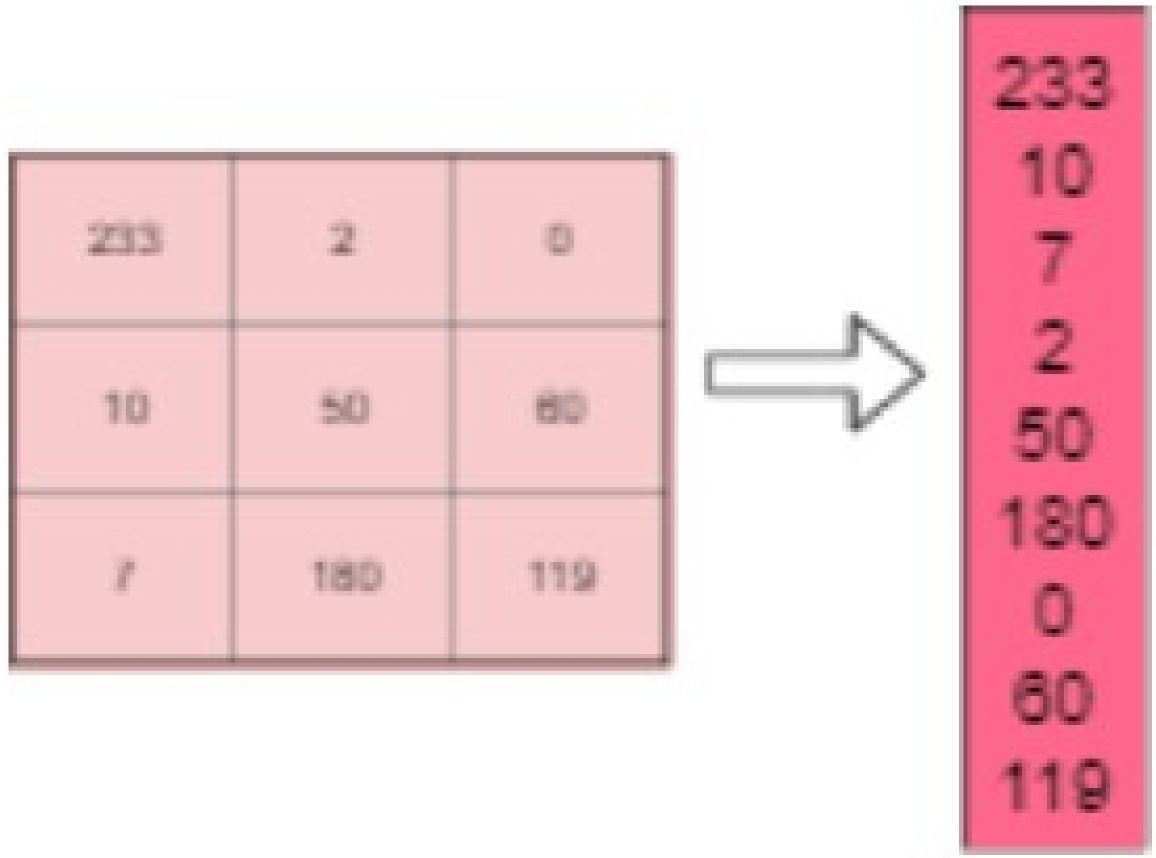
Convolution Operation



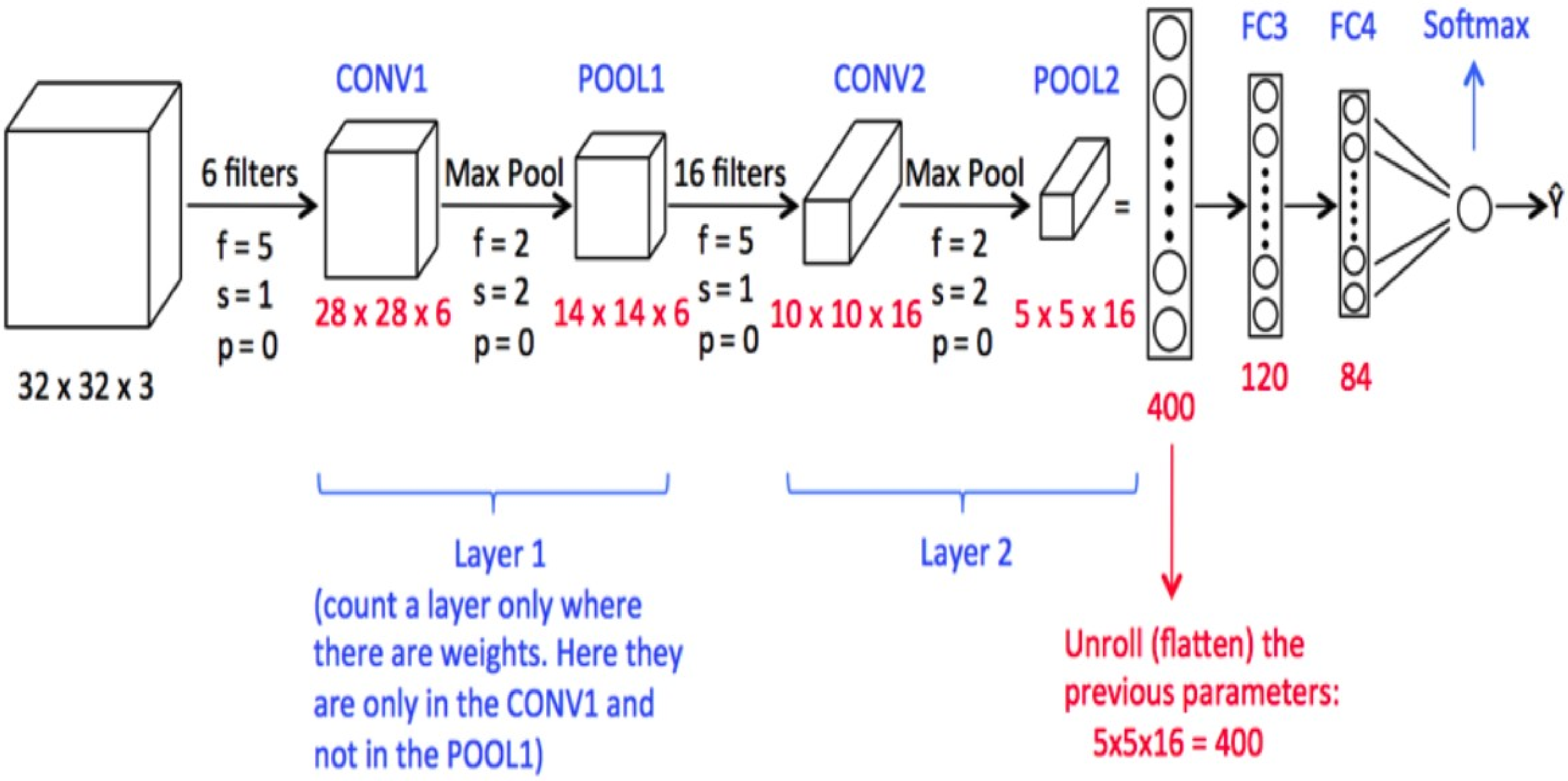
Pooling



Fatten

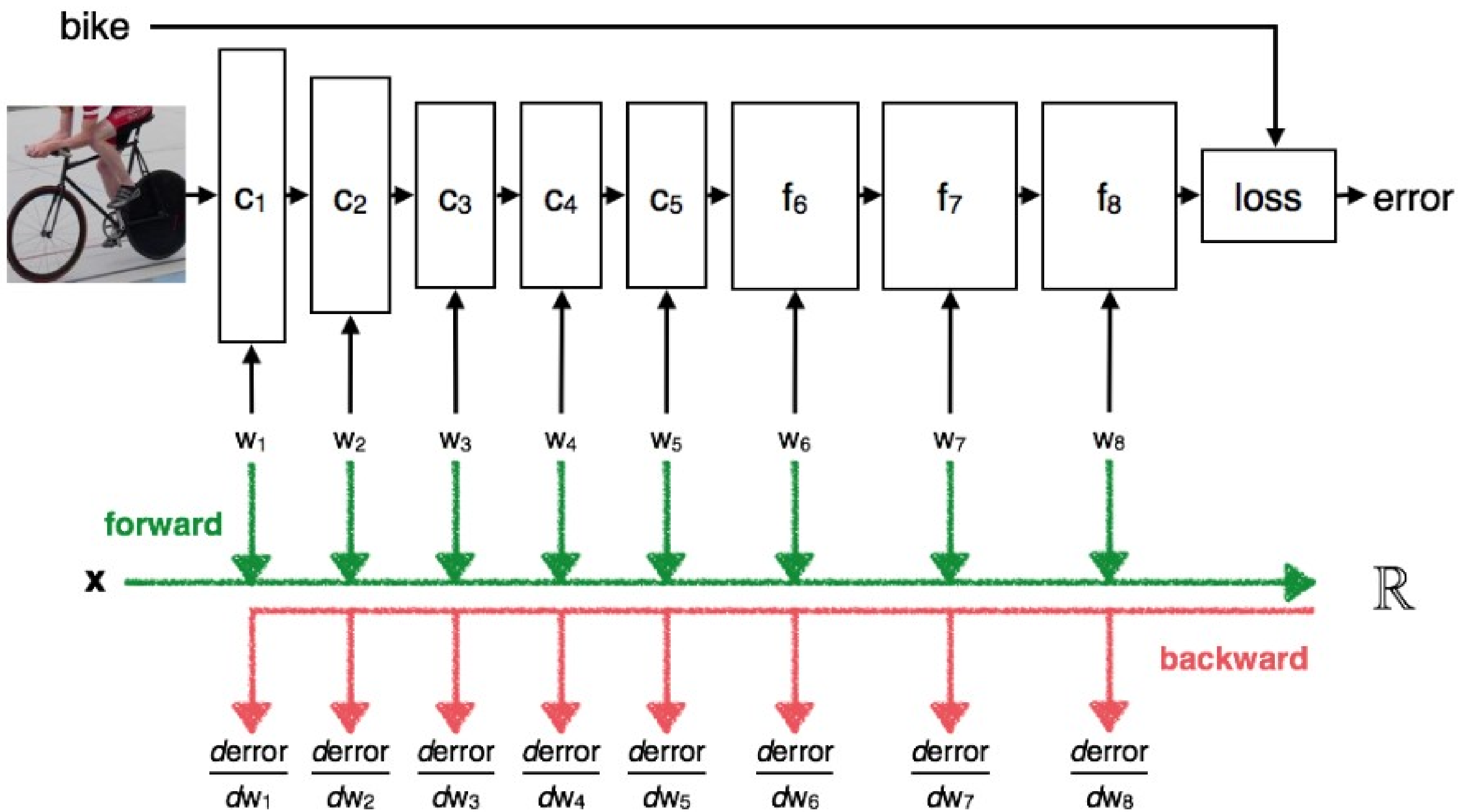


CNN

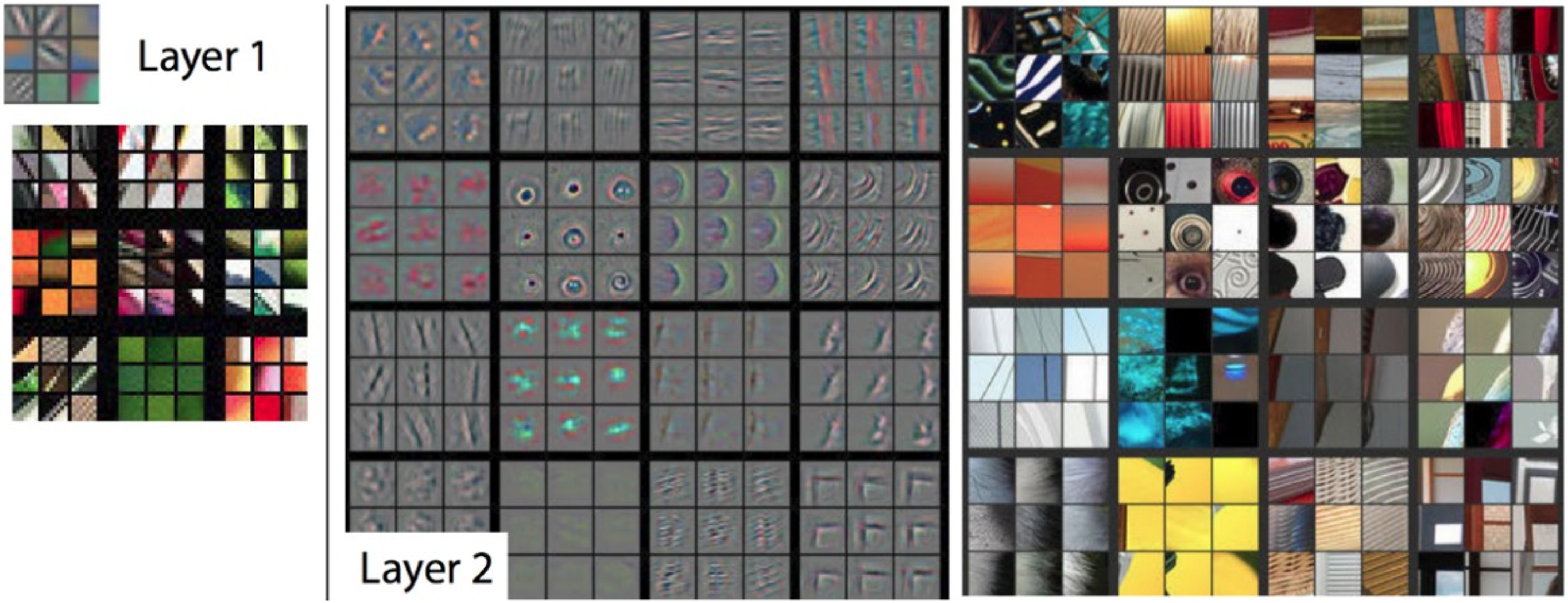


CONV1
POOL1
CONV2
POOL2
FC3
FC4
Softmax
6 filters
f = 5
s = 1
p = 0
Max Pool
f = 2
s = 2
p = 0
16 filters
f = 5
s = 1
p = 0
Max Pool
f = 2
s = 2
p = 0
=
Ŷ
32 x 32 x 3
28 x 28 x 6
14 x 14 x 6
10 x 10 x 16
5 x 5 x 16
400
120
84
Layer 1
(count a layer only where there are weights. Here they are only in the CONV1 and not in the POOL1)
Layer 2
Unroll (flatten) the previous parameters:
5x5x16 = 400

