

# Applied Text Analytics & Natural Language Processing

with Dr. Mahdi Roozbahani  
& Wafa Louhichi

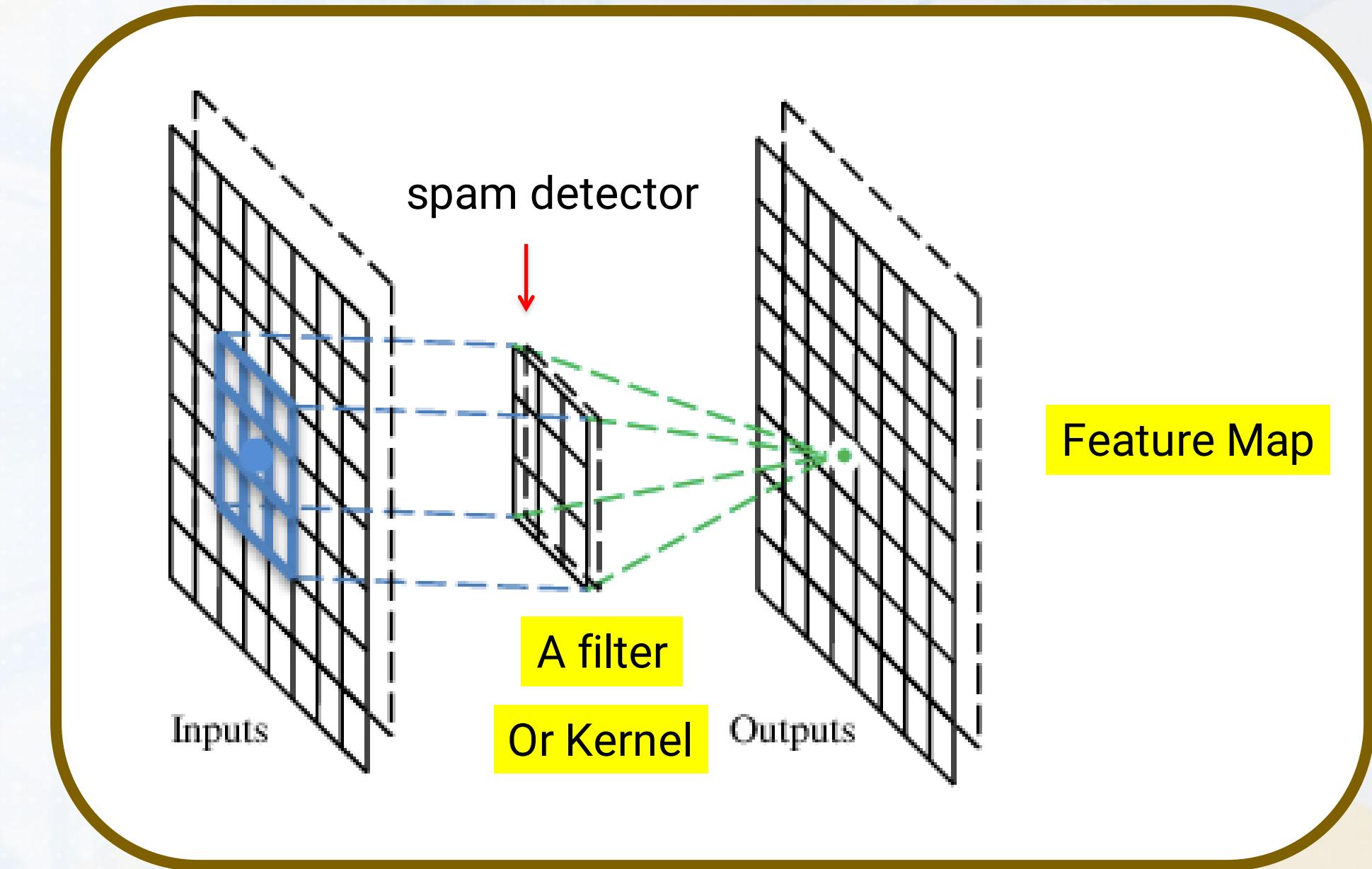
*Deep Learning*  
**Convolutional Neural Network (CNN) – Part 2**

Some of the slides are based on Ming Li (University of Waterloo – Deep Learning Part)  
with some modifications



# Important Concept: A Convolutional Layer

- Now that all documents have the same size and are converted into a matrix form; we can start feeding them into a CNN model
- A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has several filters that do the convolutional operation

