Computational Intelligence

Samaneh Hosseini

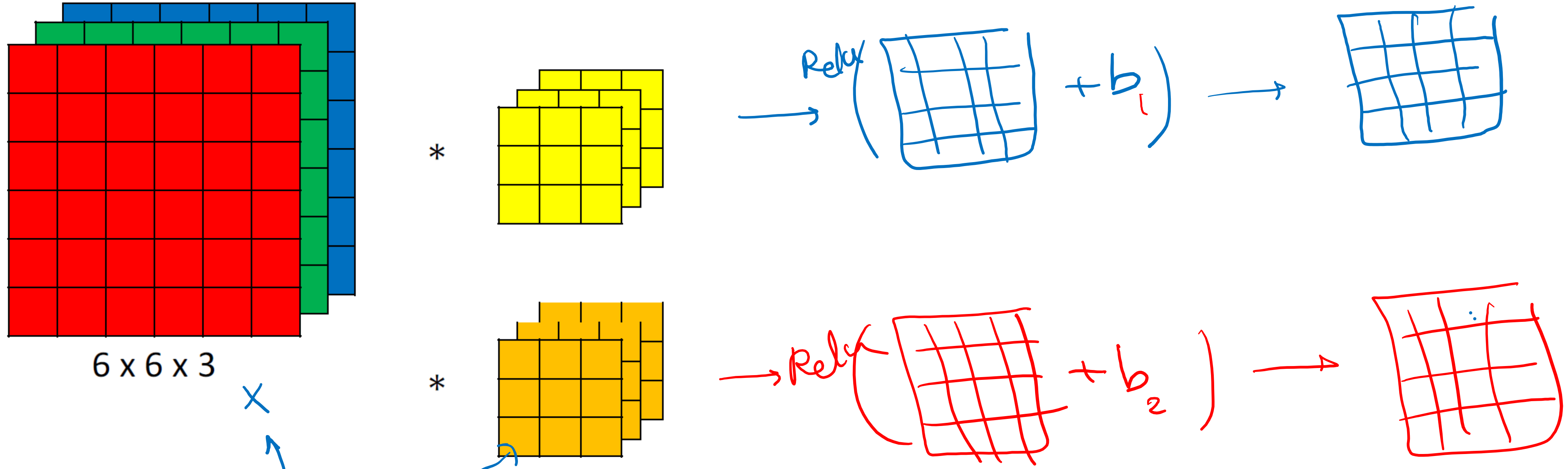
Isfahan University of Technology

Outline

- Convolutional Neural Networks
  - One Layer of a Convolutional Network
  - Simple Convolutional Network Example
  - Pooling Layers
  - CNN Example

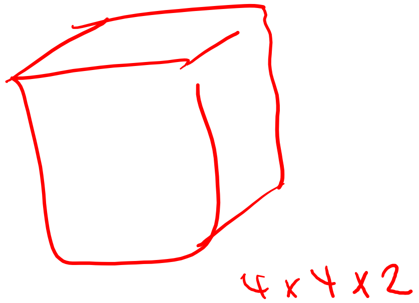
Convolutional Neural Networks:
One layer of a convolutional network

Example of a layer



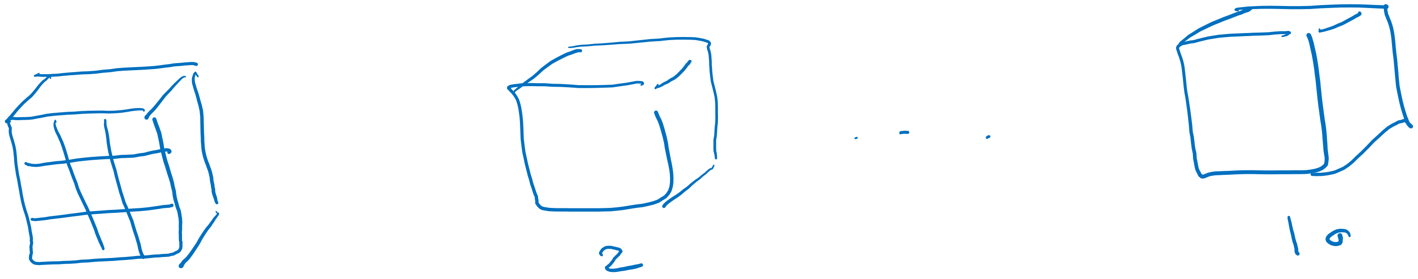
$$z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$$