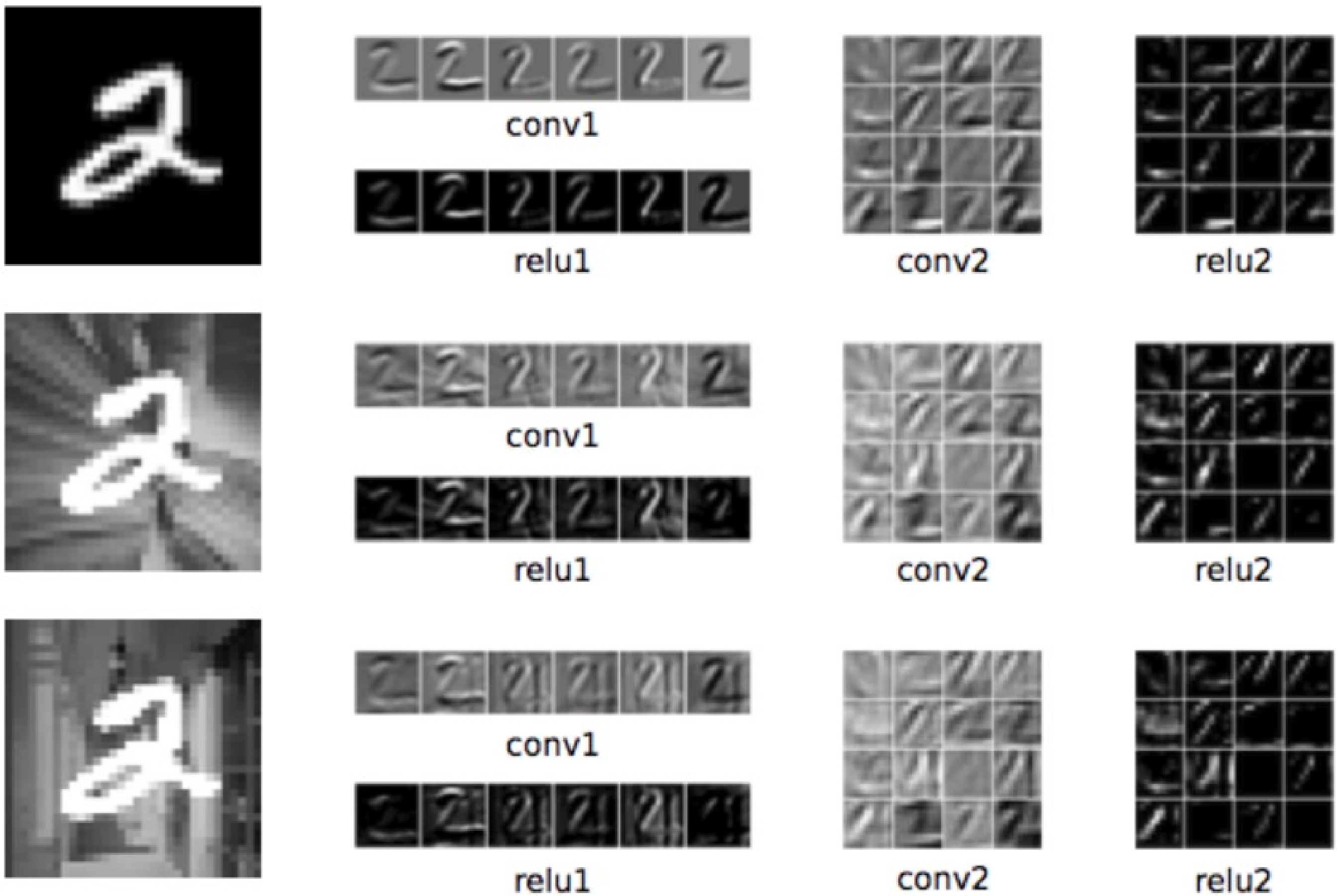
CNN - BP

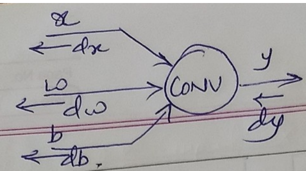

bike
c1
c2
c3
c4
c5
f6
f7
f8
loss
error
w1
w2
w3
w4
w5
w6
w7
w8
forward
x
ℝ
backward
derror/dw1
derror/dw2
derror/dw3
derror/dw4
derror/dw5
derror/dw6
derror/dw7
derror/dw8

Learned filter



Feature/Activation Map

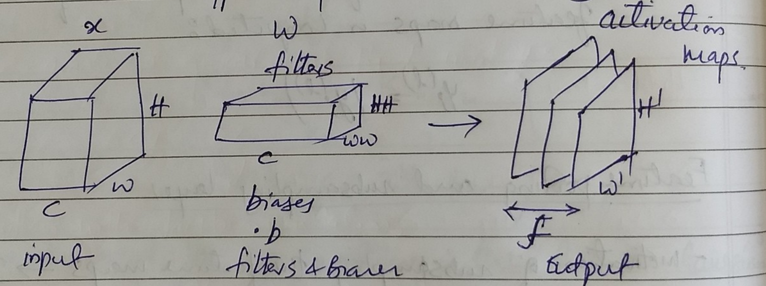




$*$ → convolution of 2 tensors (input x and a filter w)

if x and w share the same shape
$x * w$ → scaler equal $\sum_{i=1}^{N} w_i \cdot x_i$

- if $w < x$ → Obtain an activation map y where each value is the predefined convolution operation of a sub-region of x with the sizes of w.
  - This sub-region activated by the filter is sliding all across the i/p array X.



Input:

X: Input data of shape (N, C, H, W)
N - data points
C - channels
H → height
W - width.

W: Filter weights of shape (F, C, HH, WW),
F → diff filters
C → channel
HH → ht, WW → width

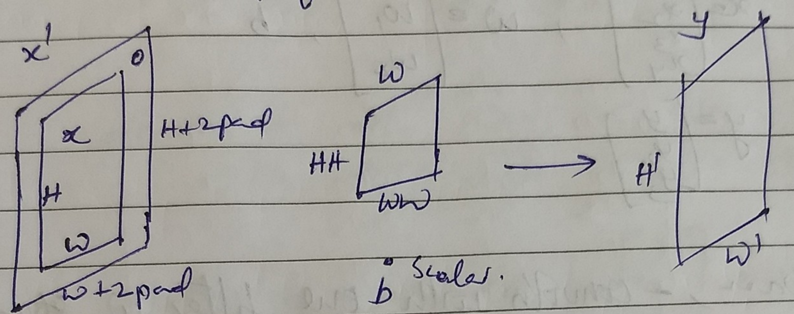
b: Biases, of shape $(F,)$

conv_parameter → 'stride', 'pad'_zeropad
