AI: Convolutional Neural Networks

CPSC 501: Advanced Programming Techniques
Fall 2020

Jonathan Hudson, Ph.D
Instructor
Department of Computer Science
University of Calgary

Wednesday, August 12, 2020

UNIVERSITY OF CALGARY

ImageNet

Deep Learning/ Convolutional Neural Networks

- Classify an image into 1000 possible classes:
  e.g. Abyssinian cat, Bulldog, French Terrier, Cormorant, Chickadee, red fox, banjo, barbell, hourglass, knot, maze, viaduct, etc.

→

cat, tabby cat (0.71)
Egyptian cat (0.22)
red fox (0.11)
.....

The Data: ILSVRC

- Imagenet Large Scale Visual Recognition Challenge (ILSVRC): Annual Competition

1000 Categories

~1000 training images per Category

~1 million images in total for training

~50k images for validation

Only images released for the test set but no annotations,
evaluation is performed centrally by the organizers (max 2 per week)

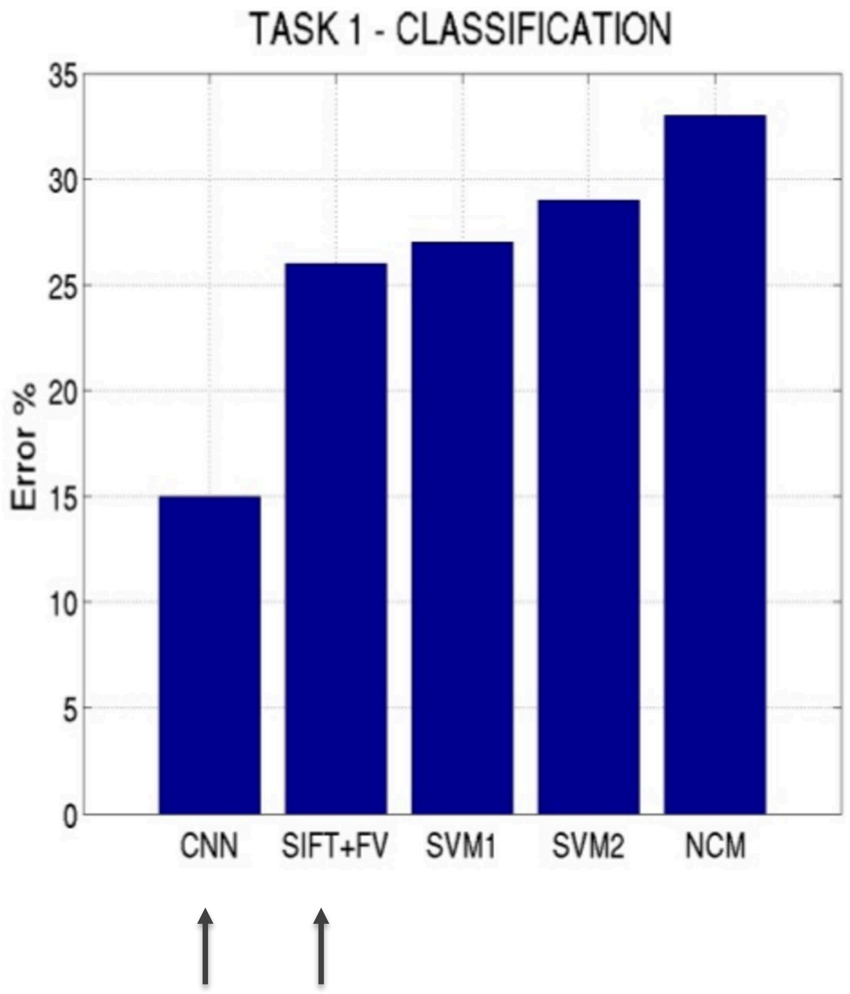
The Evaluation Metric: Top K-error

True label: Abyssinian cat

Top-1 error: 1.0	Top-1 accuracy: 0.0
Top-2 error: 1.0	Top-2 accuracy: 0.0
Top-3 error: 1.0	Top-3 accuracy: 0.0
Top-4 error: 0.0	Top-4 accuracy: 1.0
Top-5 error: 0.0	Top-5 accuracy: 1.0

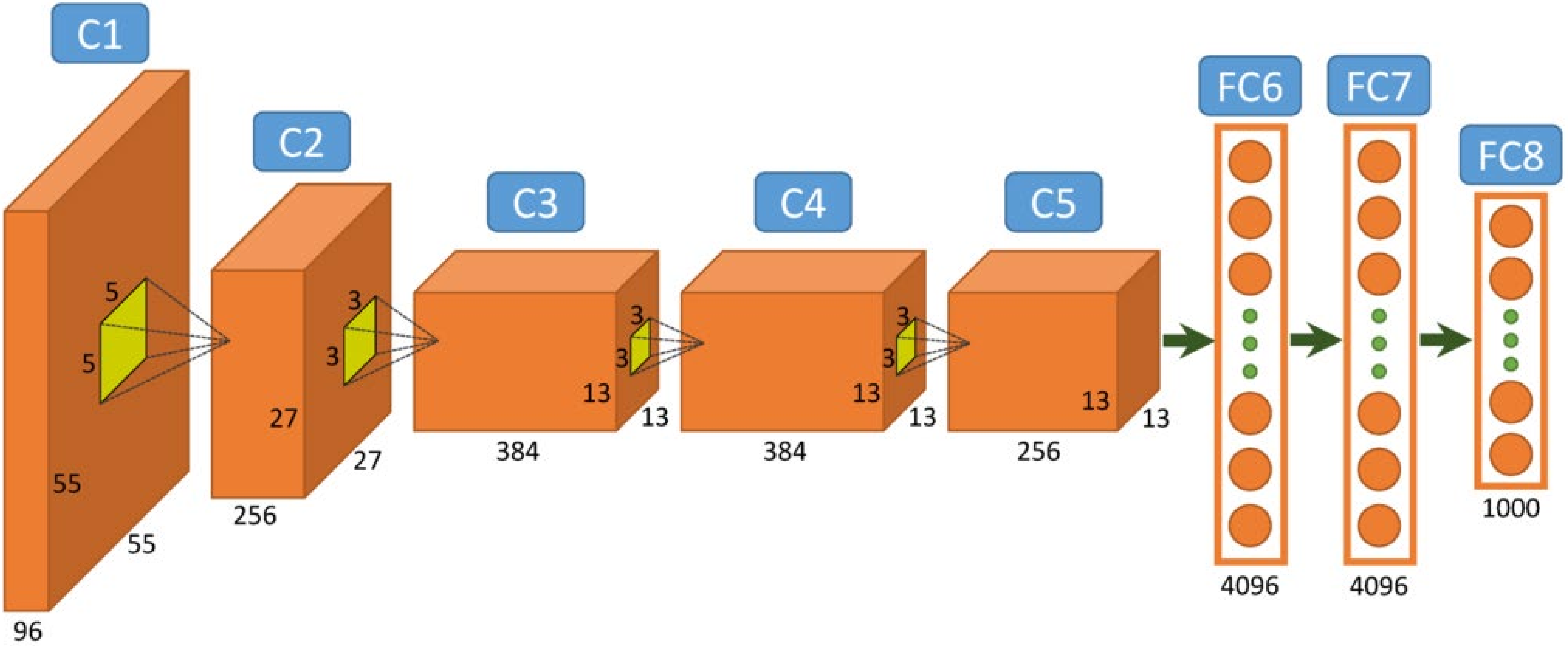
cat, tabby cat (0.61)
Egyptian cat (0.22)
red fox (0.11)
Abyssinian cat (0.10)
French terrier (0.03)
…..

Top-5 error on this competition (2012)



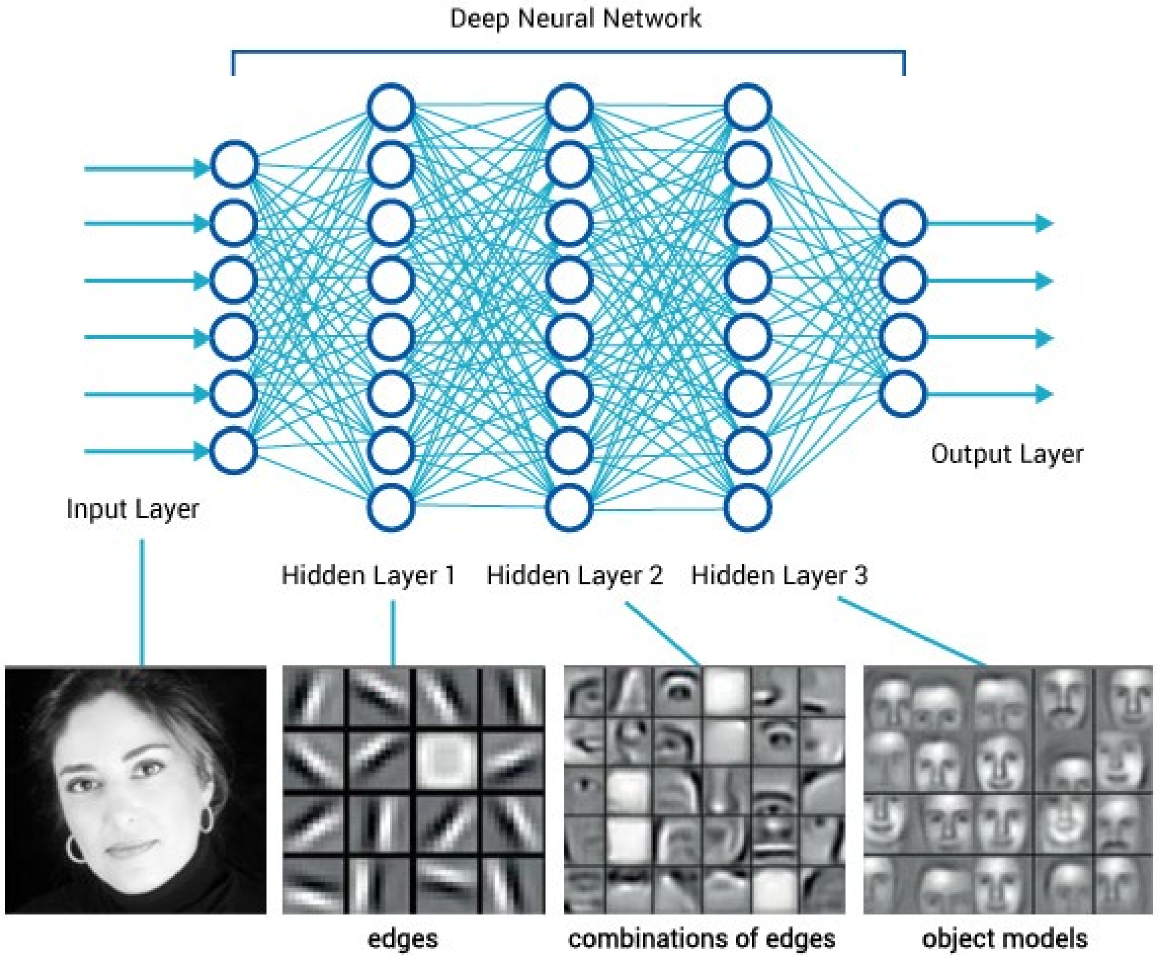
AlexNet

Alexnet



https://www.saagie.com/fr/blog/object-detection-part1

What is happening?