$$a^{[1]} = g(z^{[1]})$$



Number of parameters in one layer

If you have 10 filters that are 3 x 3 x 3 in one layer of a neural network, how many parameters does that layer have?



3x3x3

27 parmeter

+bias

28 Parameters

280 paramter

Summary of notation

- If layer l is a convolution layer:
- $f^{[l]}$ = filter size $\quad f^{[l]} \times f^{[l]}$
- $p^{[l]}$ = padding
- $s^{[l]}$ = stride
- $n_c^{[l]}$= number of filters
- Each filter is: $\quad f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \leftarrow$
- Activations: $\quad a^{[l]} \rightarrow m \times n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$
- Weights: $\quad f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]} \rightarrow$ # filters in layer l
- bias: $\quad n_c^{[l]}$

$$- \left(1, 1, 1, n_c^{[l]}\right)$$

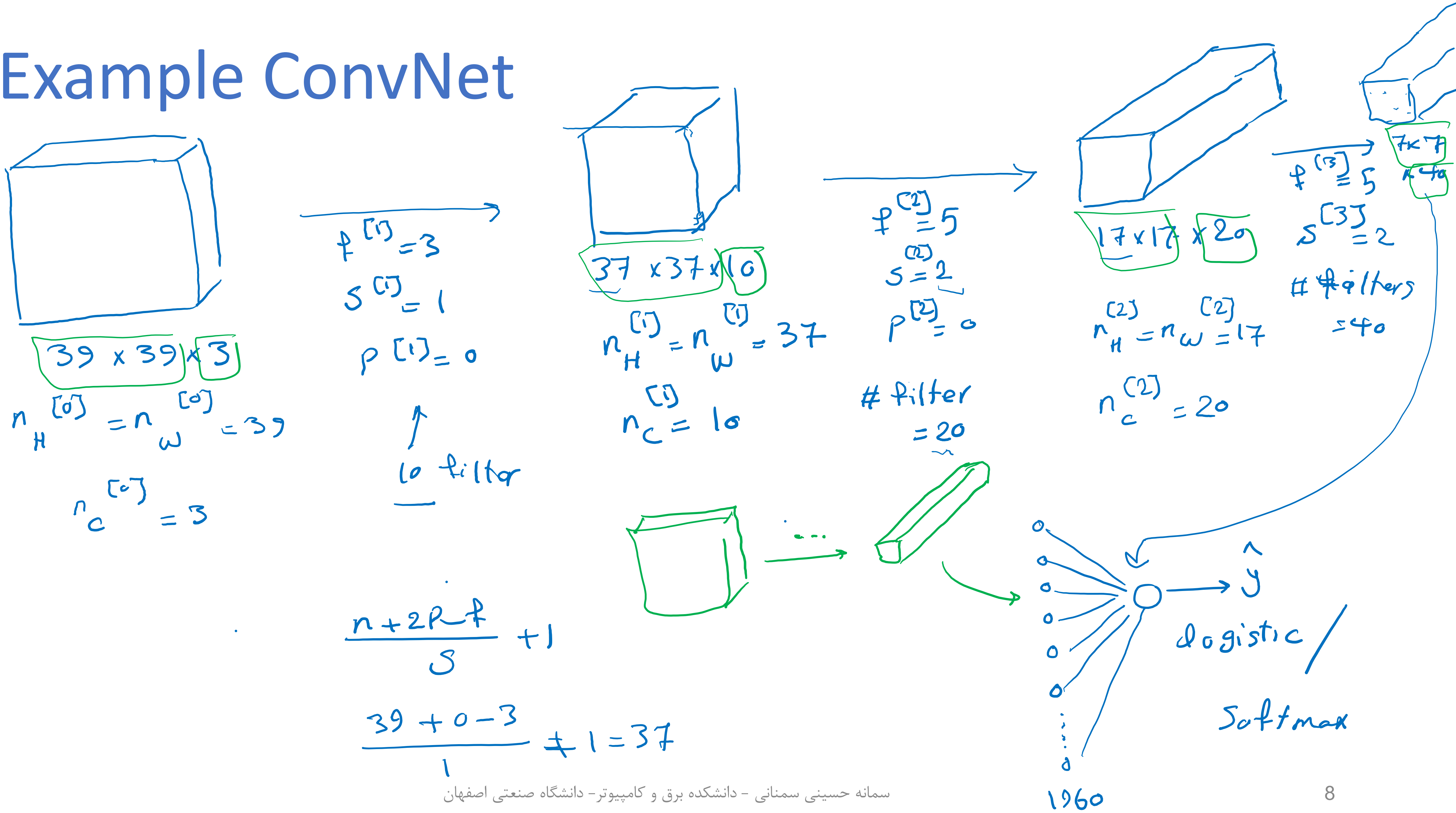
Input: $\quad n_H^{[l-1]} \times n_W^{[l-1]} \times \boxed{n_c^{[l-1]}}$