Returns a tuple of:
- out: Output data, of shape (N, F, H', W')
  $H' = 1 + (H + 2 * pad - HH) / stride$
  $W' = 1 + (W + 2 * pad - WW) / stride$

Date: /

Forward pass

Generic case (Simplified with $N=1, C=1, F=1$)



$\forall (i,j) \in [1, H'] \times [1, W']$

$$y_{ij} = \left(\sum_{k=1}^{HH} \sum_{l=1}^{WW} w_{kl} \, x'_{si+k-1,\, sj+l-1} \right) + b$$

Specific case: stride $=1$, pad $=0$, no bias

$$y_{ij} = \sum_k \sum_l w_{kl} \cdot x_{i+k-1,\, j+l-1}$$

Backpropagation

$$dy = \left(\frac{\partial L}{\partial y_{ij}} \right)$$

also $\frac{\partial L}{\partial x}$, $\frac{\partial L}{\partial w}$ + $\frac{\partial L}{\partial b}$

Trivial case: input X is a vector (1D)

I/p: $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$, $w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$, b

O/p: $y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$

Forward pass, - convol'n with one filter w, stride = 1, padding =

$$y_1 = w_1 x_1 + w_2 x_2 + b$$
$$y_2 = w_1 x_2 + w_2 x_3 + b$$
$$y_3 = w_1 x_3 + w_2 x_4 + b$$