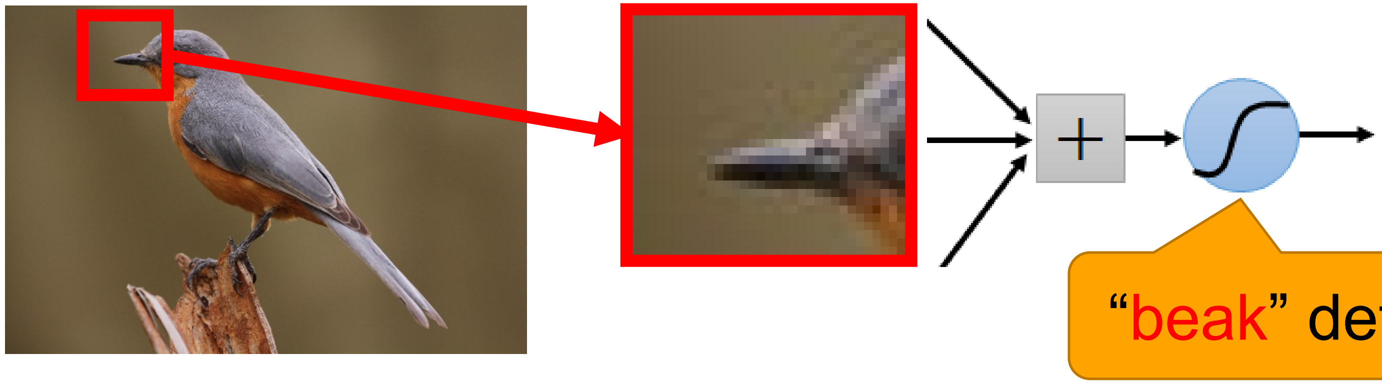
https://www.saagie.com/fr/blog/object-detection-part1

Convolution

Consider learning an image:

- Some patterns are much smaller than the whole image

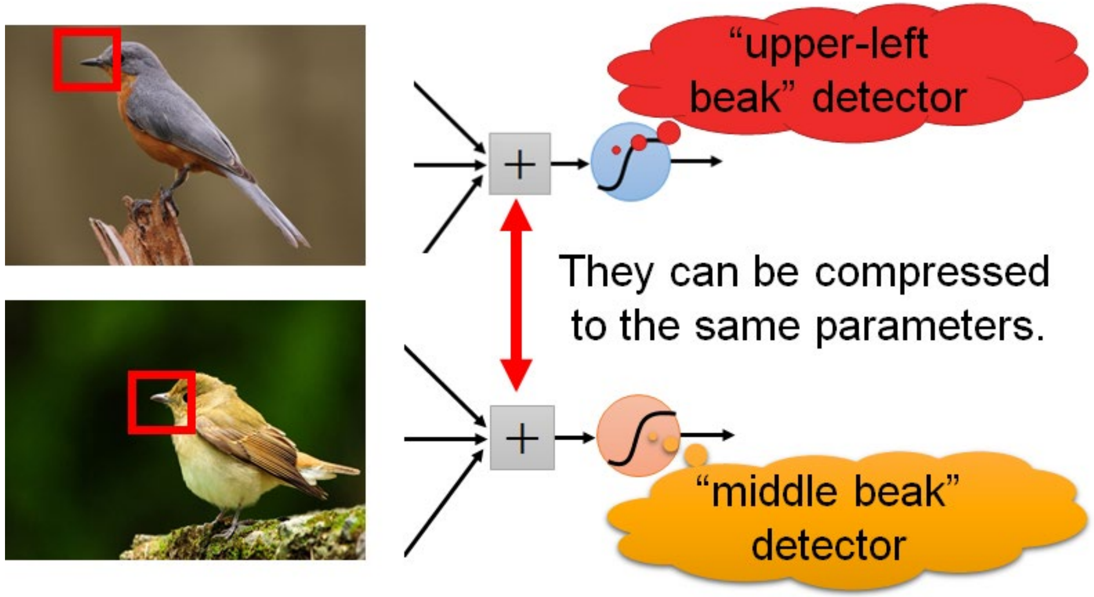
Can represent a small region with fewer parameters



"beak" detector

Detectors

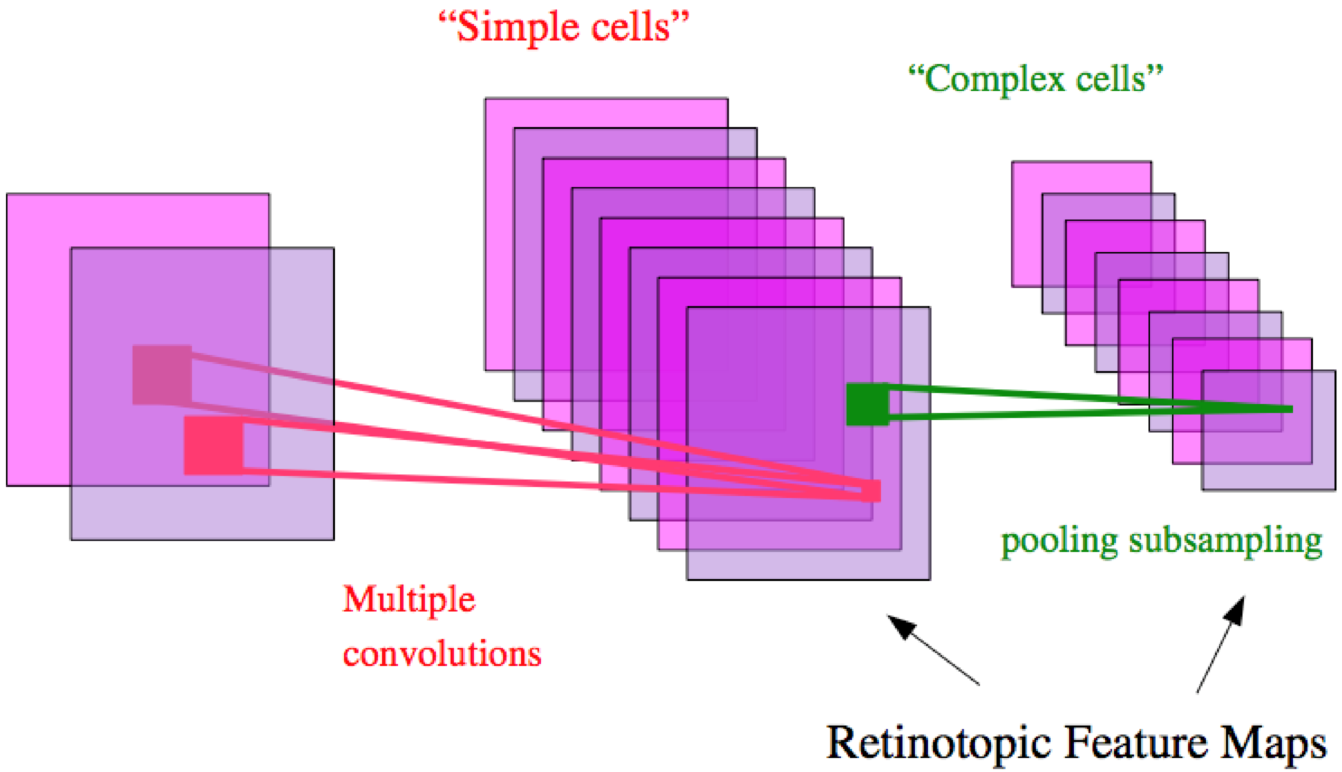
- Same pattern appears in different places: They can be compressed! What about training a lot of such "small" detectors and each detector must "move around".



"upper-left beak" detector
They can be compressed to the same parameters.
"middle beak" detector

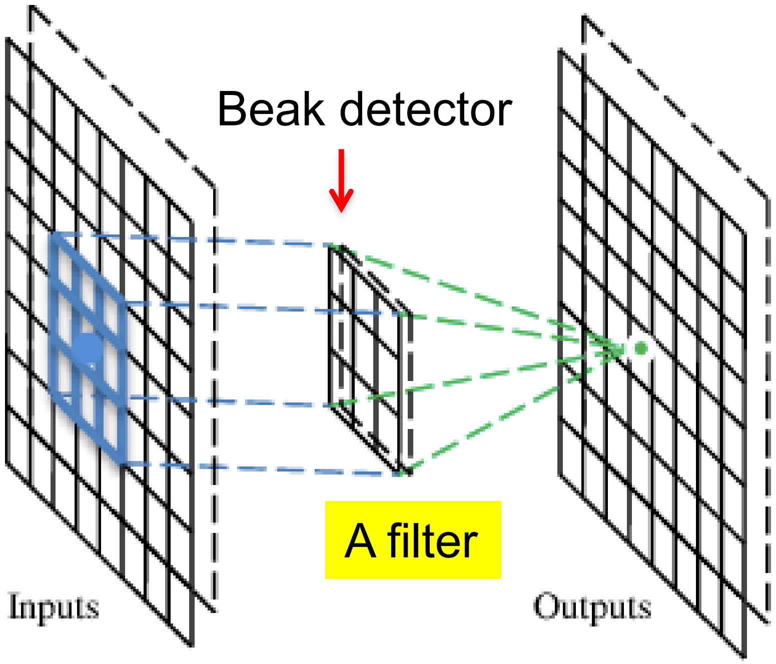
Model of vision in animals

- [Hubel & Wiesel 1962]:
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.



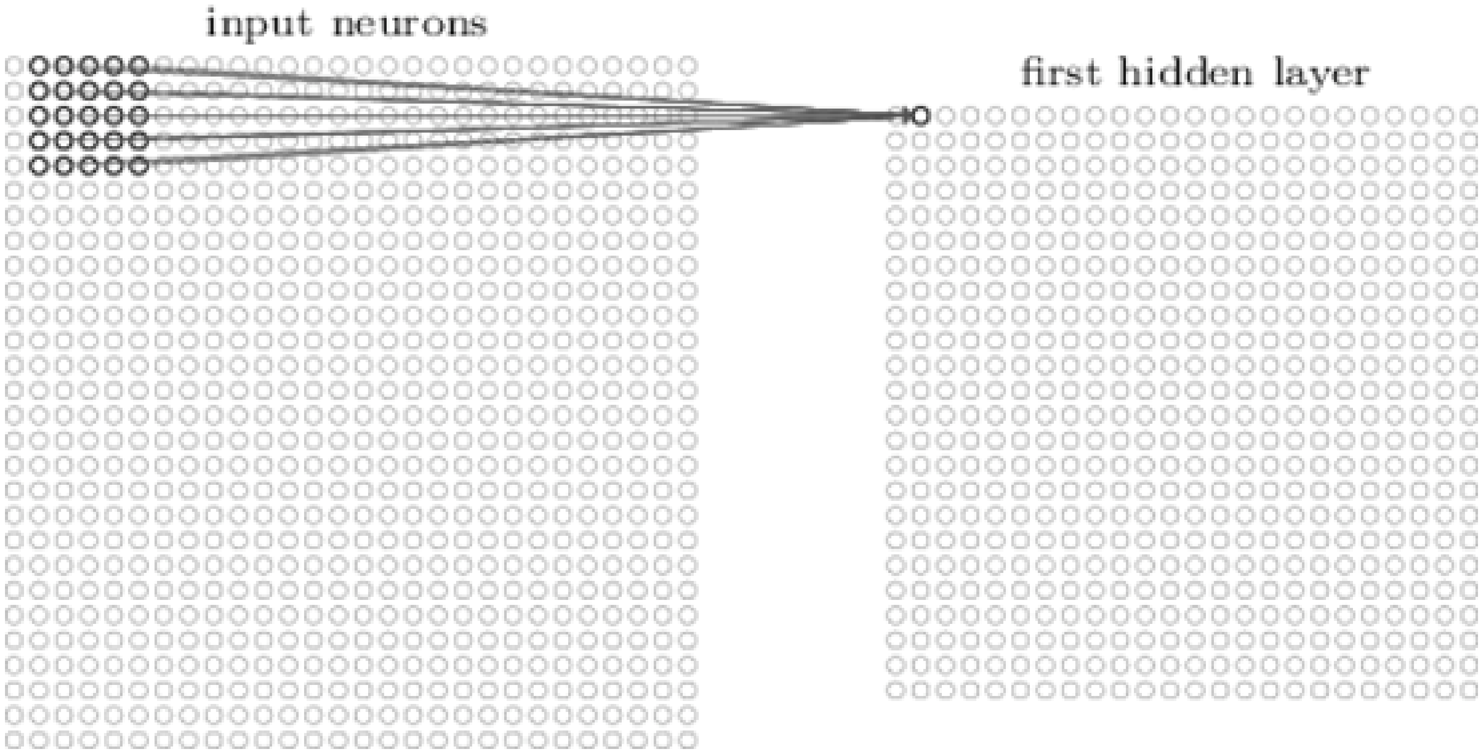
A convolutional layer

- A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation.