Output: $\quad n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$ $\qquad n_c \times n_H , n_W$

$$n_{\cancel{H}\,W}^{[l]} = \left\lfloor \frac{n_{\cancel{H}\,W}^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$$

Convolutional Neural Networks:
A simple convolution network example

Example ConvNet



$39 \times 39 \times 3$

$n_H^{[0]} = n_W^{[0]} = 39$

$n_c^{[0]} = 3$

$f^{[1]} = 3$

$s^{[1]} = 1$

$p^{[1]} = 0$

10 filter

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

$$\frac{39+0-3}{1} + 1 = 37$$

$37 \times 37 \times 10$

$n_H^{[1]} = n_W^{[1]} = 37$

$n_c^{[1]} = 10$

$f^{[2]} = 5$

$s^{[2]} = 2$

$p^{[2]} = 0$

filter = 20

$17 \times 17 \times 20$

$n_H^{[2]} = n_W^{[2]} = 17$

$n_c^{[2]} = 20$

$f^{[3]} = 5$

$s^{[3]} = 2$

filters = 40

$7 \times 7 \times 40$

1960

$\hat{y}$

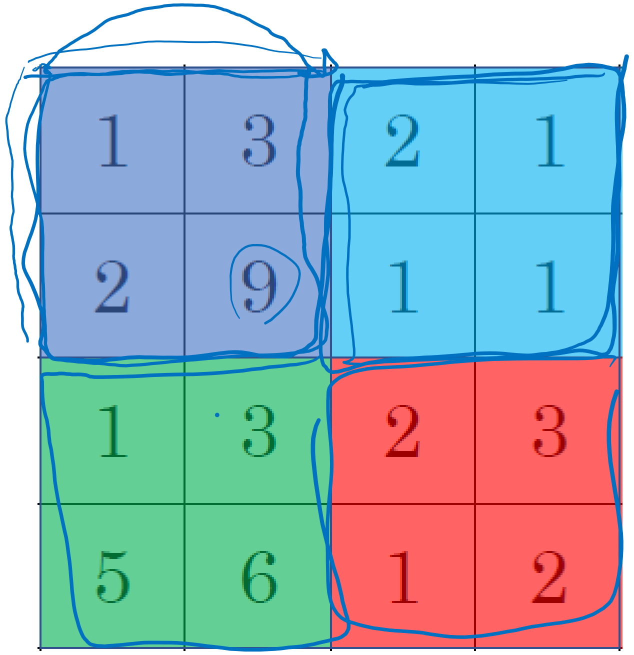
logistic / Softmax

Types of layer in a convolutional network

- Convolution (Conv)
- Pooling (PooL)
- Fully connected (FC)