BP, gradient of our cost fn 'L' w.r.t y:

$$dy = \frac{\partial L}{\partial y}$$

- written with Jacobian notation,

$$dy = \left[\frac{\partial L}{\partial y_1}, \frac{\partial L}{\partial y_2} \quad \frac{\partial L}{\partial y_3}\right]$$

$$dy = [dy_1 \quad dy_2 \quad dy_3]$$

dy and y share the same shape:

$$dy = (dy1, dy2, dy3)$$

we are looking for,

$$dx = \frac{\partial L}{\partial x}, \quad dw = \frac{\partial L}{\partial w}, \quad db = \frac{\partial L}{\partial b}$$

db

$$db = \frac{\partial L}{\partial b} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial b} = dy \cdot \frac{\partial y}{\partial b}$$

$$db = \sum_{j} \frac{\partial L}{\partial y_j} \cdot \frac{\partial y_j}{\partial b} = \sum_{j=1}^{3} \frac{\partial L}{\partial y_j} \cdot \frac{\partial y_j}{\partial b}$$

$$db = \begin{bmatrix} dy_1 & dy_2 & dy_3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \underline{\underline{dy_1 + dy_2 + dy_3}}$$

dw

$$dW = \frac{\partial L}{\partial W} = dy \cdot \frac{\partial y}{\partial W} = \begin{bmatrix} dy_1 & dy_2 & dy_3 \end{bmatrix} \frac{\partial y}{\partial W}$$

$$\frac{\partial y}{\partial W} = \begin{bmatrix} \frac{\partial y_1}{\partial w_1} & \frac{\partial y_1}{\partial w_2} \\ \frac{\partial y_2}{\partial w_1} & \frac{\partial y_2}{\partial w_2} \\ \frac{\partial y_3}{\partial w_1} & \frac{\partial y_3}{\partial w_2} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \\ x_3 & x_4 \end{bmatrix}$$

$$dW = \begin{bmatrix} dw_1 \\ dw_2 \end{bmatrix} = \begin{bmatrix} dy_1 & dy_2 & dy_3 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \\ x_3 & x_4 \end{bmatrix}$$

$$dw_1 = x_1 dy_1 + x_2 dy_2 + x_3 dy_3$$
$$dw_2 = x_2 dy_1 + x_3 dy_2 + x_4 dy_3$$

$$dw = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} * \begin{bmatrix} dy_1 \\ dy_2 \\ dy_3 \end{bmatrix}$$

dx

$$dx = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x} = dy^T \cdot \frac{\partial y}{\partial x}$$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} & \frac{\partial y_1}{\partial x_4} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} & \frac{\partial y_2}{\partial x_4} \\ \frac{\partial y_3}{\partial x_1} & \frac{\partial y_3}{\partial x_2} & \frac{\partial y_3}{\partial x_3} & \frac{\partial y_2}{\partial x_4} \end{bmatrix}$$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} w_1 & w_2 & 0 & 0 \\ 0 & w_1 & w_2 & 0 \\ 0 & 0 & w_1 & w_2 \end{bmatrix}$$

$dx_1 = w_1 dy_1$

$dx_2 = w_2 dy_1 + w_1 dy_2$

$dx_3 = w_2 dy_2 + w_1 dy_3$

$dx_4 = w_2 dy_3$

$$dx = \begin{bmatrix} 0 \\ dy_1 \\ dy_2 \\ dy_3 \\ 0 \end{bmatrix} * \begin{bmatrix} w_2 \\ w_1 \end{bmatrix}$$

Input : X is a matrix (2D)

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix}, \quad W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}, \; b$$

Output

stride = 1 and no padding

$$y = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix}$$

Forward pass

$$y_{11} = w_{11}x_{11} + w_{12}x_{12} + w_{21}x_{21} + w_{22}x_{22} + b$$

$$y_{12} = w_{11}x_{12} + w_{12}x_{13} + w_{21}x_{22} + w_{22}x_{23} + b$$

$$\vdots$$

Written with subscripts,

$$y_{ij} = \left(\sum_{k=1}^{2} \sum_{l=1}^{2} w_{kl} \, x_{i+k-1,\, j+l-1} \right) + b \quad \forall (i,j) \in \{1,2,3\}^2$$