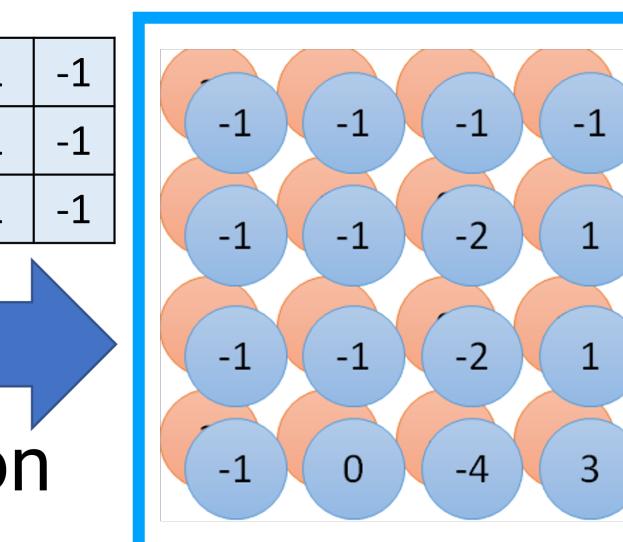


# Convolution v.s. Fully Connected

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

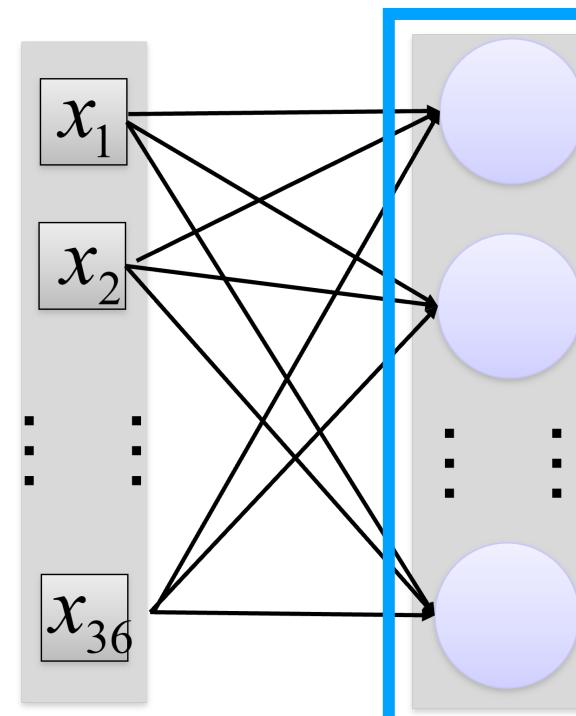
Data point

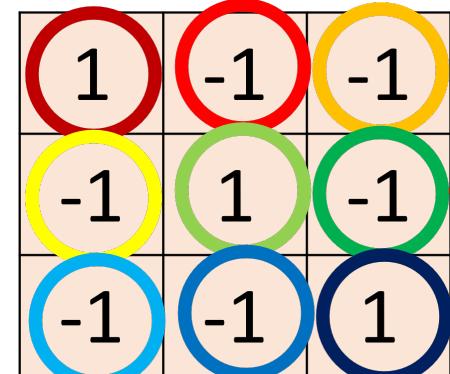
1	-1	-1
-1	1	-1
-1	-1	1



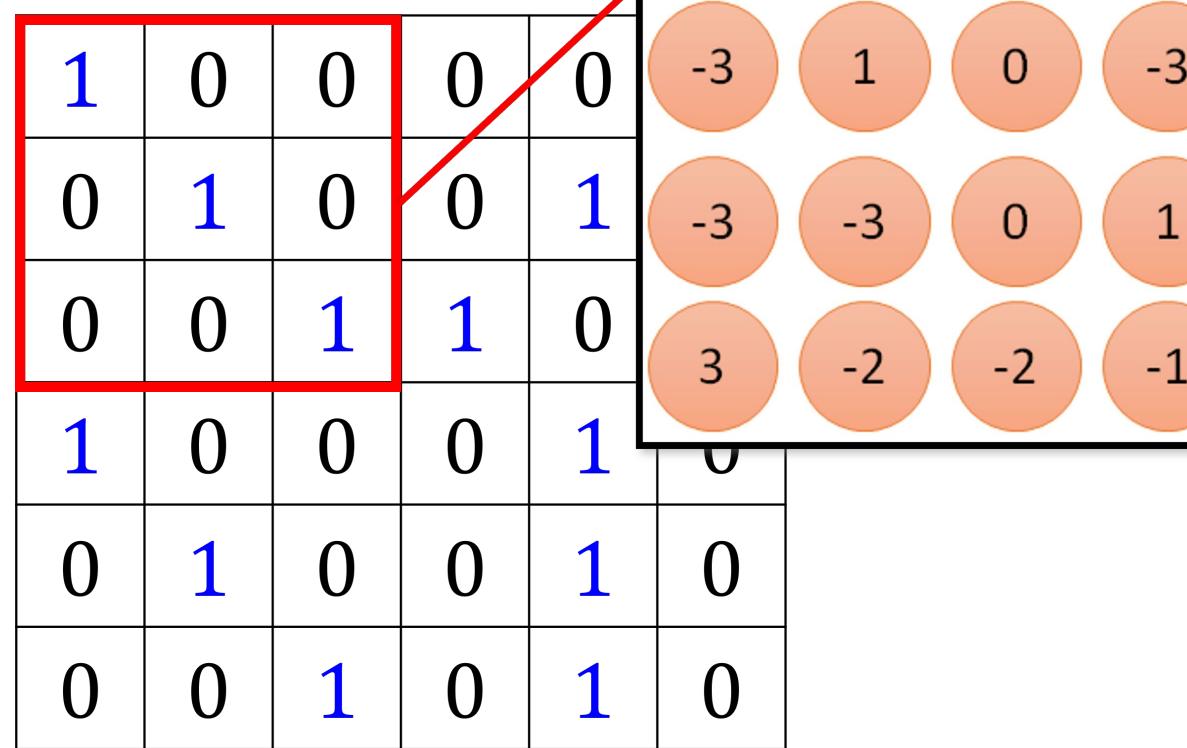
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Fully-connected