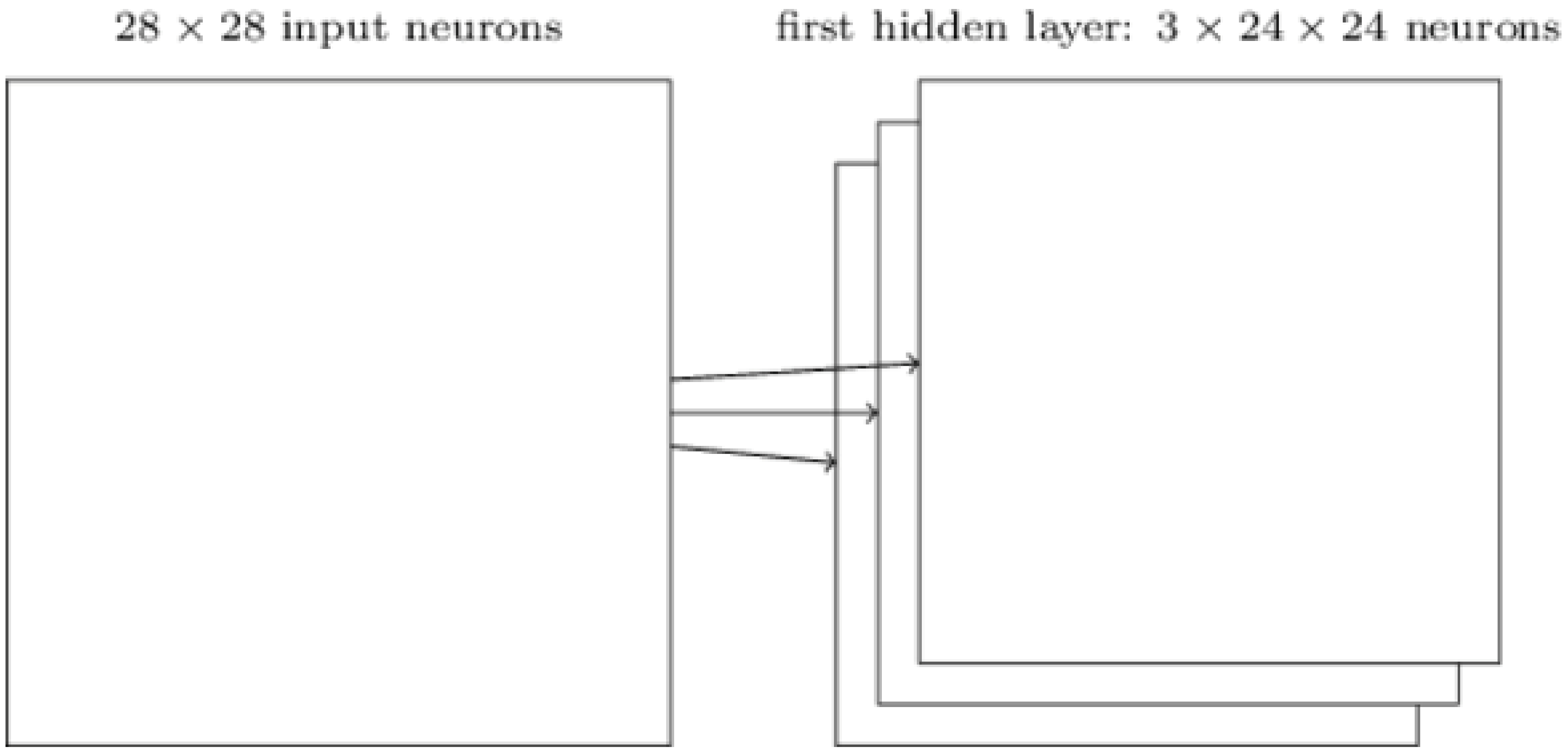


How do we convolve an image with an ANN?

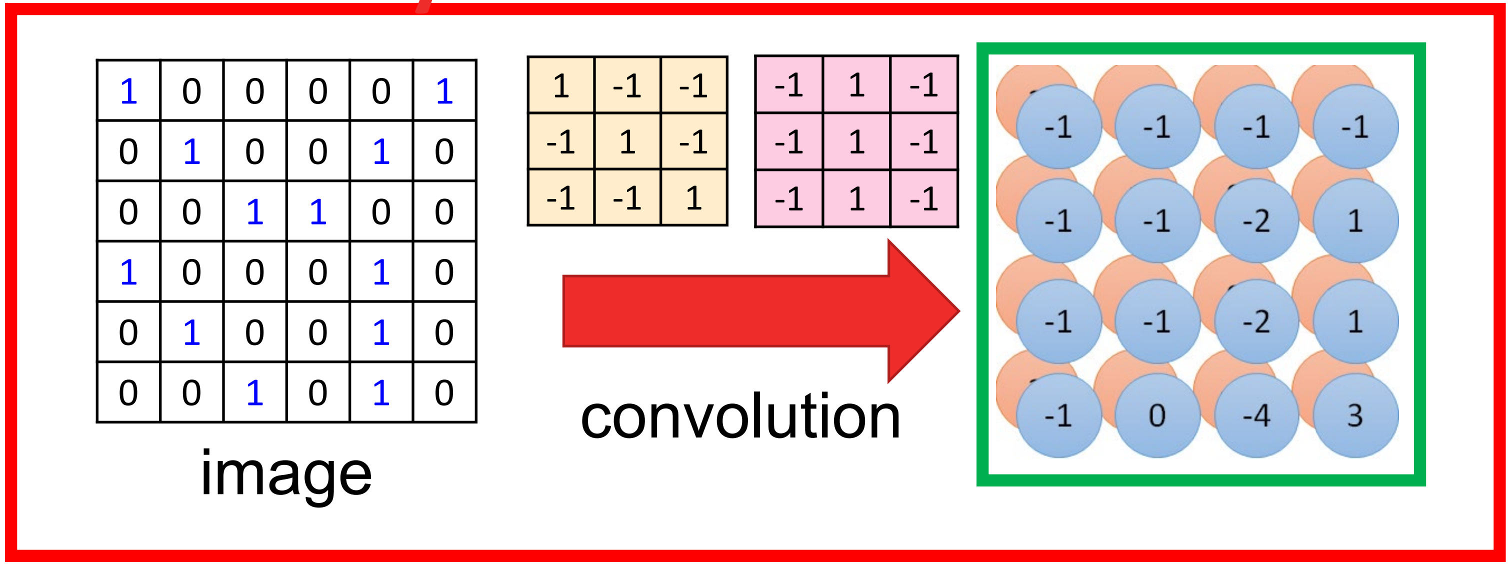
Note that the parameters in the matrix defining the convolution are **tied** across all places that it is used



How do we do many convolutions of an image with an ANN?



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

image

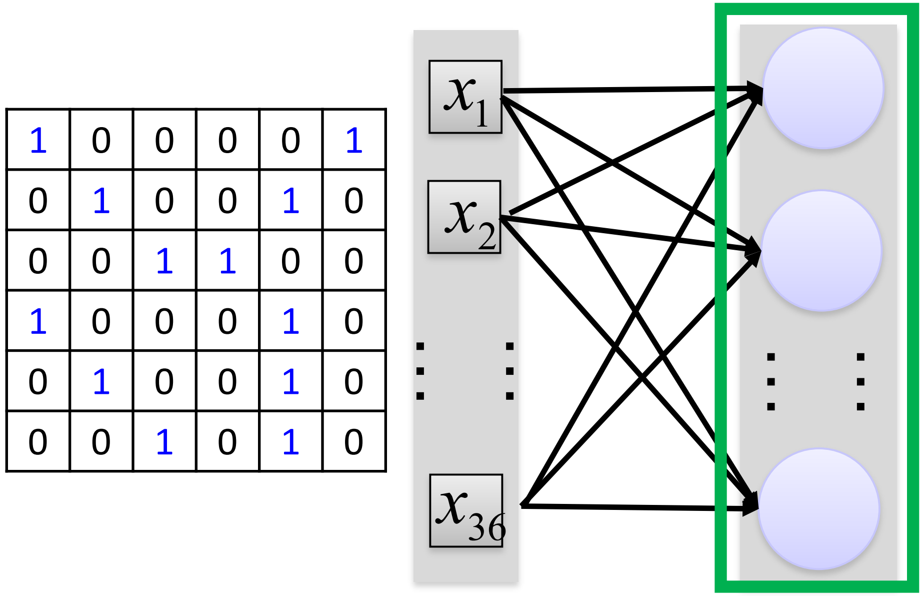
1	-1	-1
-1	1	-1
-1	-1	1

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

convolution

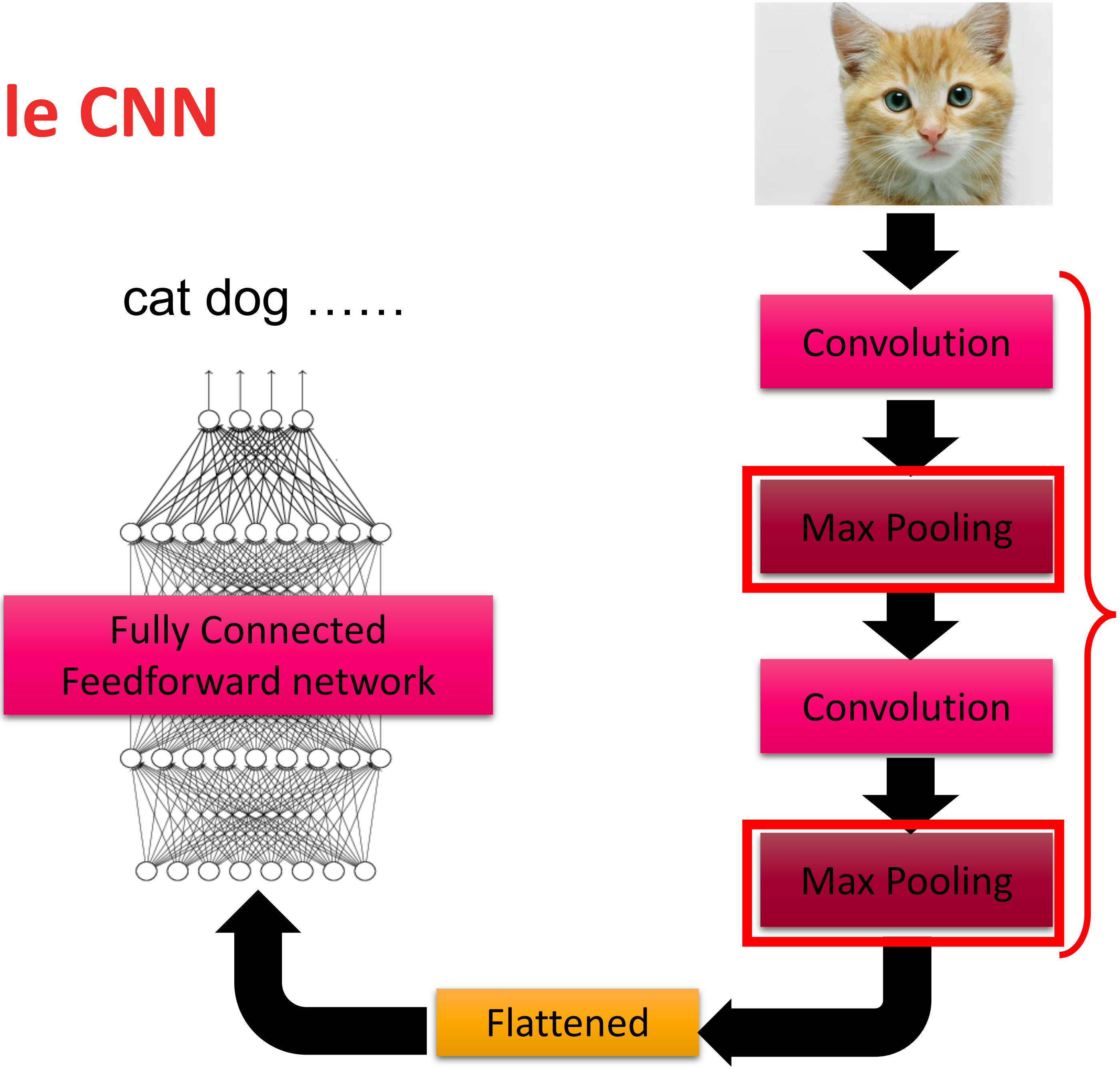
-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