Backpropagation :

$$dy_{ij} = \frac{\partial L}{\partial y_{ij}}$$

db

$$db = dy_{ij} \cdot \frac{\partial y_{ij}}{\partial b}$$

$$db = \sum_{i=1}^{3} \sum_{j=1}^{3} dy_{ij} \, , \quad \forall (i,j) \; \frac{\partial y_{ij}}{\partial b} = 1$$

$\underline{dw}$

$$dw = \frac{\partial L}{\partial y_{ij}} \cdot \frac{\partial y_{ij}}{\partial w} = dy_{ij} \cdot \frac{\partial y}{\partial w}$$

$$dw_{mn} = dy_{ij} \cdot \frac{\partial y_{ij}}{\partial w_{mn}}$$

$$\frac{\partial y_{ij}}{\partial w_{mn}} = \sum_{k=1}^{2} \sum_{l=1}^{2} \frac{\partial w_{kl}}{\partial w_{mn}} \cdot x_{i+k-1,\, j+l-1}$$

All term.

$$\frac{\partial w_{kl}}{\partial w_{mn}} = 0$$

Except for $(k,l) = (m,n)$ where it is 1,

$$\Rightarrow \frac{\partial y_{ij}}{\partial w_{mn}} = x_{i+k-1,\, j+l-1}$$

$$\Rightarrow dw_{mn} = dy_{ij} \cdot x_{i+k-1,\, j+l-1} = \sum_{i=1}^{3} \sum_{j=1}^{3} dy_{ij} \cdot x_{i+k-1,\, j+l-1}$$

$$dw = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} * \begin{bmatrix} dy_{11} & dy_{12} & dy_{13} \\ dy_{21} & dy_{22} & dy_{23} \\ dy_{31} & dy_{32} & dy_{33} \end{bmatrix}$$

$$dw = x * dy$$

dx

$$dx_{mn} = dy_{ij} \cdot \frac{\partial y_{ij}}{\partial x_{mn}}$$

$$\frac{\partial y_{ij}}{\partial x_{mn}} = \sum_{k=1}^{2}\sum_{l=1}^{2} w_{kl} \cdot \frac{\partial x_{i+k-1,j+l-1}}{\partial x_{mn}}$$

$$\frac{\partial x_{i+k-1,j+l-1}}{\partial x_{mn}} = \begin{cases} 1 & \text{iff } m = i+k-1 \Rightarrow n = j+l-1 \\ 0 & \text{iff non} \end{cases}$$

$$\begin{cases} m = i+k-1 \\ n = j+l-1 \end{cases} \Rightarrow \begin{cases} k = m-i+1 \\ l = n-j+1 \end{cases}$$

in given Eg: range sets for indices are:

$m, n \in [1,4]$ i/ps

$k, l \in [1,2]$ filters

$i, j \in [1,3]$ o/ps

~~whereas~~ when we set $k = m-i+1 \Rightarrow$ going to be out of the defined boundaries: $(m-i+1) \in [-1,4]$

- extend the defn of matrix W with 0 values as soon as ~~the~~ indices will go out of the defined range.
- In double sum,

$$\frac{\partial y_{ij}}{\partial x_{mn}} = w_{m-i+1,\, n-j+1}$$

Where W is ~~not~~ 0-extended initial filter,

$$dx_{mn} = \sum_{i=1}^{3}\sum_{j=1}^{3} dy_{ij} \cdot w_{m-i+1,\, n-j+1}$$

Page No.
Date : / /2

$$w'_{ij} = w_{3-i,3-j}$$

$$dx = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & dy_{11} & dy_{12} & dy_{13} & 0 \\ 0 & dy_{21} & dy_{22} & dy_{23} & 0 \\ 0 & dy_{31} & dy_{32} & dy_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} w_{22} & w_{21} \\ w_{12} & w_{11} \end{bmatrix}$$