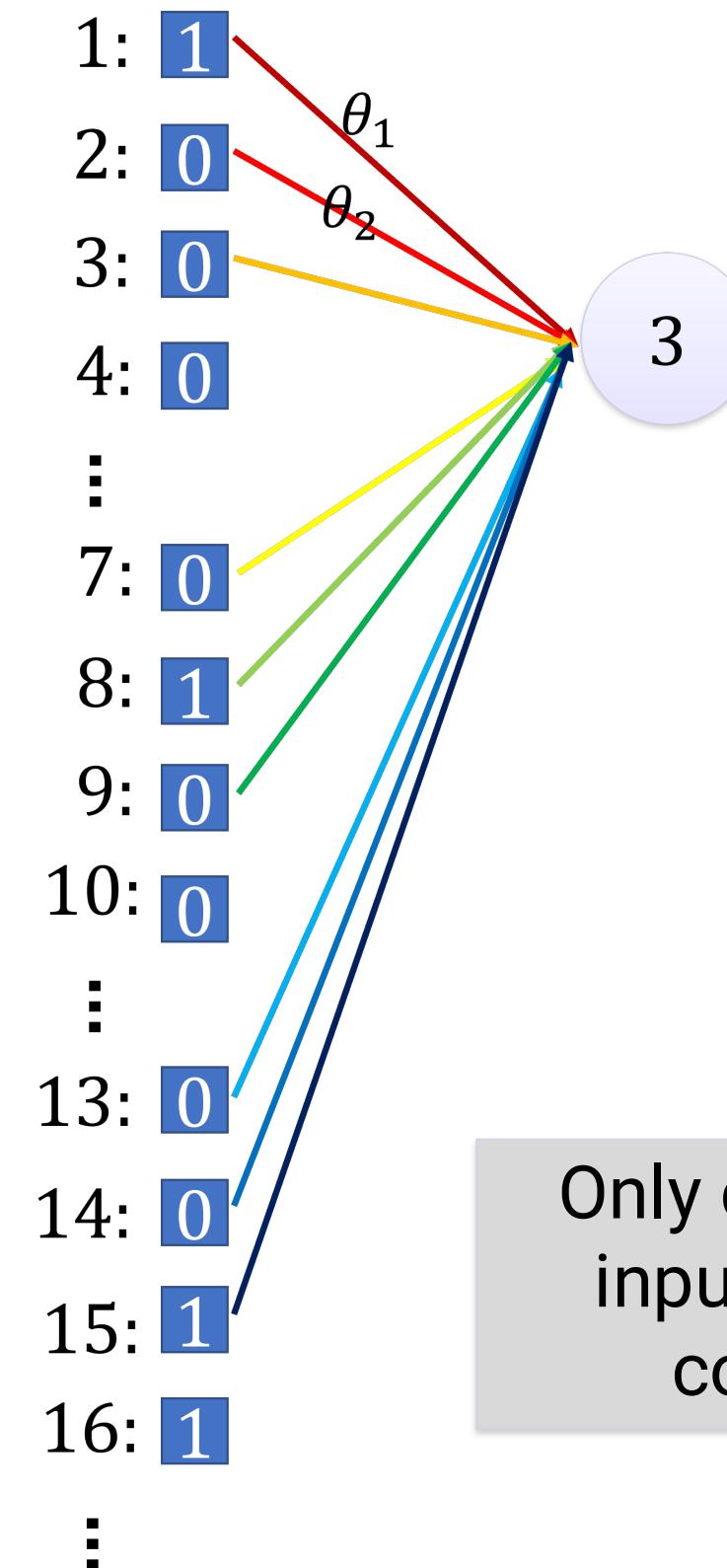


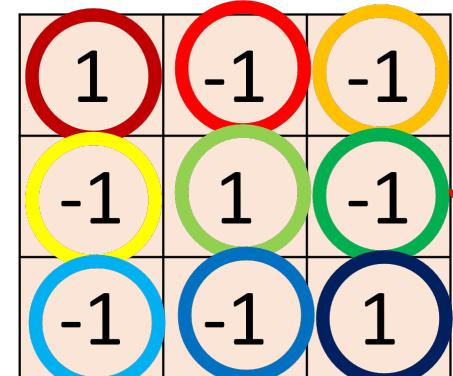
Filter 1



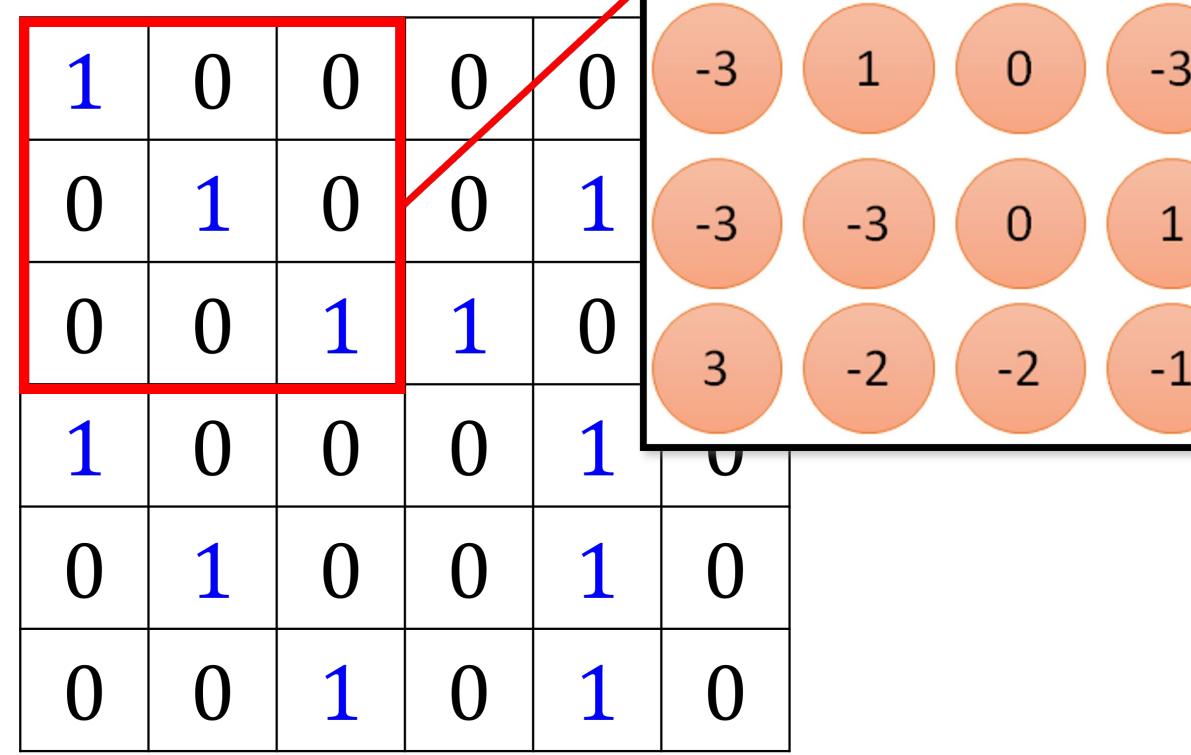
6 x 6 document

Fewer parameters!



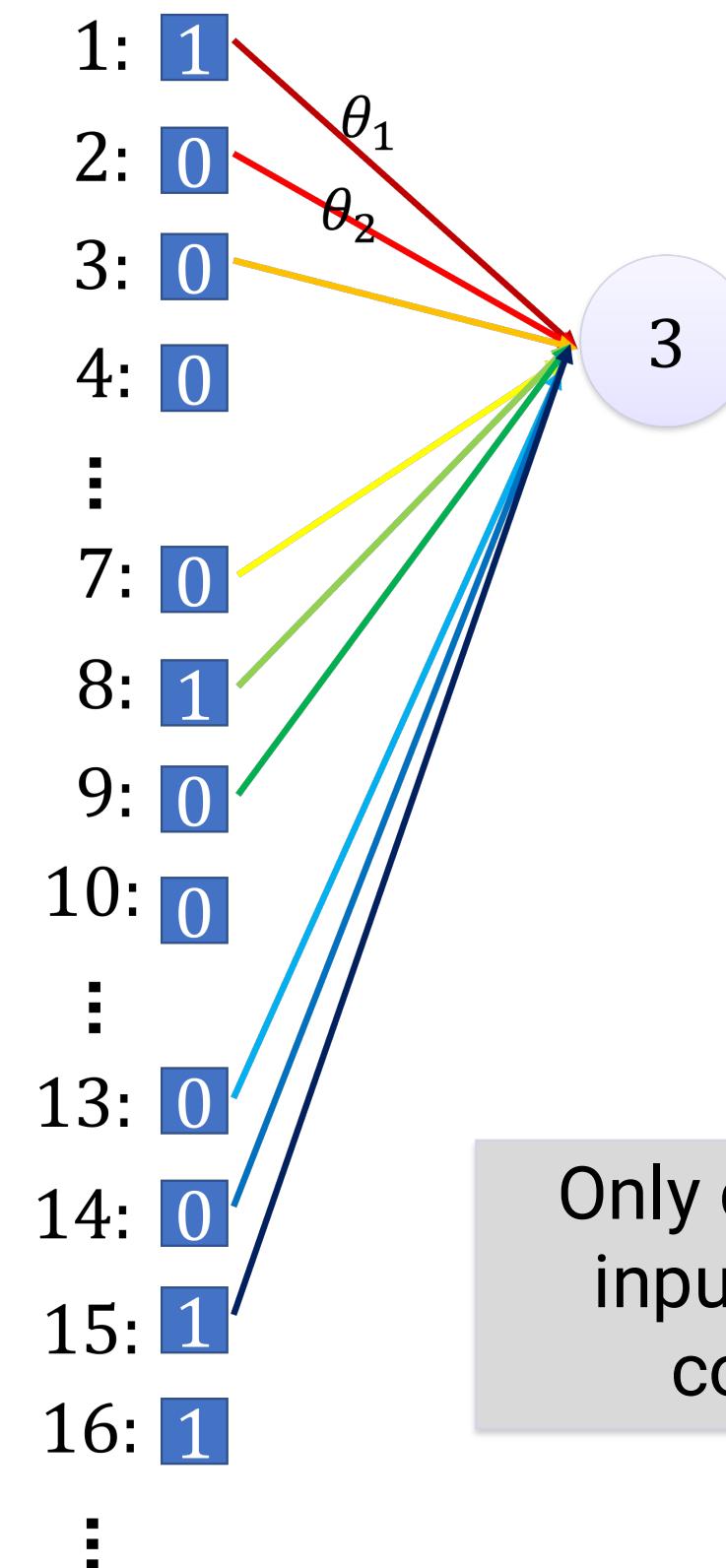


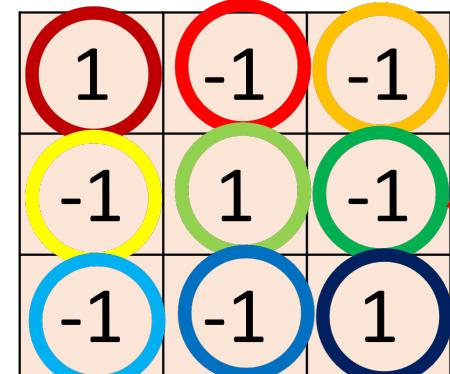
Filter 1



6 x 6 document

Fewer parameters!