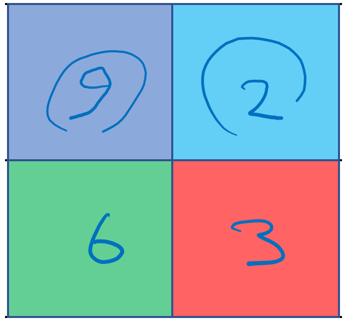
Convolutional Neural Networks:
Pooling layers

Pooling layer: Max pooling



1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

4x4



9	2
6	3

2x2

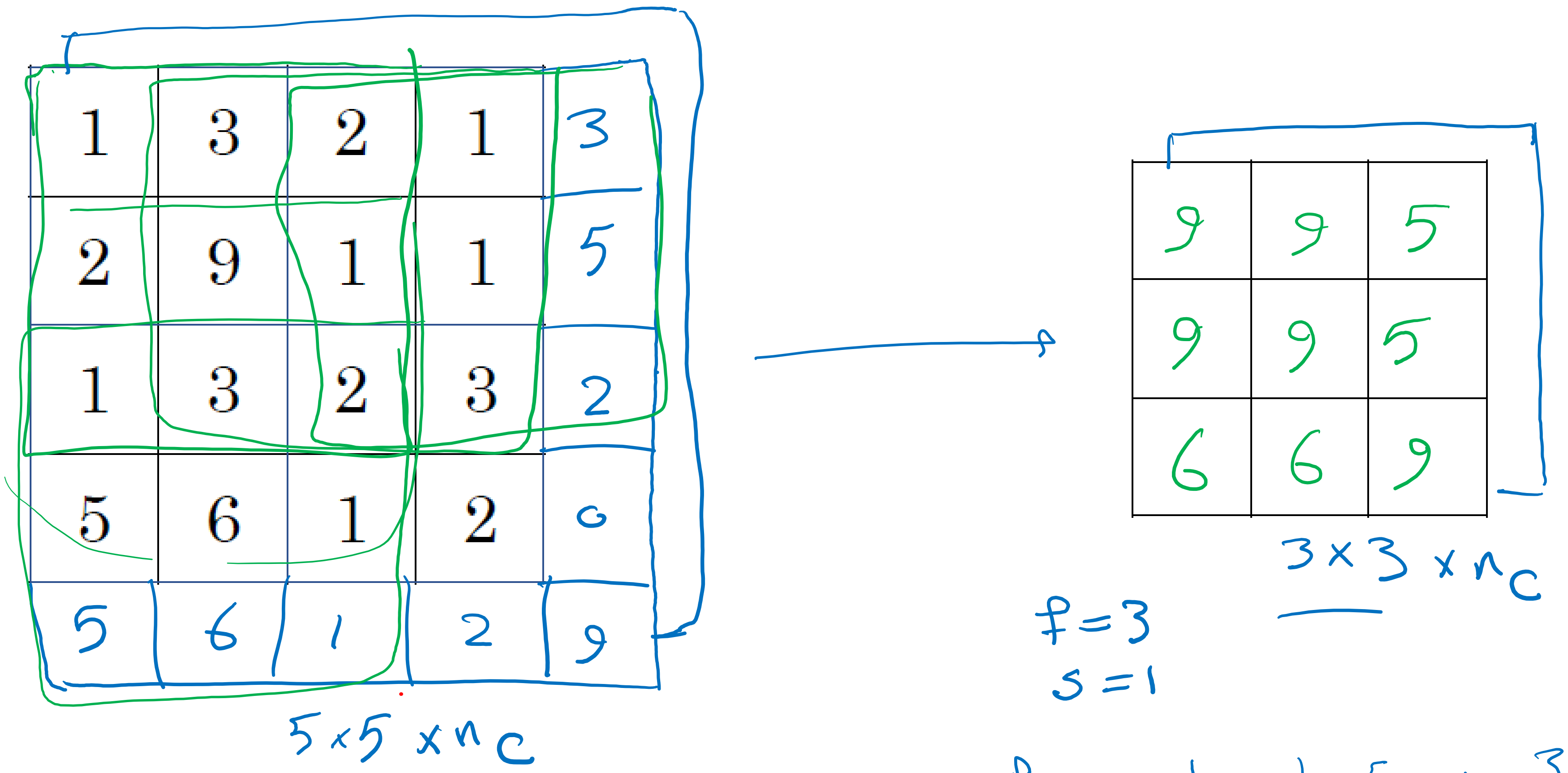
hyperparameter

f = 2

s = 2

no training parameter

Pooling layer: Max pooling