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

The whole CNN



cat dog ……

Fully Connected Feedforward network

Convolution

Max Pooling

Convolution

Max Pooling

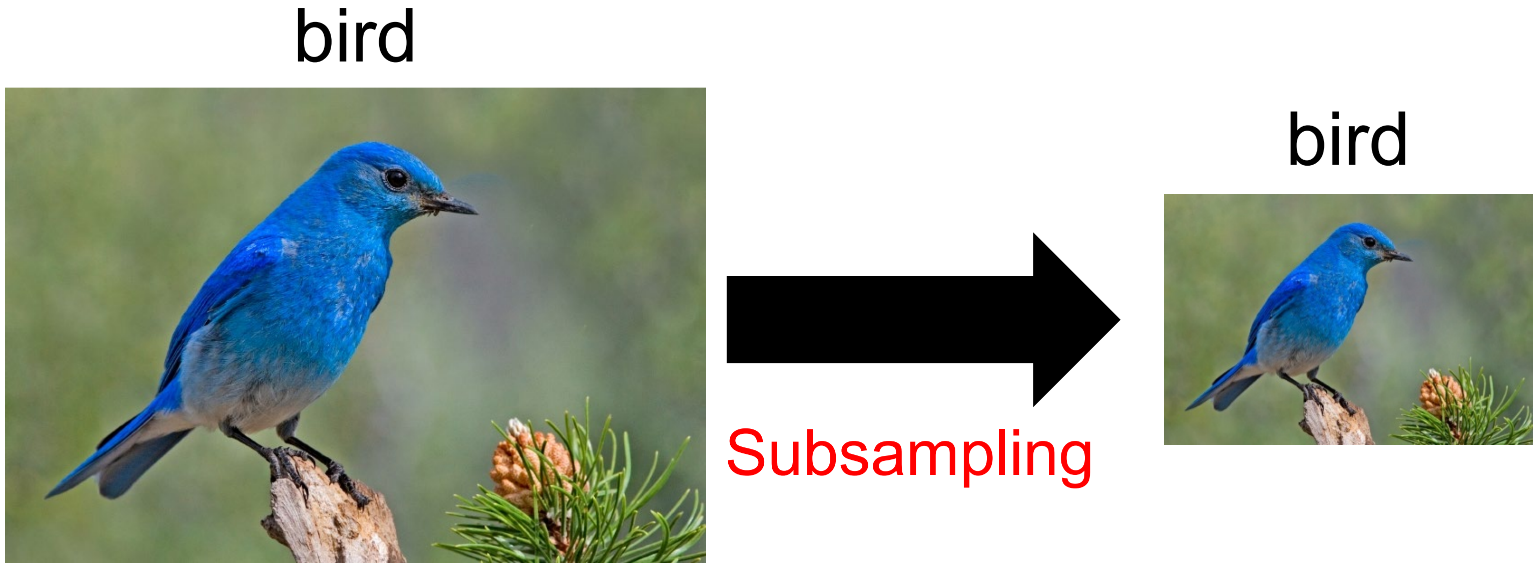
Flattened

Can repeat many times

Pooling

Why Pooling

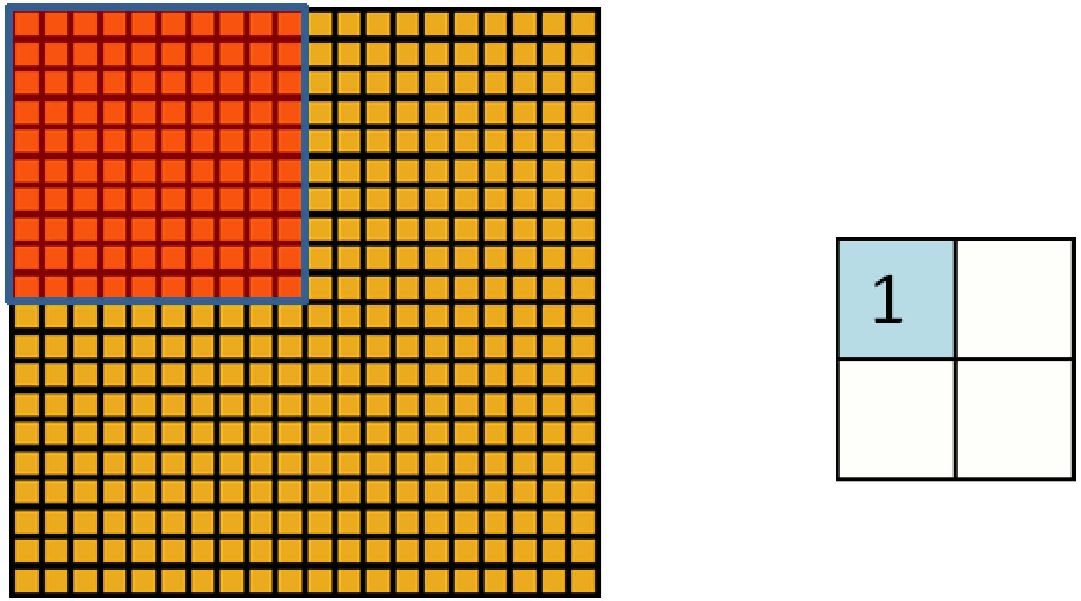
- Subsampling pixels will not change the object



We can subsample the pixels to make image smaller

→ fewer parameters to characterize the image

Pooling



Convolved feature

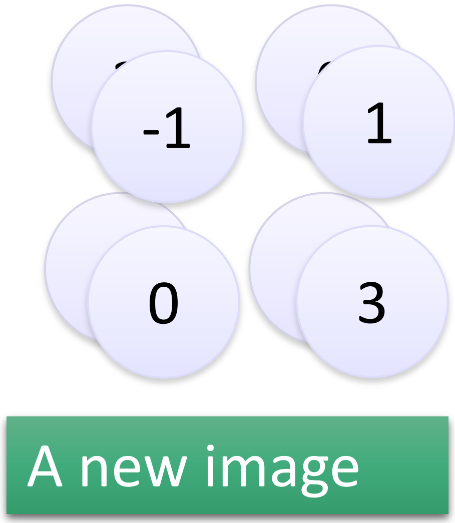
Pooled feature

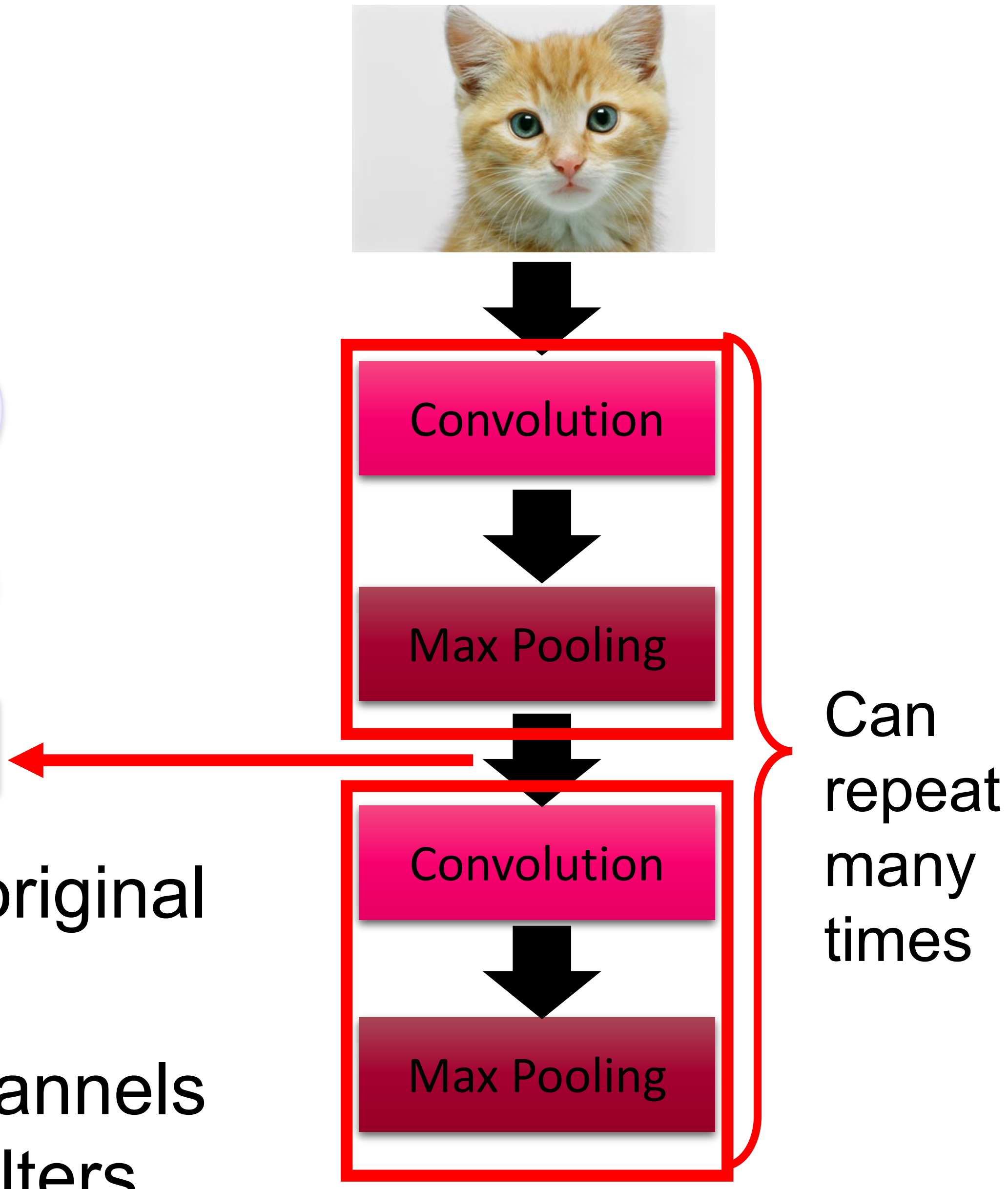
Full Convolution NN

A CNN compresses a fully connected network in two ways:

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

The whole CNN

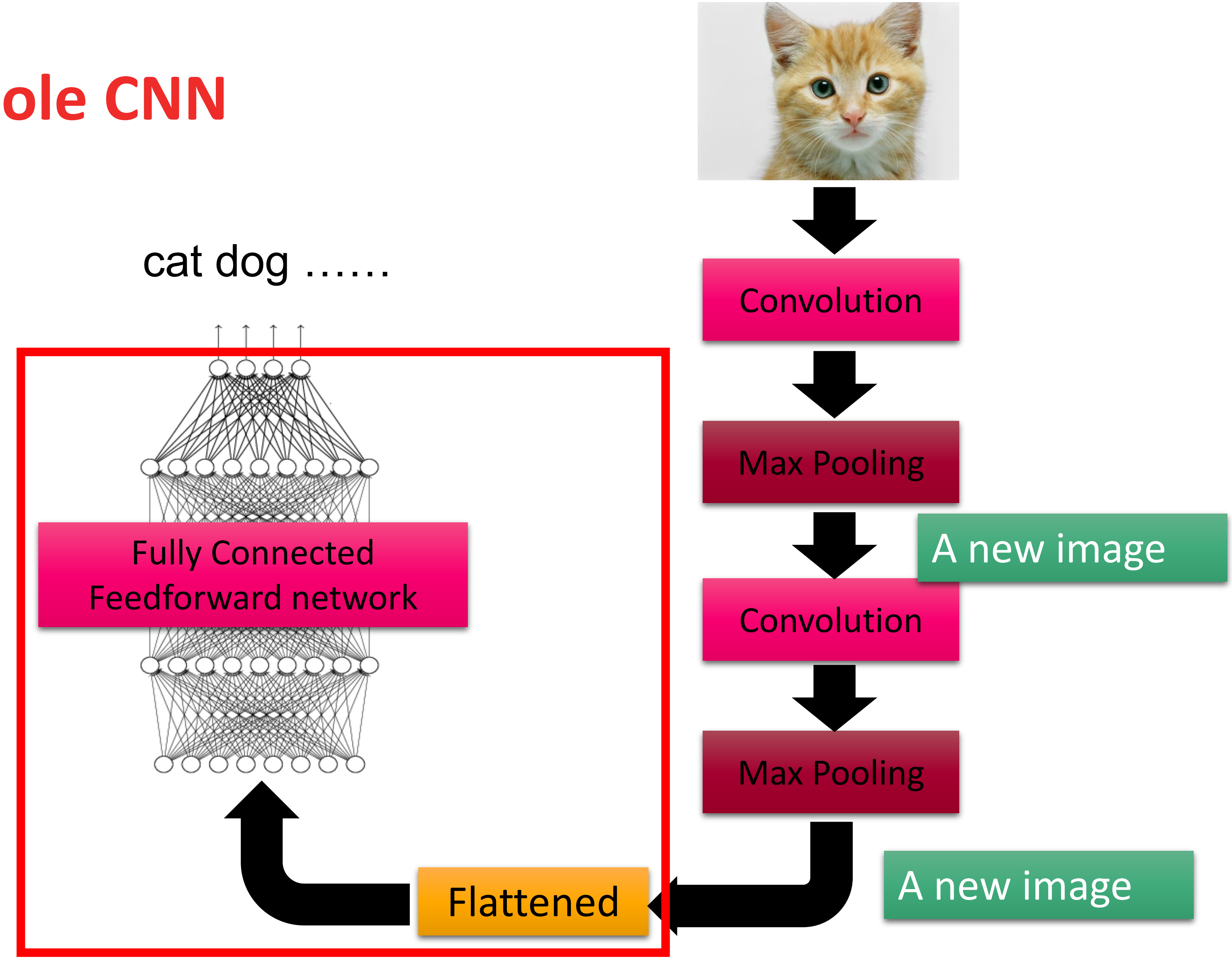




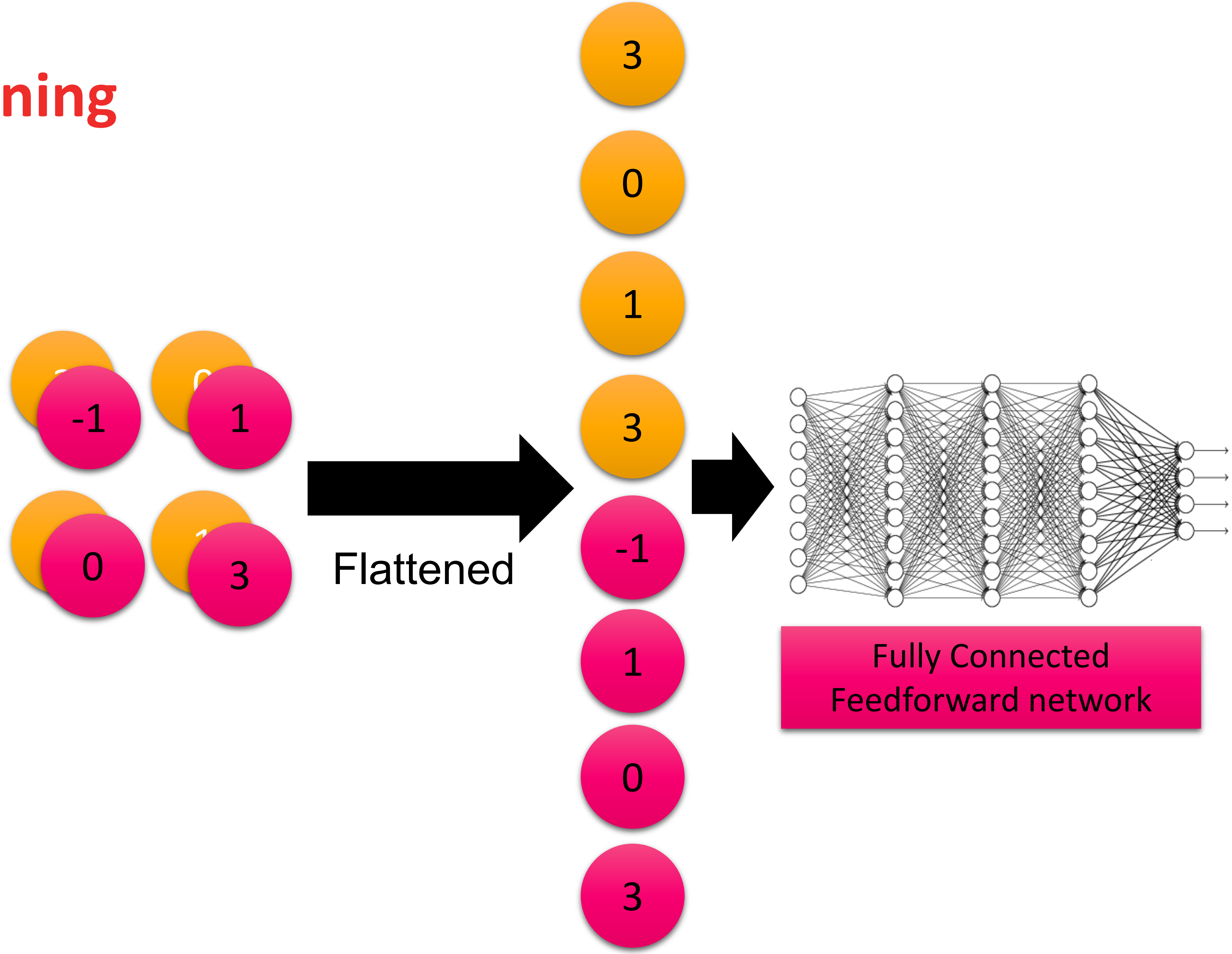
Smaller than the original image

The number of channels is the number of filters

The whole CNN



Flattening



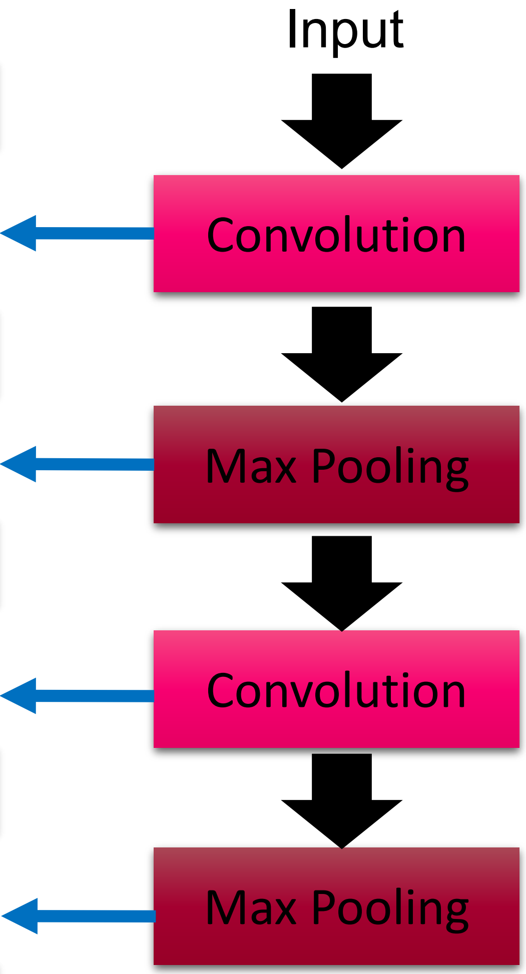
Flattened

Fully Connected Feedforward network

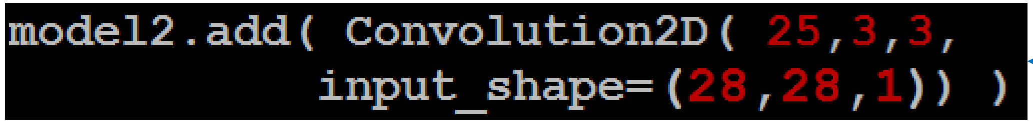
Full Convolution NN in Keras

CNN in Keras

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



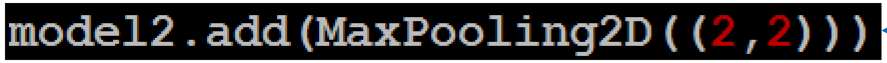
1 x 28 x 28



```
model2.add( Convolution2D( 25,3,3,
            input_shape=(28,28,1)) )
```

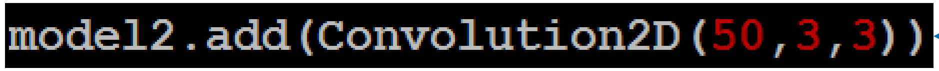
How many parameters for each filter?

25 x 26 x 26



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

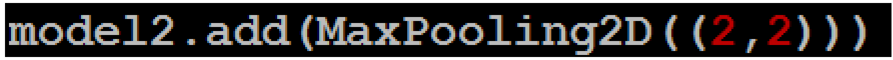
25 x 13 x 13



```
model2.add(Convolution2D(50,3,3))
```

How many parameters for each filter?

50 x 11 x 11

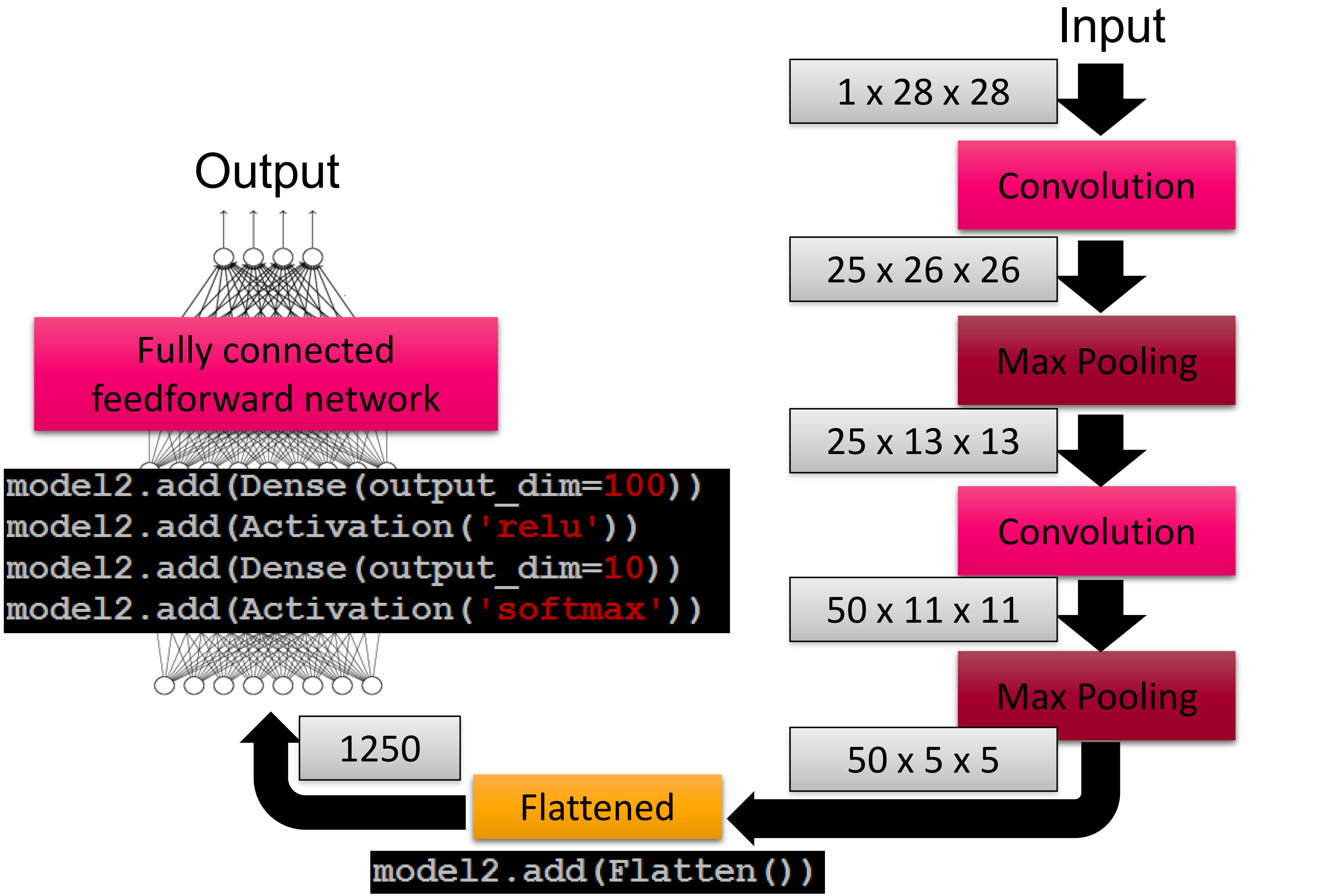


```
model2.add(MaxPooling2D((2,2)))
```