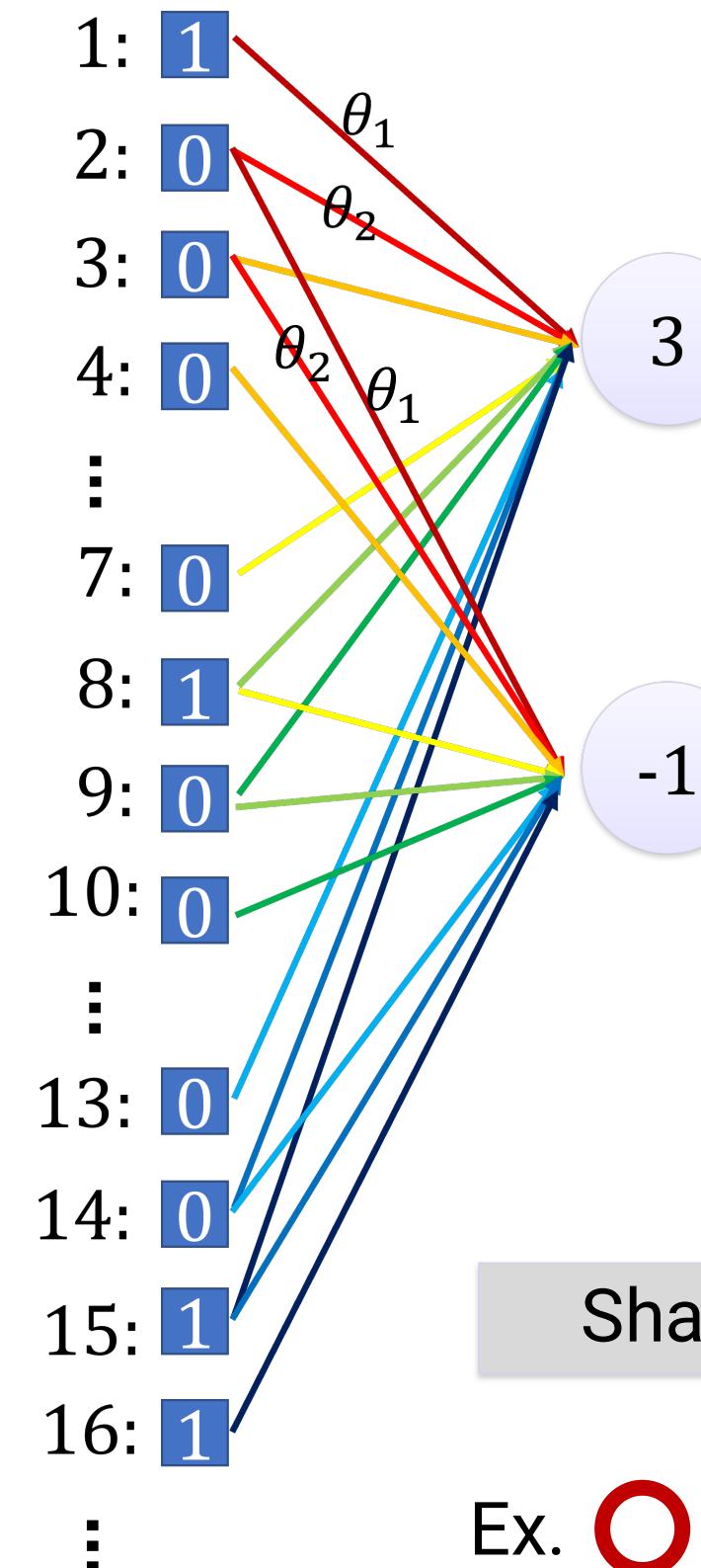
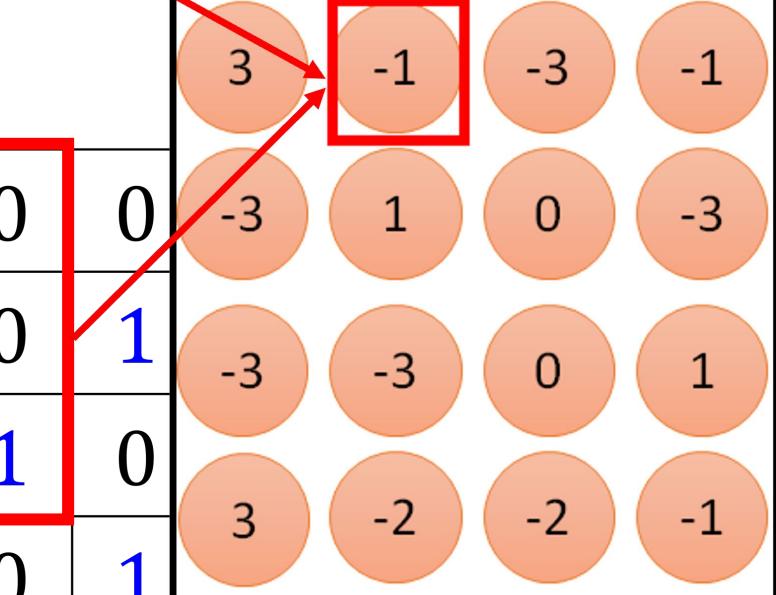
Filter 1

1	0	0	0	0	0
0	1	0	0	1	
0	0	1	1	0	
1	0	0	0	1	
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 document

Fewer parameters!