$$dx = dy_0 * w'$$

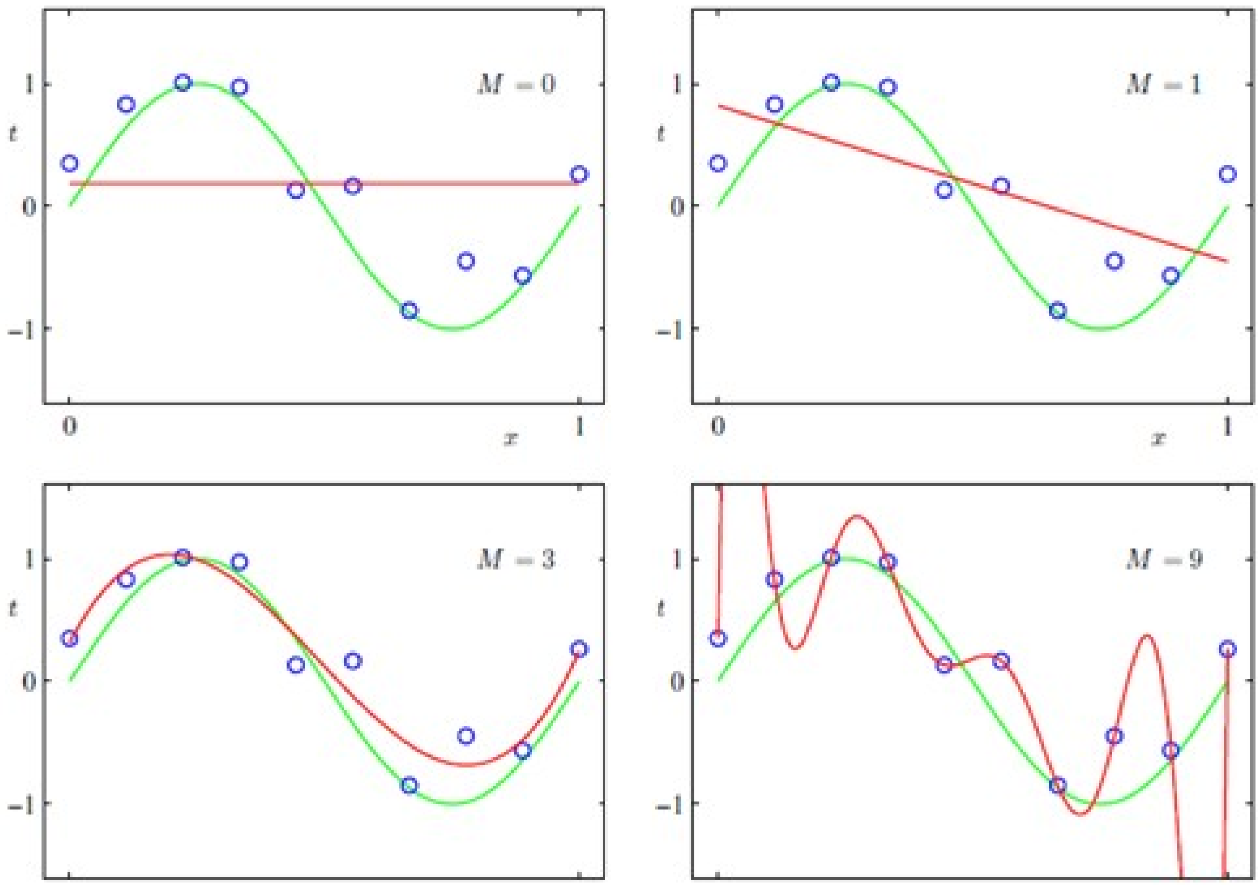
<u>Summary of backpropn eqns</u>

$$db = \sum_{i=1}^{3} \sum_{j=1}^{3} dy_{ij}$$

$$dw = x * dy$$

$$dx = dy_0 * w'$$

Fitting



Pooling layer:

- o/p feature maps are sensitive to the location of the features in the i/p.

- One approach to reduce sensitive to the location of feature — down sample the feature map.
  - robust to changes in the position of the feature in the image — "local translation invariance".

1. average pooling — summarize the average presense of a feature
2. max pooling — most activated presence of feature

Another down sampling → change stride in convolution layer

input Image → convolutional layer → Non-linearity fn → pooling layer

pooling filter. → 2X2 pixels with a stride of 2 pixels

→ reduce feature map size to quater of original size

Global pooling layers

- down samples the entire feature map to a single value.
  - pool size set to size of the i/p feature map.

- alternative to FCN from feature maps to an o/p prediction
  - aggressively summarize of a feature in a image
- Global average pooling
- Global max pooling.

Regularization — make things regular or acceptable.

- techniques used to reduce the error by fitting a fn appropriately on the given training set and avoid of overfitting.

- technique used for tuning the fn by adding an additional penalty term in the error fn.
  - ↳ adds term controls the excessively fluctuating fn such that the coefficients don't take extreme values.
  - ↳ techniques of reducing the value of error coefficients are called shrinkage methods or wt decay in case of NN.

- Overfitting can be controlled by increasing the size of training dataset.

$$E = E_D(W) + \lambda E_W(W)$$

$$E = \frac{1}{2}\sum_{n=1}^{N}(t_n - W^T\phi(x_n))^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|W_j|^q$$

$$\sum_{j=1}^{M}|W_j|^q \leq \eta$$

$q = 1$ or 2.

Lasso reg → L1 (Lasso reg) Ridge reg → L2

freq of many