50 x 5 x 5

CNN in Keras

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

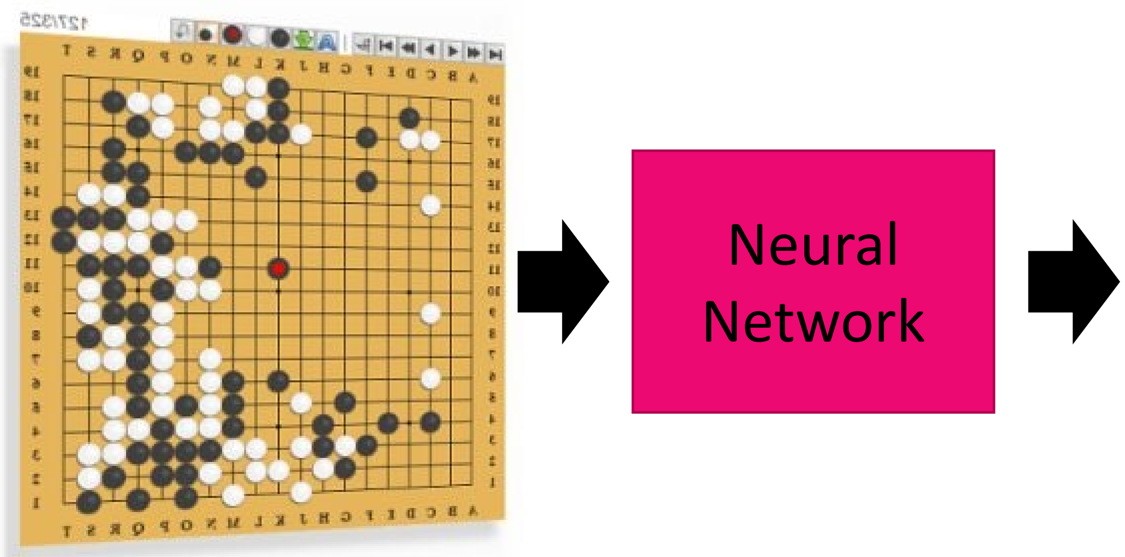


```
model2.add(Dense(output_dim=100))
model2.add(Activation('relu'))
model2.add(Dense(output_dim=10))
model2.add(Activation('softmax'))
```

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

Examples CNN

AlphaGo



Neural Network

Next move

(19 x 19 positions)

19 x 19 matrix

Black: 1

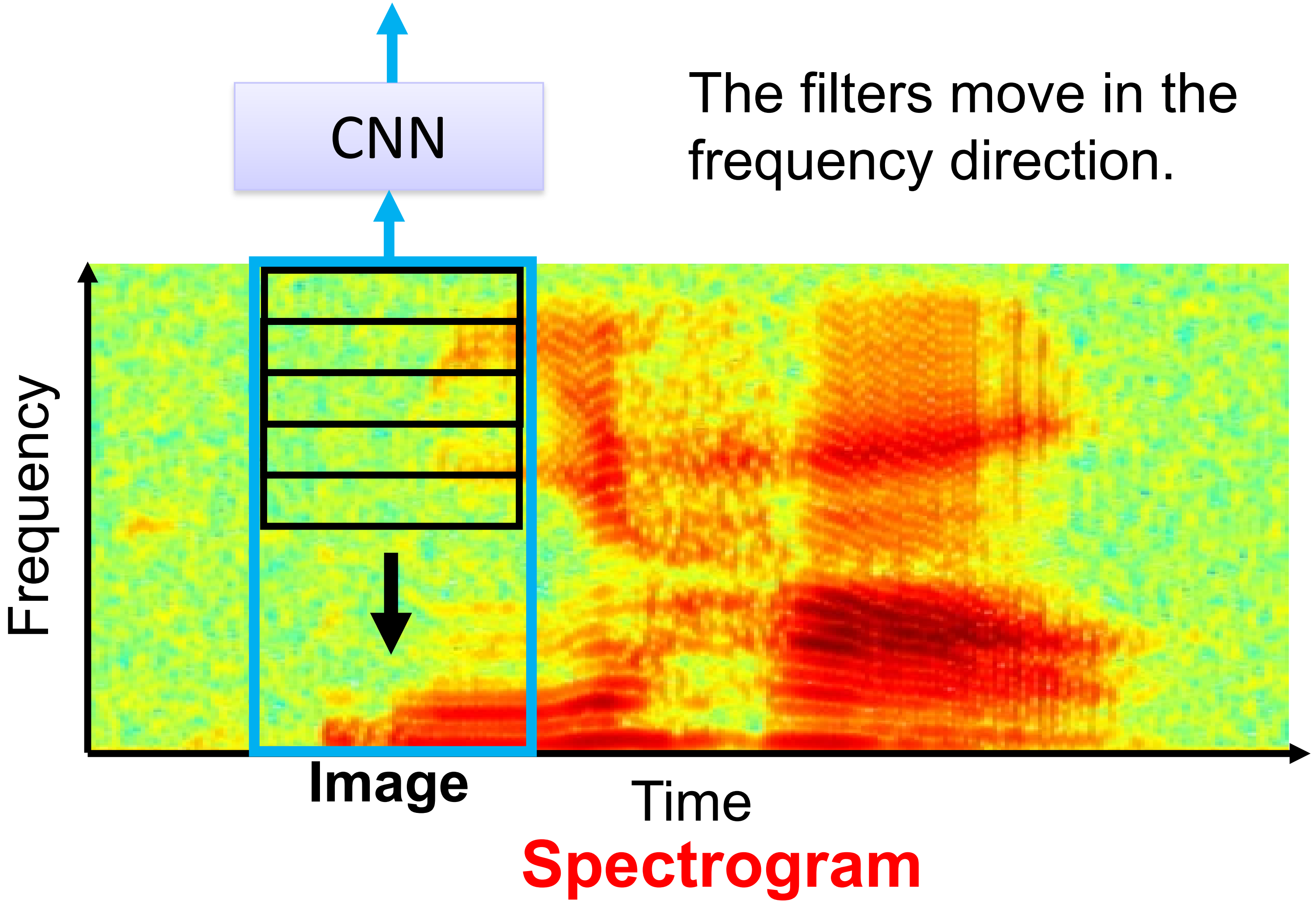
white: -1

none: 0

Fully-connected feedforward network can be used

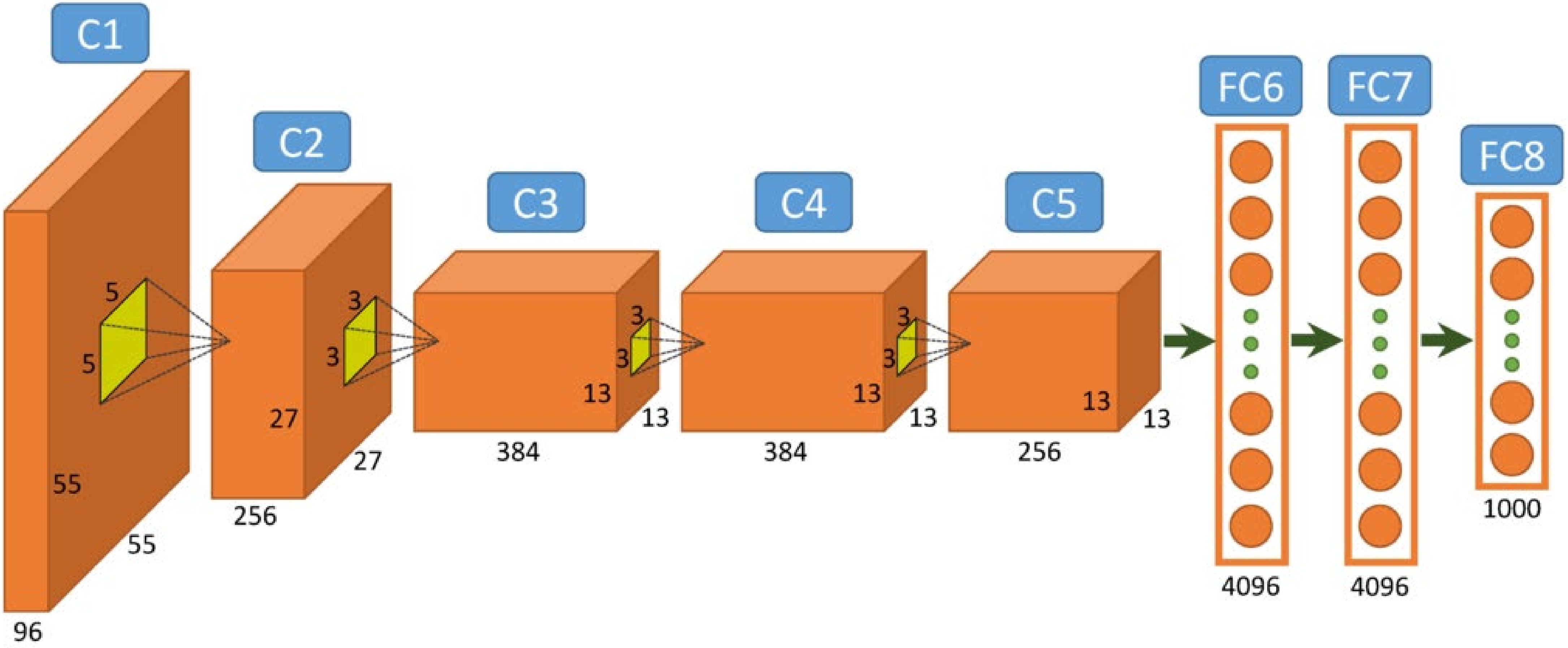
But CNN performs much better

CNN in speech recognition



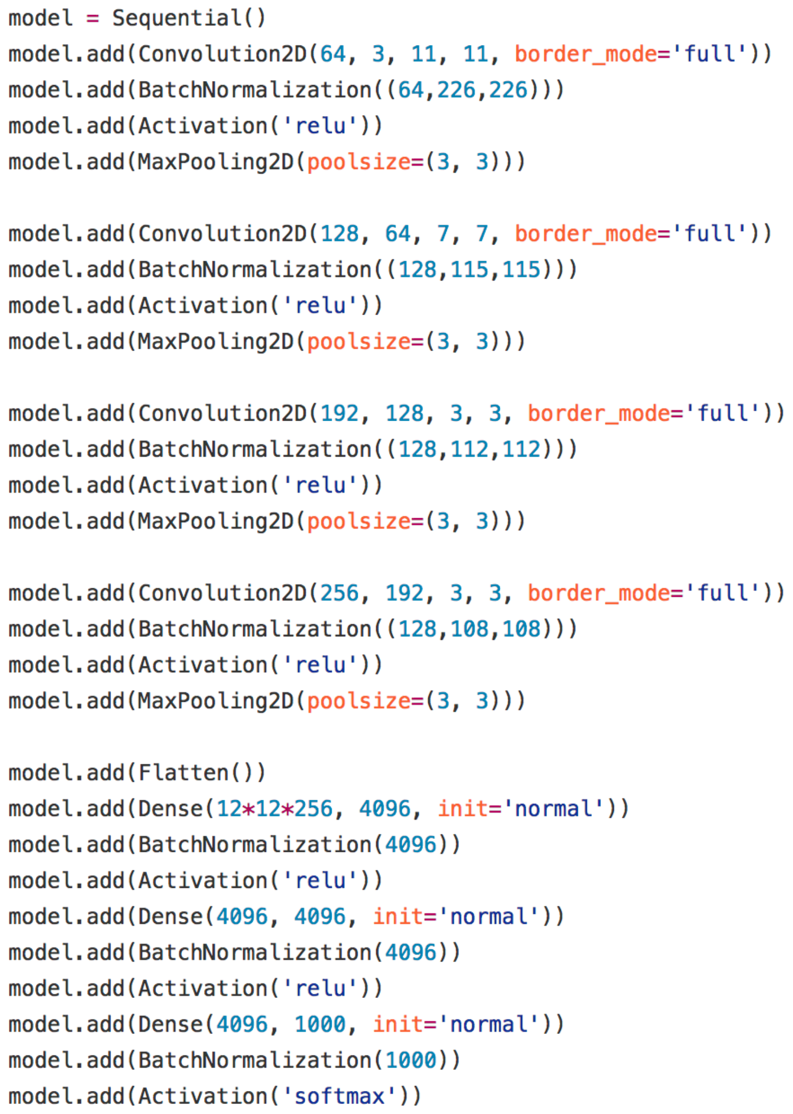
The filters move in the frequency direction.