Even fewer parameters

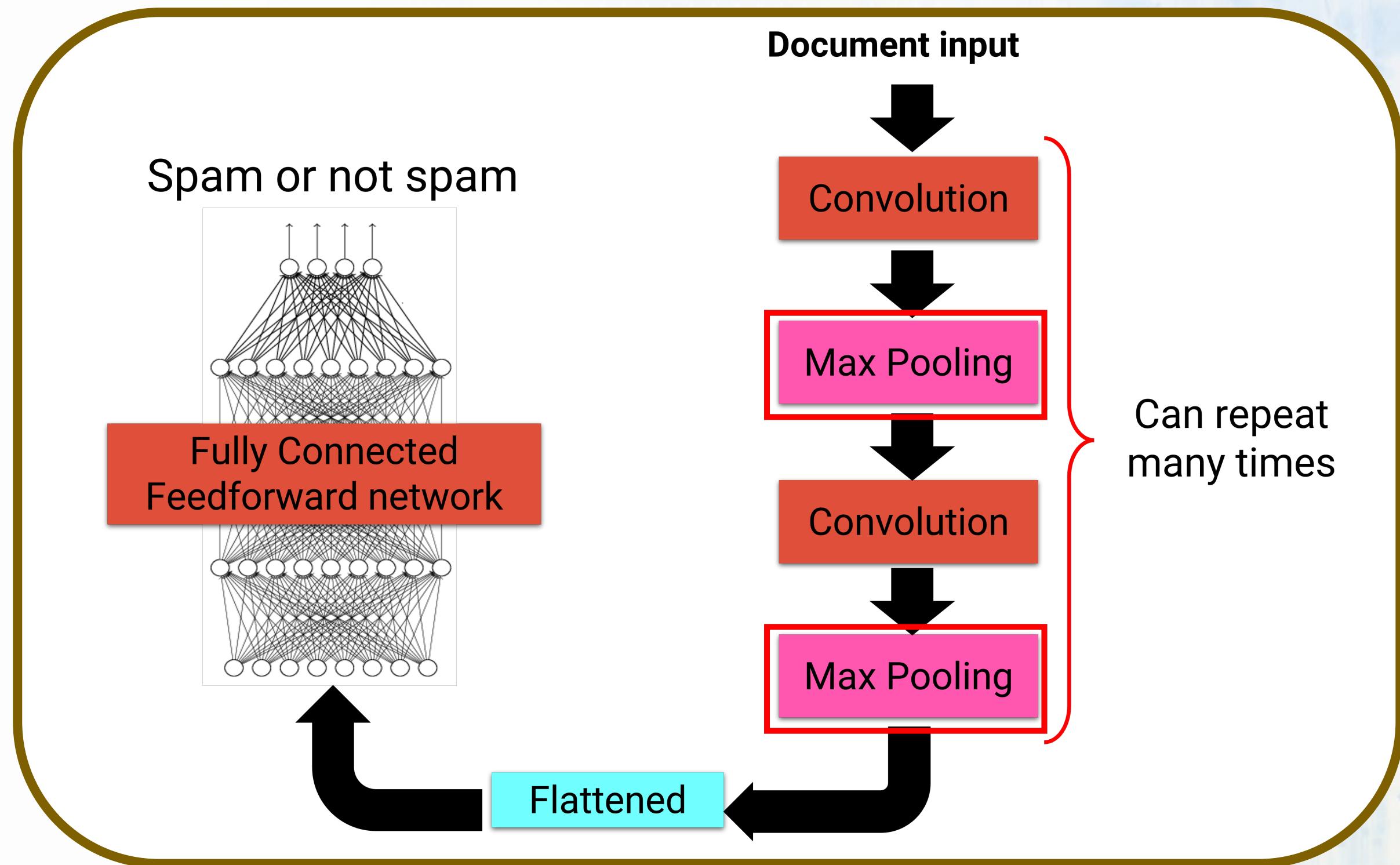


Shared weights

Ex. constrained to be identical



# The Whole CNN



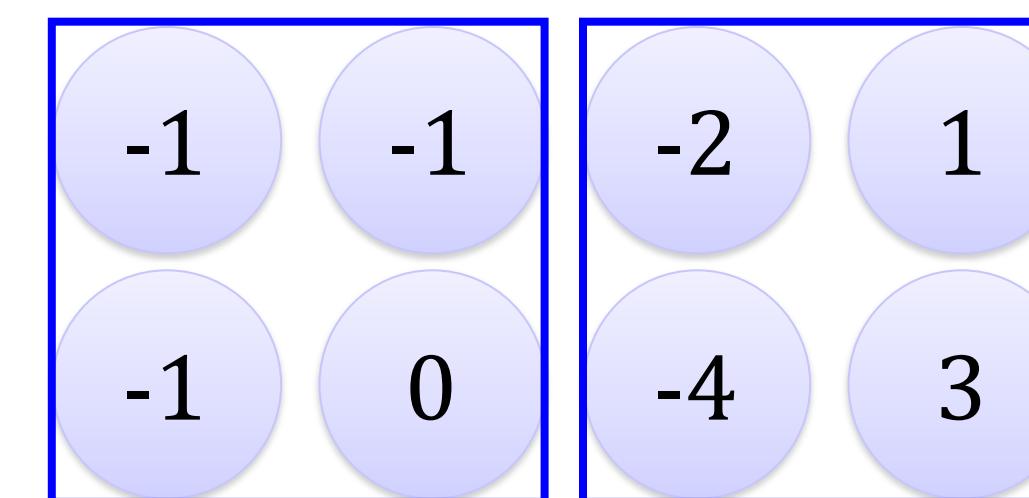
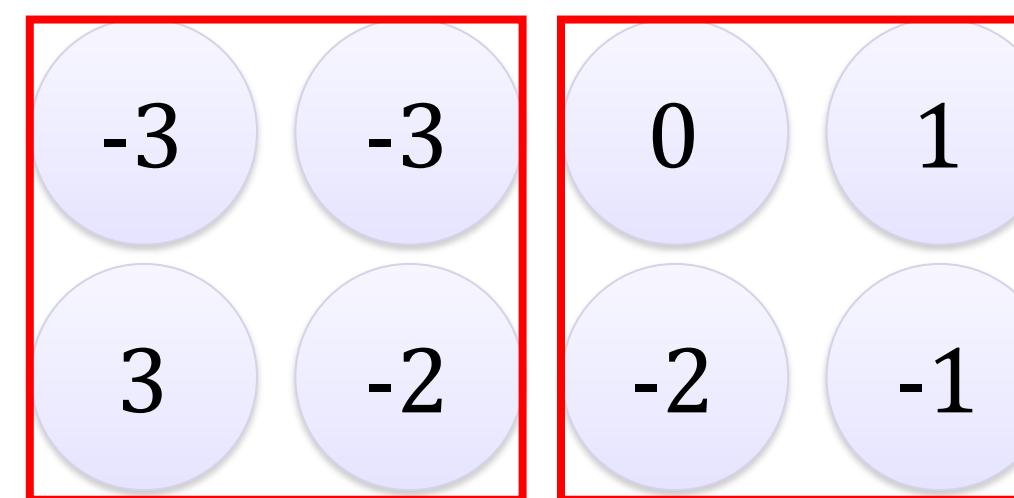
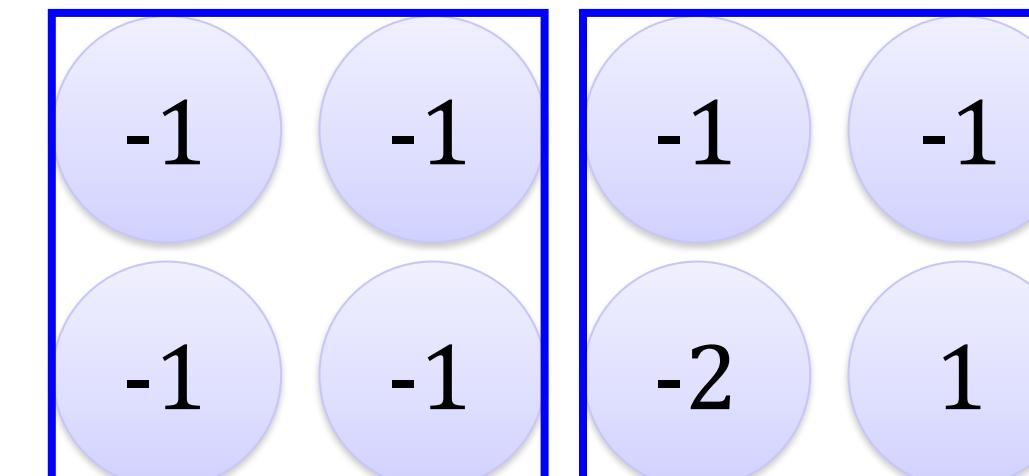
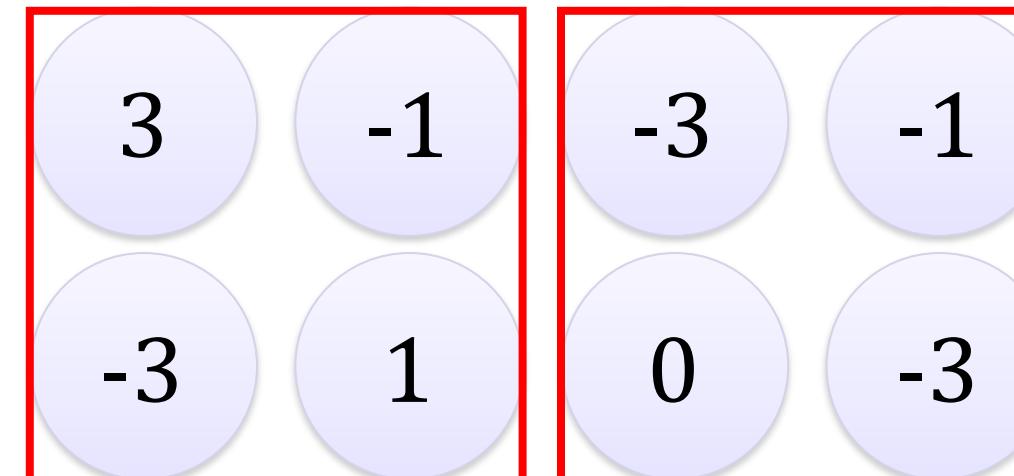
# Max Pooling