1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
5	6	1	2	0
5	6	1	2	9

$5 \times 5 \times n_c$

9	9	5
9	9	5
6	6	9

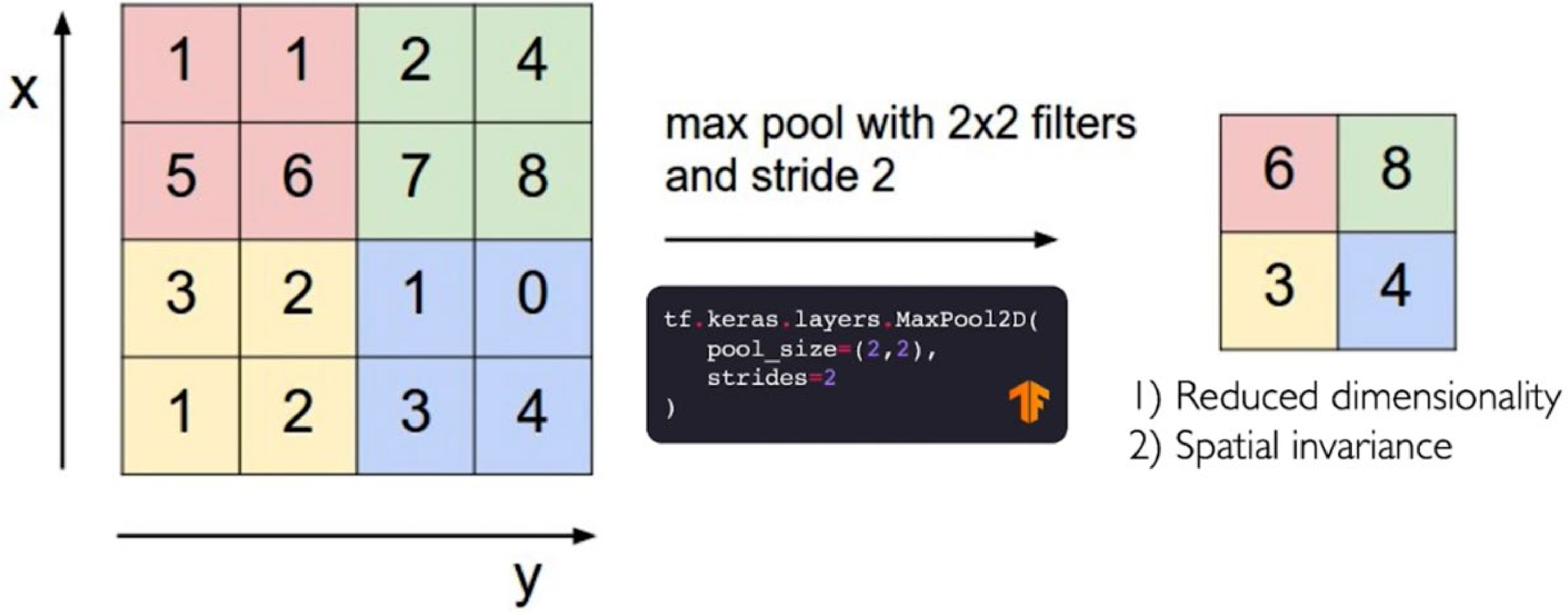
$3 \times 3 \times n_c$

$f=3$

$s=1$

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \quad \left\lfloor \frac{5+0-3}{1} + 1 \right\rfloor = 3$$