Alexnet



https://www.saagie.com/fr/blog/object-detection-part1

Alexnet in Keras

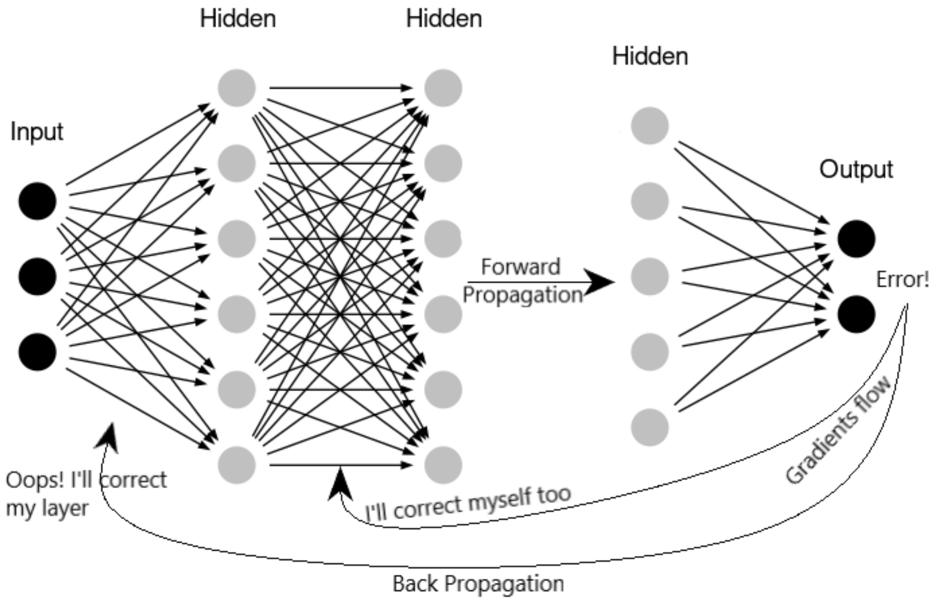


```
model = Sequential()
model.add(Convolution2D(64, 3, 11, 11, border_mode='full'))
model.add(BatchNormalization((64,226,226)))
model.add(Activation('relu'))
model.add(MaxPooling2D(poolsize=(3, 3)))

model.add(Convolution2D(128, 64, 7, 7, border_mode='full'))
model.add(BatchNormalization((128,115,115)))
model.add(Activation('relu'))
model.add(MaxPooling2D(poolsize=(3, 3)))

model.add(Convolution2D(192, 128, 3, 3, border_mode='full'))
model.add(BatchNormalization((128,112,112)))
model.add(Activation('relu'))
model.add(MaxPooling2D(poolsize=(3, 3)))

model.add(Convolution2D(256, 192, 3, 3, border_mode='full'))
model.add(BatchNormalization((128,108,108)))
model.add(Activation('relu'))
model.add(MaxPooling2D(poolsize=(3, 3)))

model.add(Flatten())
model.add(Dense(12*12*256, 4096, init='normal'))
model.add(BatchNormalization(4096))
model.add(Activation('relu'))
model.add(Dense(4096, 4096, init='normal'))
model.add(BatchNormalization(4096))
model.add(Activation('relu'))
model.add(Dense(4096, 1000, init='normal'))
model.add(BatchNormalization(1000))
model.add(Activation('softmax'))
```

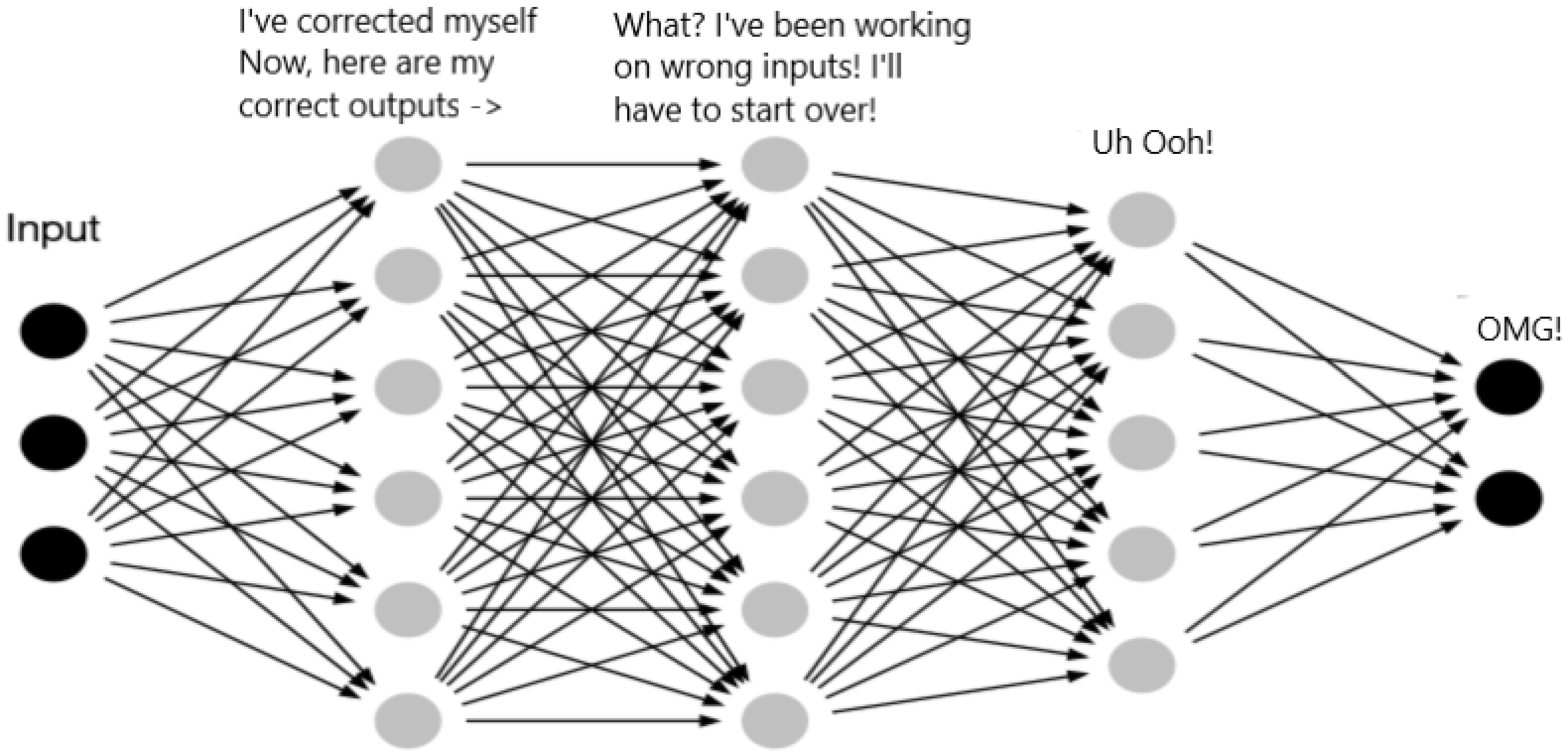
Batch Normalization

Batch Normalization?

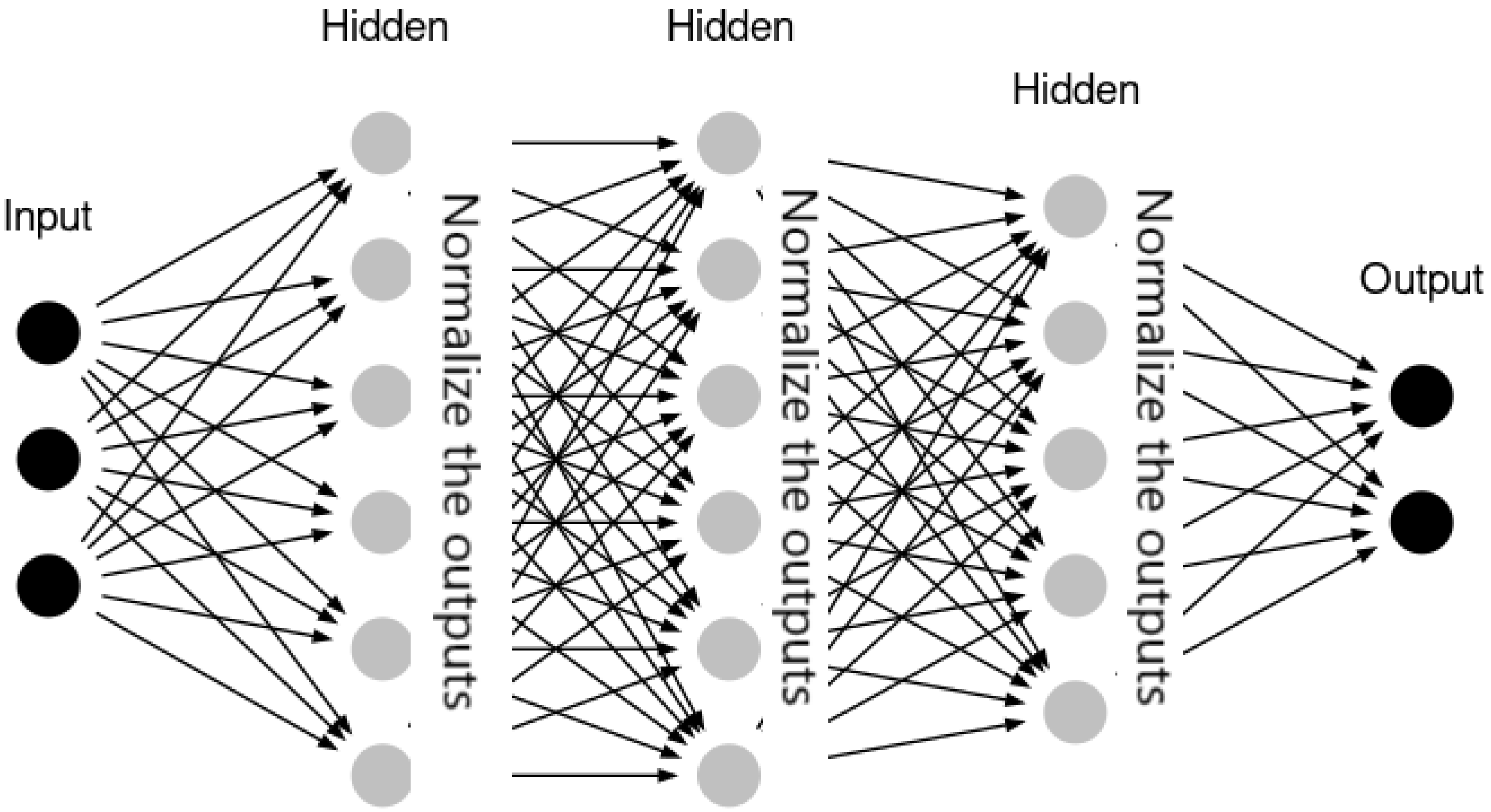
- Neural networks learn the problem using BackPropagation algorithm.
- BackPropagation involves computing gradients for each layer
- In deep networks this time explodes for training



Batch Normalization?


I've corrected myself Now, here are my correct outputs ->
What? I've been working on wrong inputs! I'll have to start over!
Uh Ooh!
Input
OMG!

Batch Normalization?