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

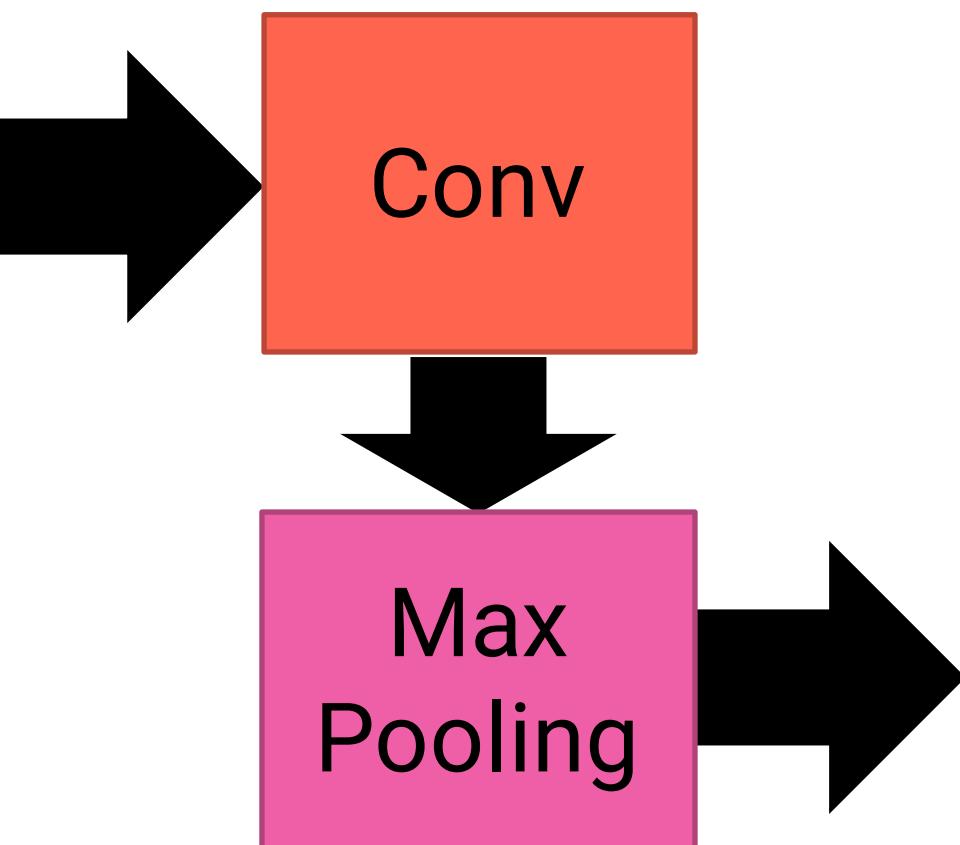
Filter 2



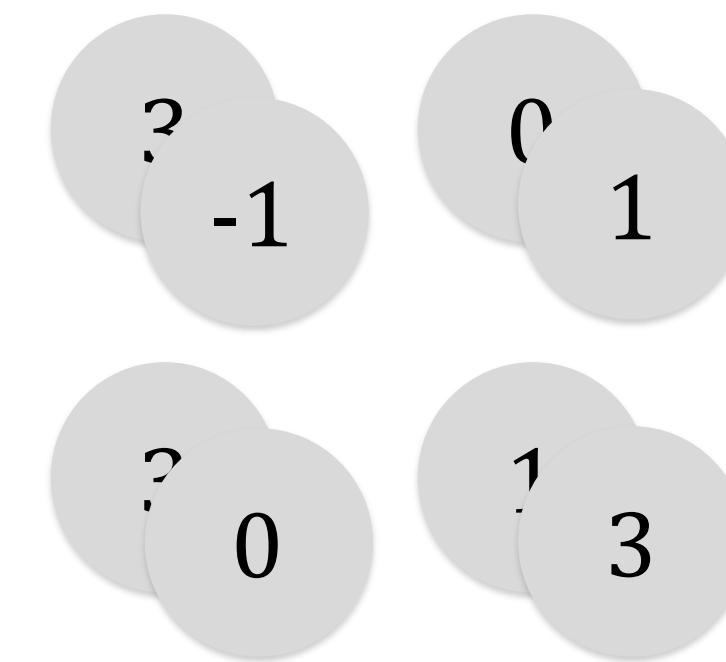
# Max Pooling

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 document



New document  
but smaller



2 x 2 document  
Each filter  
is a channel

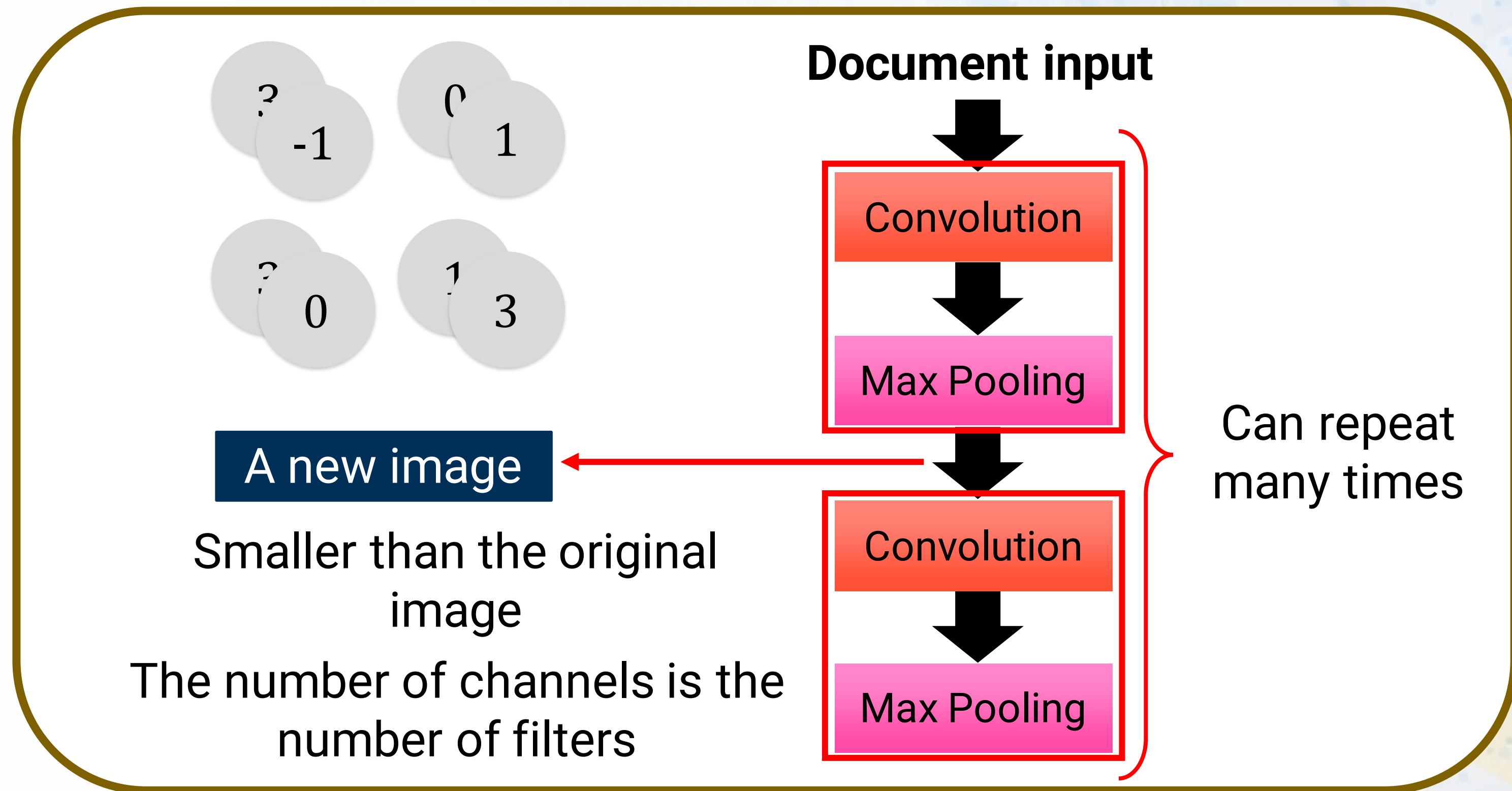


# A CNN Compresses a Fully Connected Network in Three Ways:

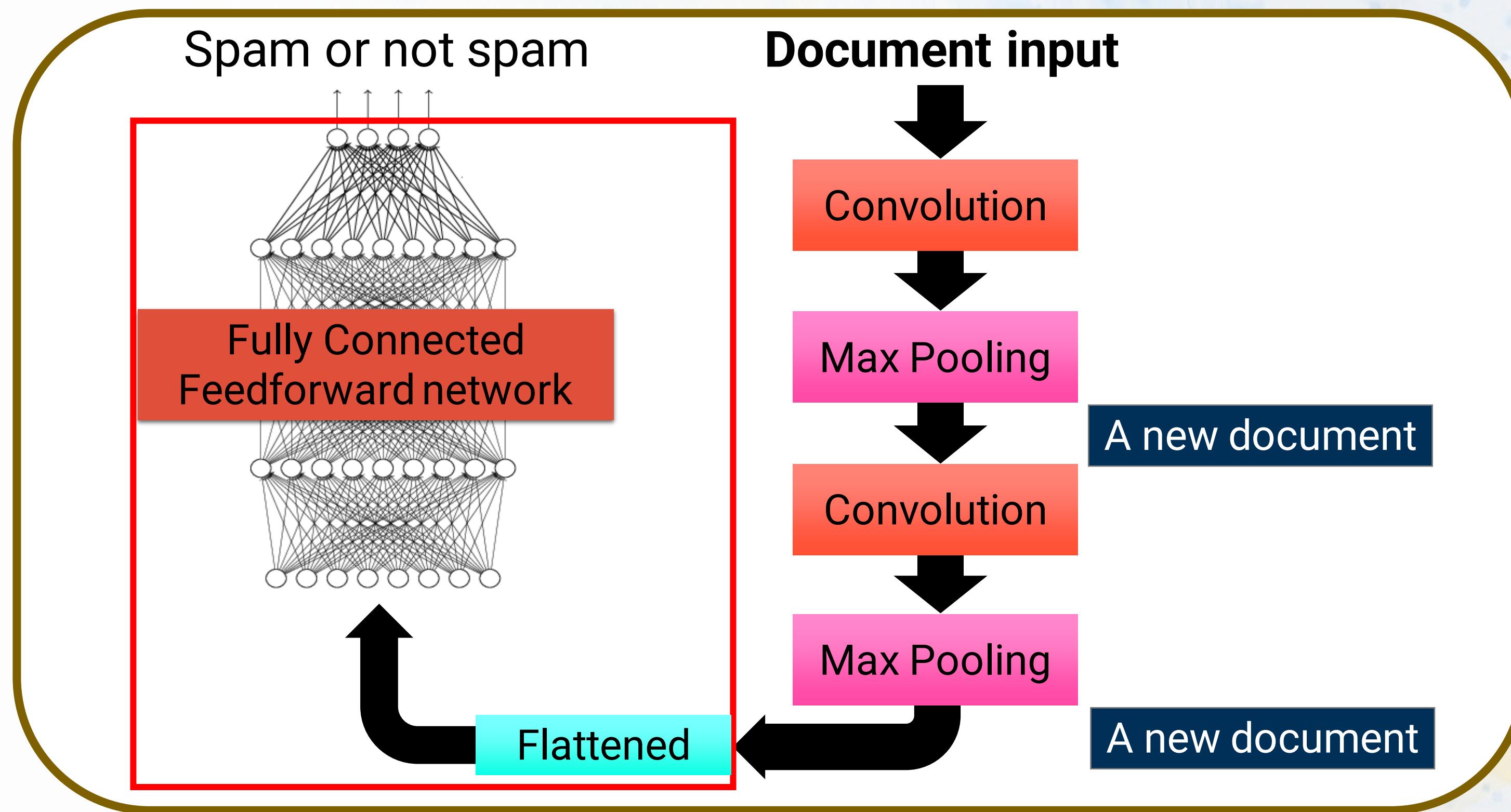
- Reducing the number of connections
- Shared weights on the edges