Pooling layer: Max pooling



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters and stride 2

```
tf.keras.layers.MaxPool2D(
    pool_size=(2,2),
    strides=2
)
```

6	8
3	4

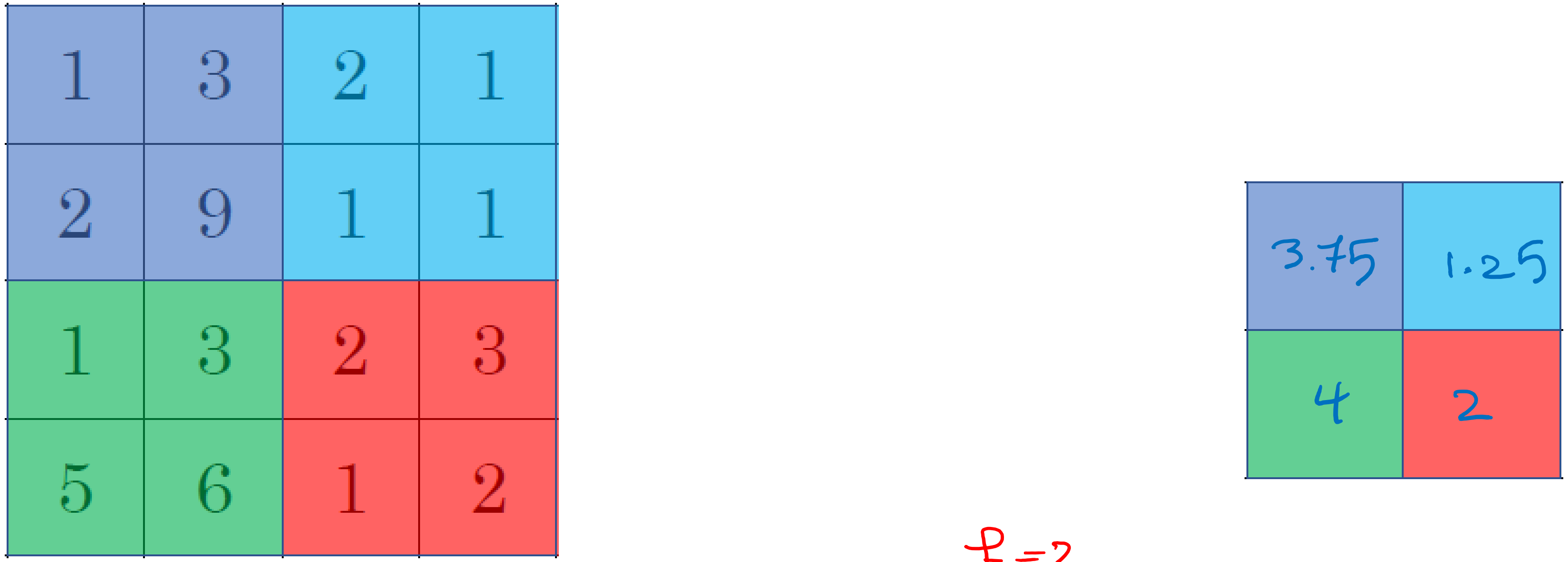
1) Reduced dimensionality
2) Spatial invariance

How else can we downsample and preserve spatial invariance?