- Max pooling further reduces the complexity



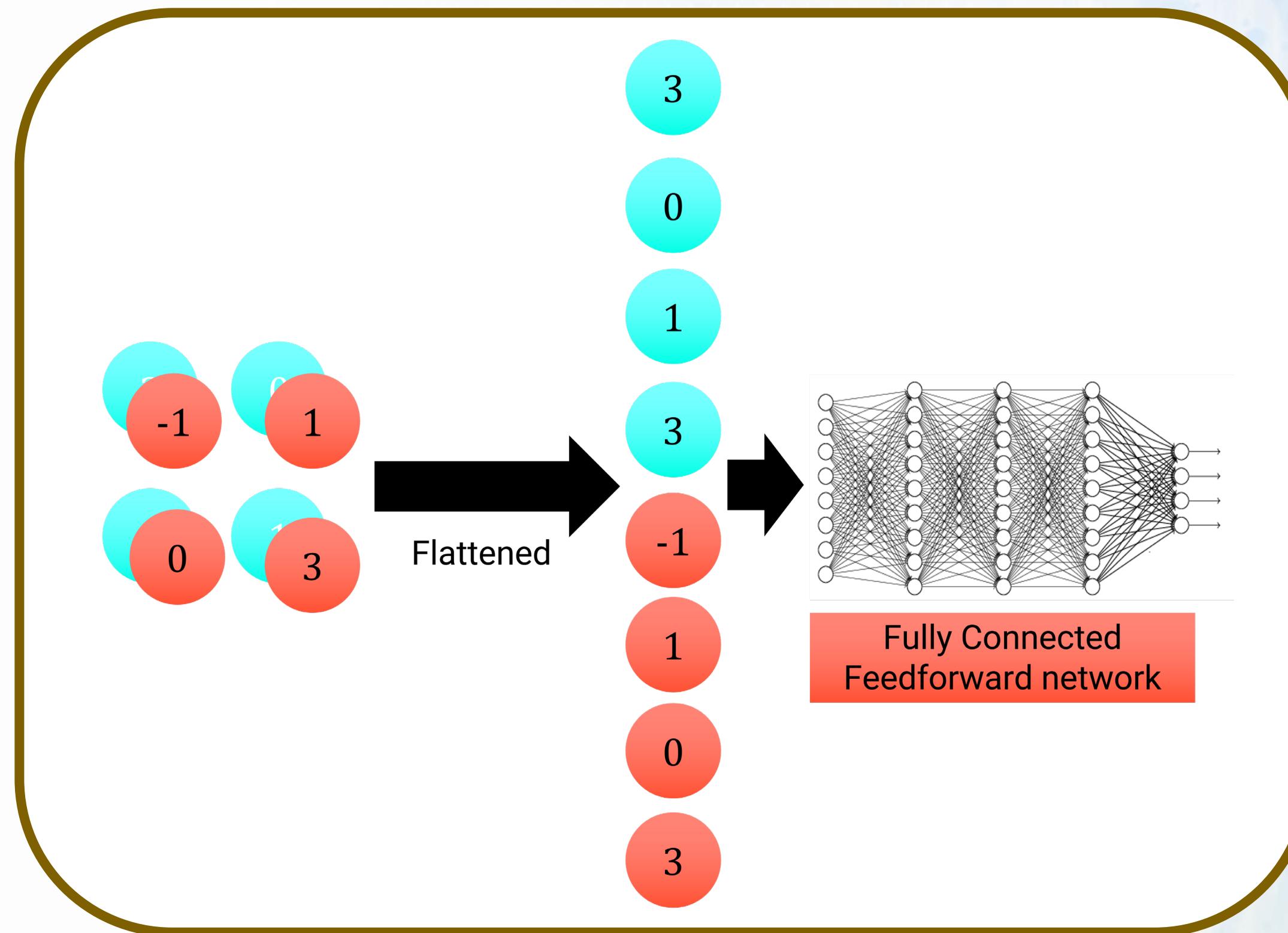
# The Whole CNN



# The Whole CNN



# Flattening



# CNN in Keras

Only modified the *network structure* and *input format* (vector -> 3-D array)

```
Model2.add( Convolution2D (25,3,3)  
Input_shape = (24,20,1)
```

$$\begin{array}{|c|c|c|} \hline 1 & -1 & 1 \\ \hline -1 & 1 & -1 \\ \hline -1 & -1 & 1 \\ \hline \end{array}$$

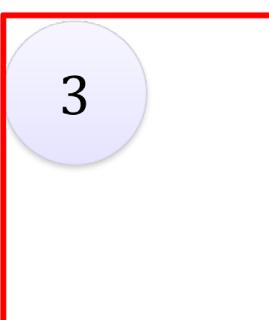
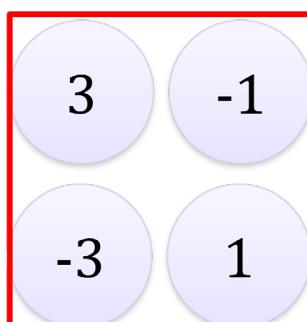
...  
...  
...

Input\_shape = ( 24, 20, 1 )

24 x 20 matrix

1: one channel matrix

```
model2.add (MaxPooling2D ( (2,2) ))
```



Input



Convolution



Max Pooling



Convolution

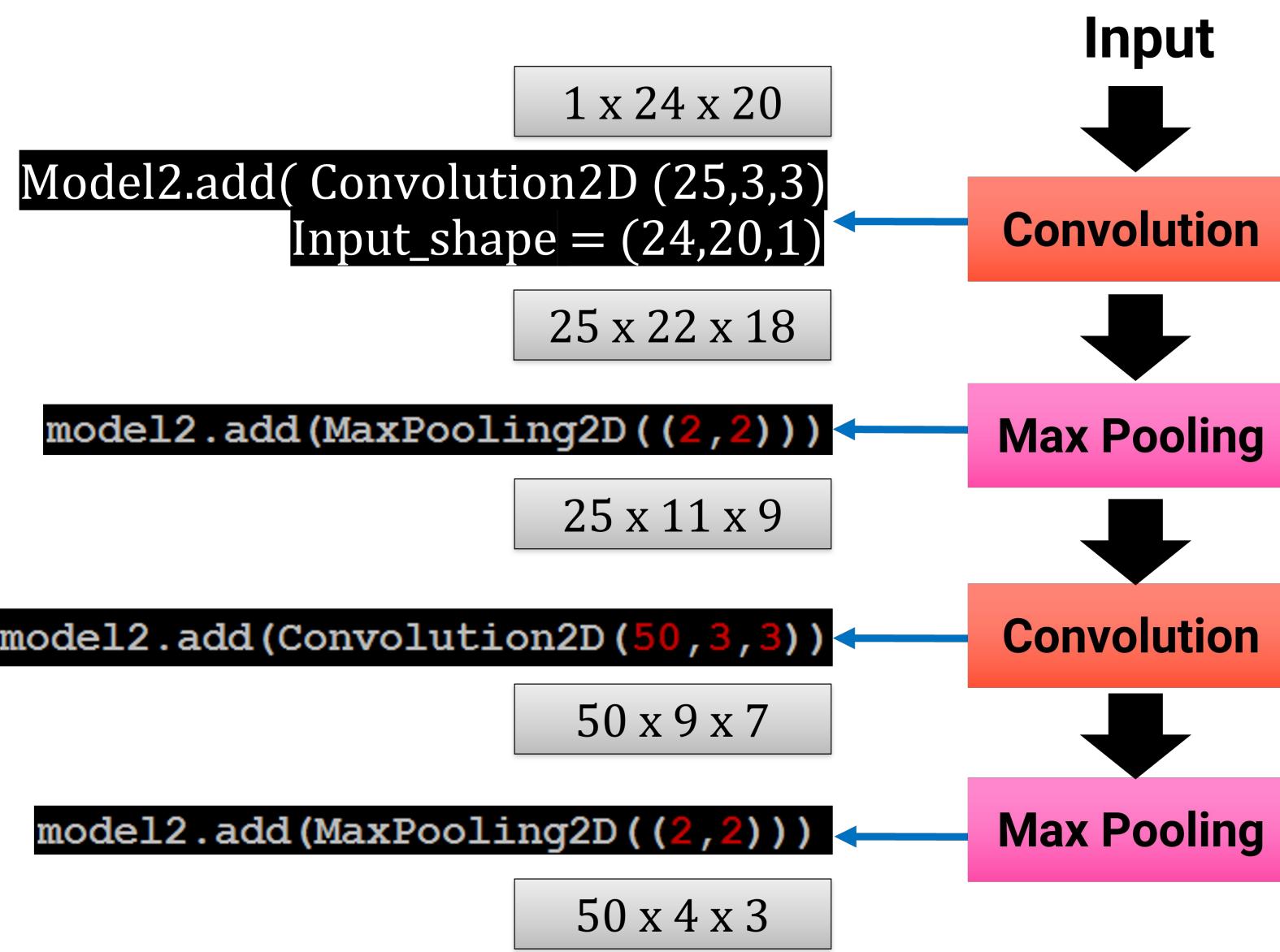


Max Pooling



# CNN in Keras

Only modified the *network structure* and *input format* (vector -> 3-D array)



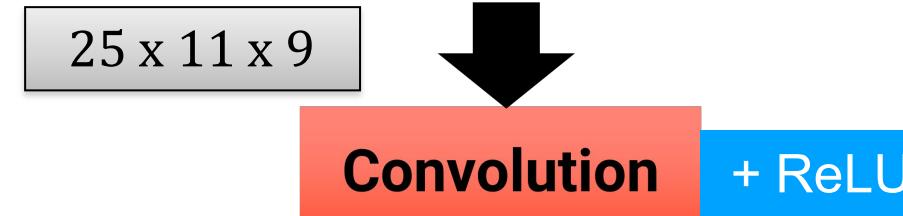
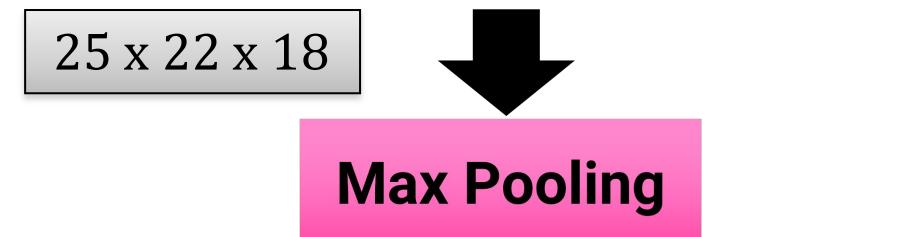
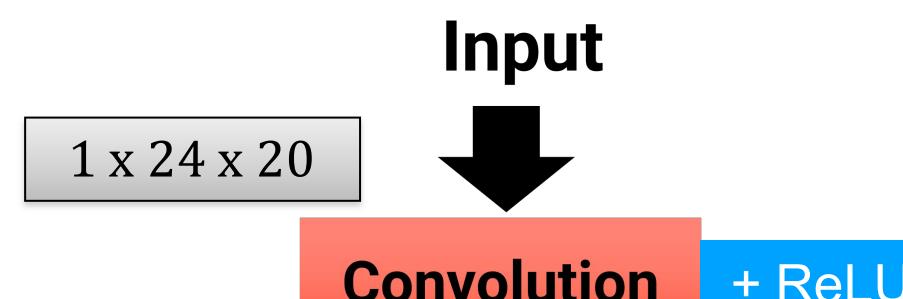
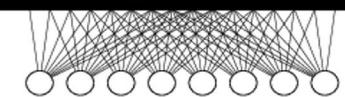
# CNN in Keras

Only modified the *network structure* and *input format* (vector -> 3-D array)

Spam or not spam

Fully connected feedforward network

```
model2.add(Dense(output_dim=100))  
model2.add(Activation('relu'))  
model2.add(Dense(output_dim=10))  
model2.add(Activation('softmax'))
```



600

Flattened

```
model2.add(Flatten())
```



# Summary

- We learned about Deep Neural Network
- We went over the CNN model and convolution part
- We converted a corpus into a matrix which can be fed into CNN
- We used Keras library to create a CNN architecture