Pooling layer: Average pooling

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

Pooling layer: Average pooling



1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

3.75	1.25
4	2

$f = 2$

$s = 2$

$7 \times 7 \times 1000 \longrightarrow 1 \times 1 \times 1000$

Summary of pooling

- Hyperparameters:
- f : filter size
- s : stride
- Max or average pooling

$f=2 \quad s=2$

$f=3 \quad s=2$

$n_H \times n_W \times n_C$

$$\left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_W - f}{s} + 1 \right\rfloor$$

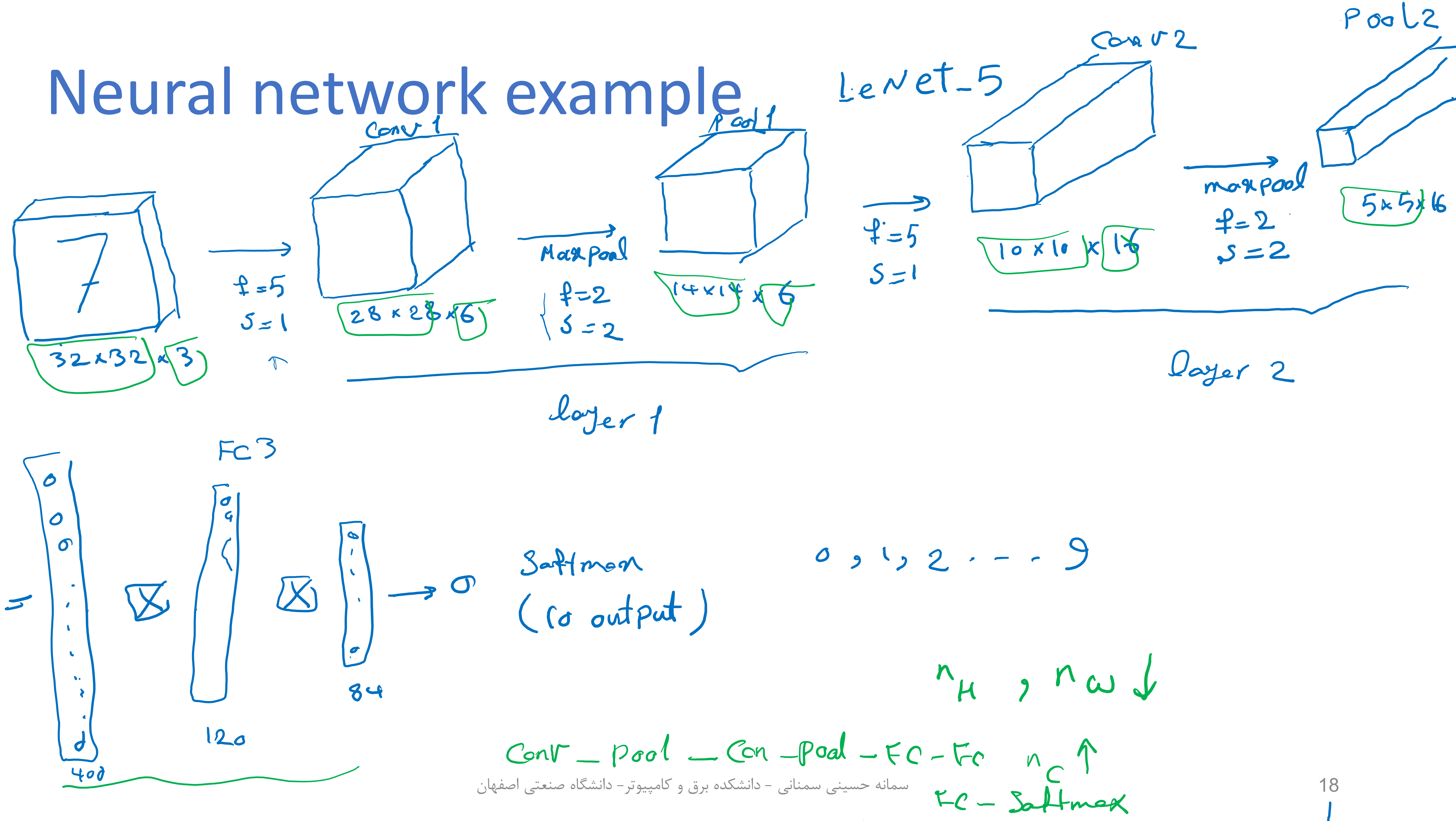
$\times \, n_C$

No parameter to learn.

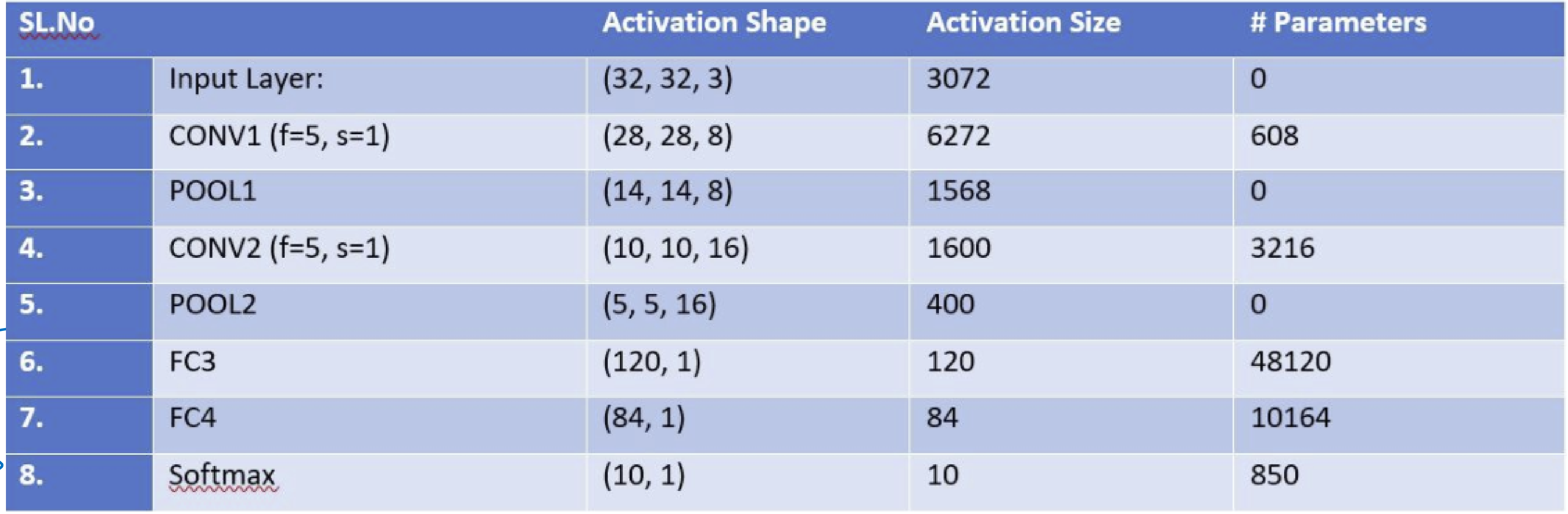
~~P : padding~~

Convolutional Neural Networks:
CNN Example

Neural network example



Neural network example



SL.No		Activation Shape	Activation Size	# Parameters
1.	Input Layer:	(32, 32, 3)	3072	0
2.	CONV1 (f=5, s=1)	(28, 28, 8)	6272	608
3.	POOL1	(14, 14, 8)	1568	0
4.	CONV2 (f=5, s=1)	(10, 10, 16)	1600	3216
5.	POOL2	(5, 5, 16)	400	0
6.	FC3	(120, 1)	120	48120
7.	FC4	(84, 1)	84	10164
8.	Softmax	(10, 1)	10	850

Outline

- Convolutional Neural Networks
    - One Layer of a Convolutional Network
    - Simple Convolutional Network Example
    - Pooling Layers
    - CNN Example