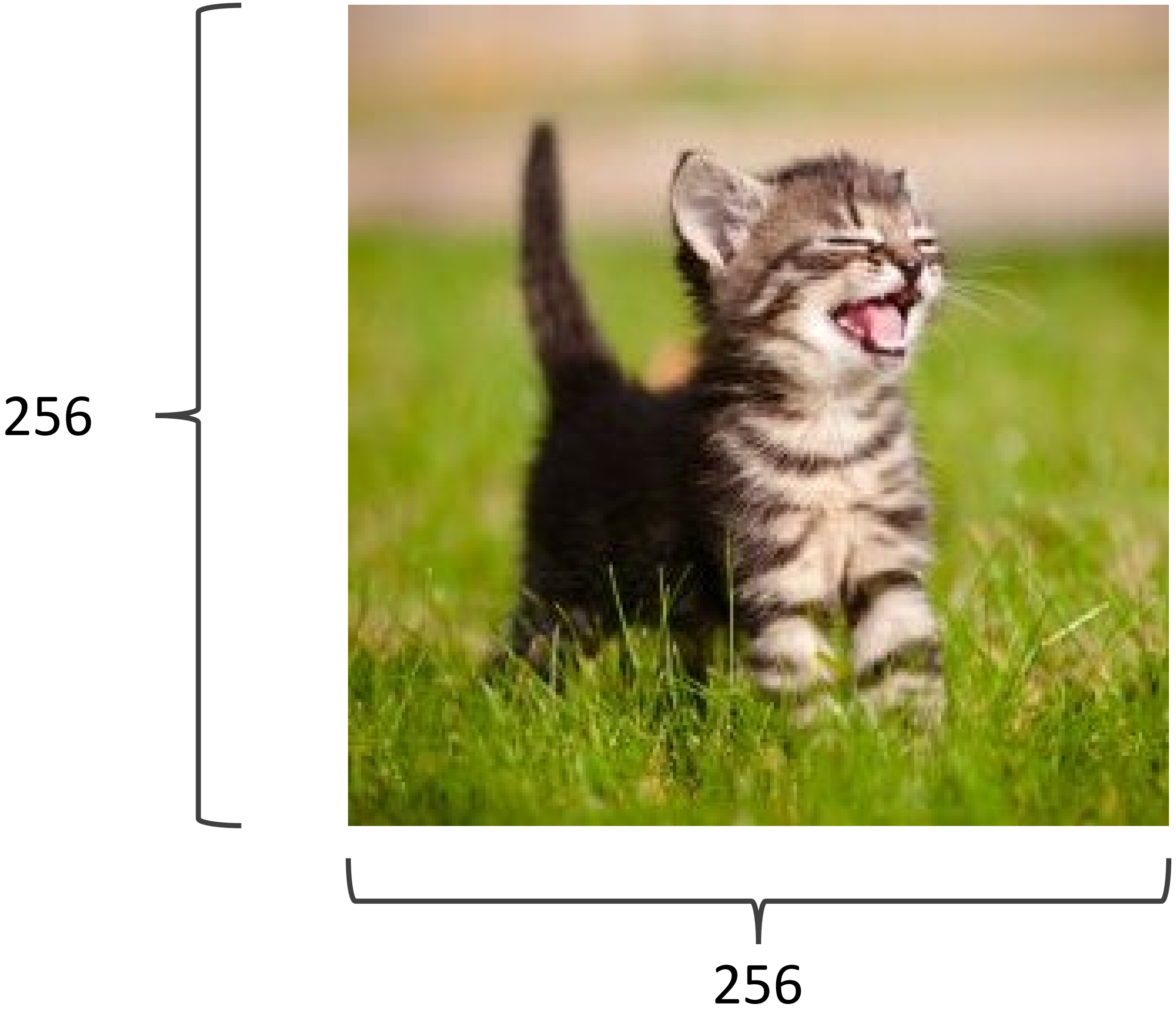
Batch Normalization?

- Normalization brings all the inputs centered around 0.
- This way, there is not much change in each layer input.
- So, layers in the network can learn from the back-propagation simultaneously, without waiting for the previous layer to learn.
- This speeds up the training of networks.

Pre-Processing

Preprocessing and Data Augmentation

Preprocessing and Data Augmentation

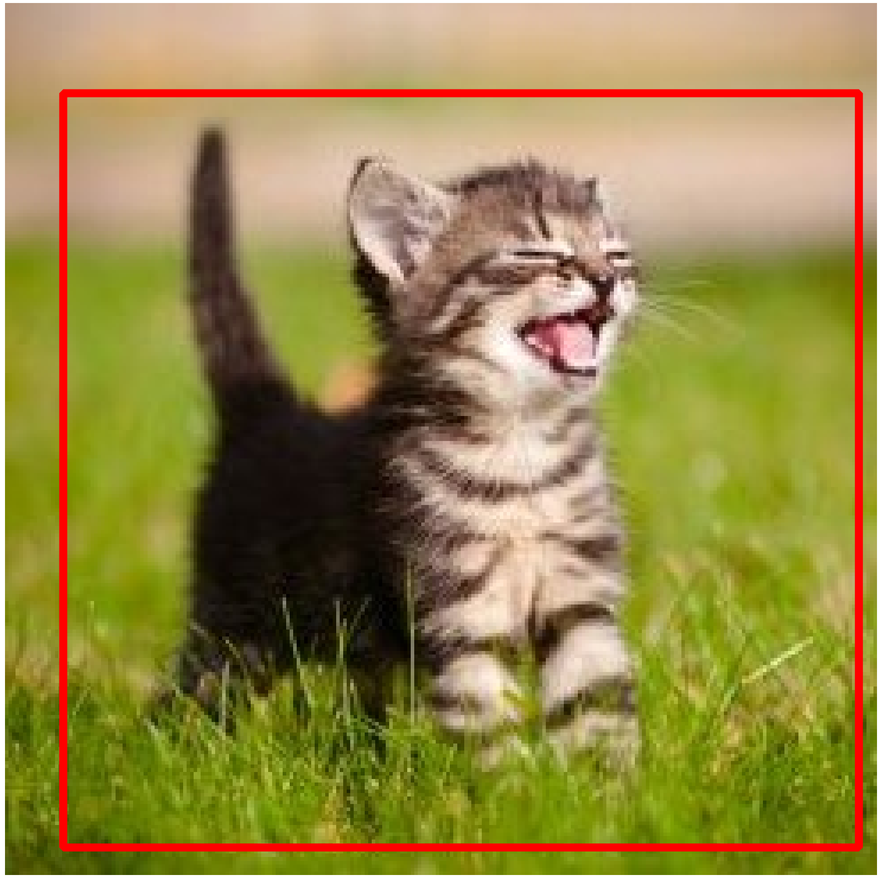


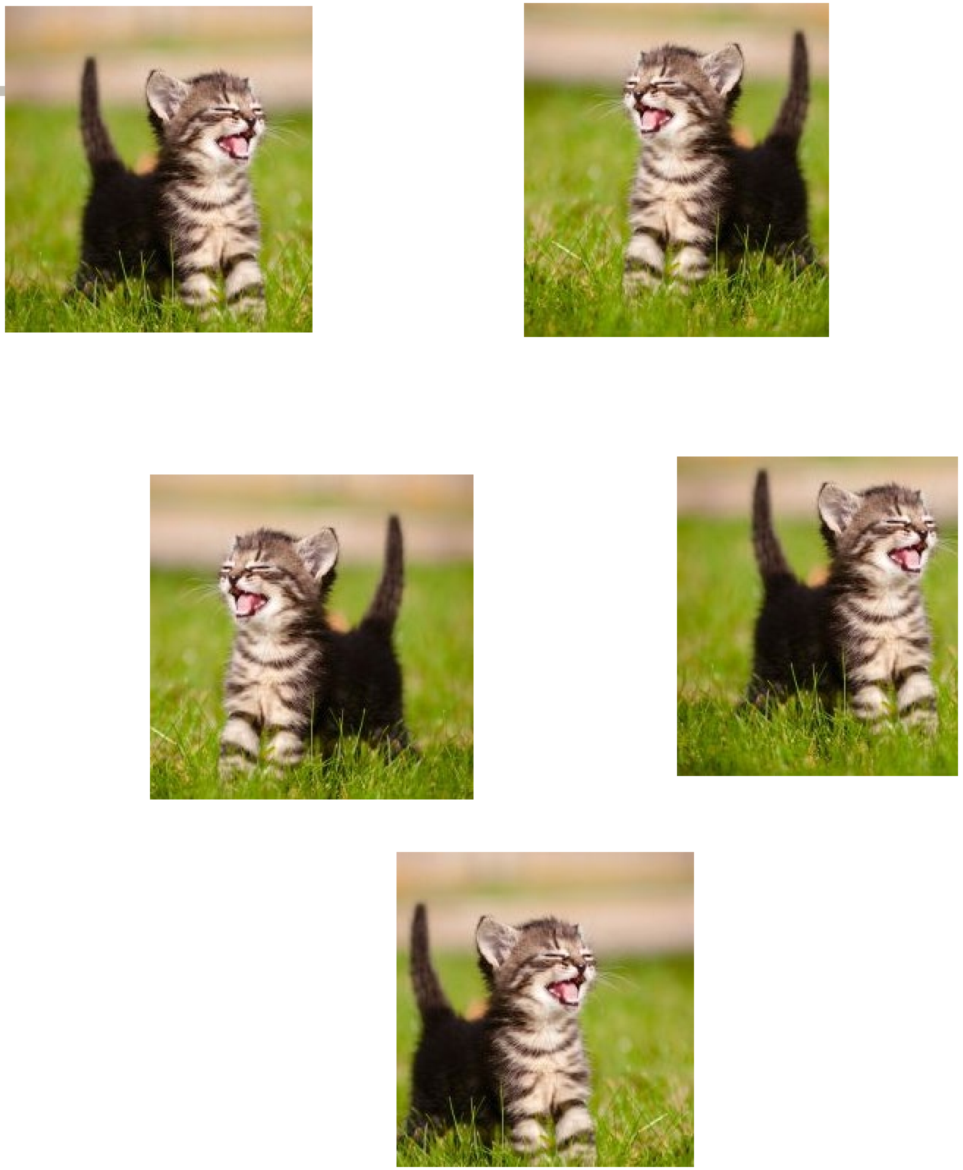
Preprocessing and Data Augmentation

224x224

Preprocessing and Data Augmentation

224x224



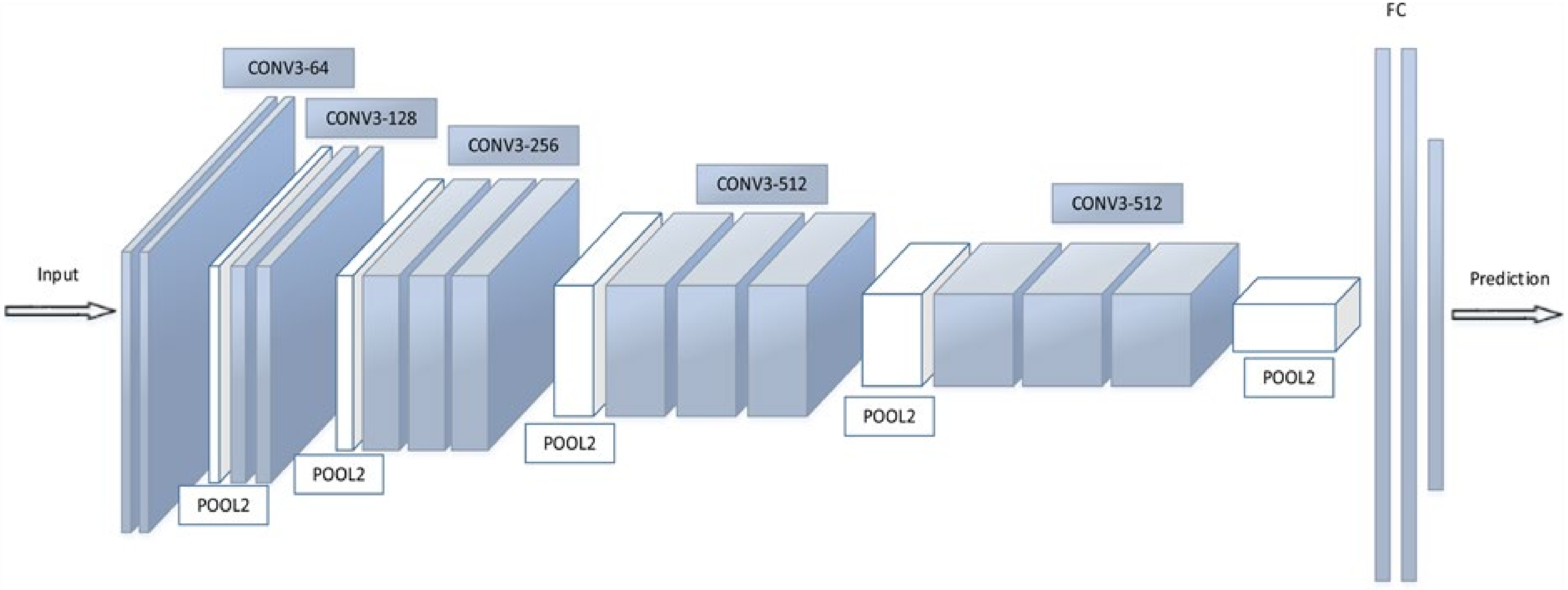


True label: Abyssinian cat

Other CNNs

VGG Network

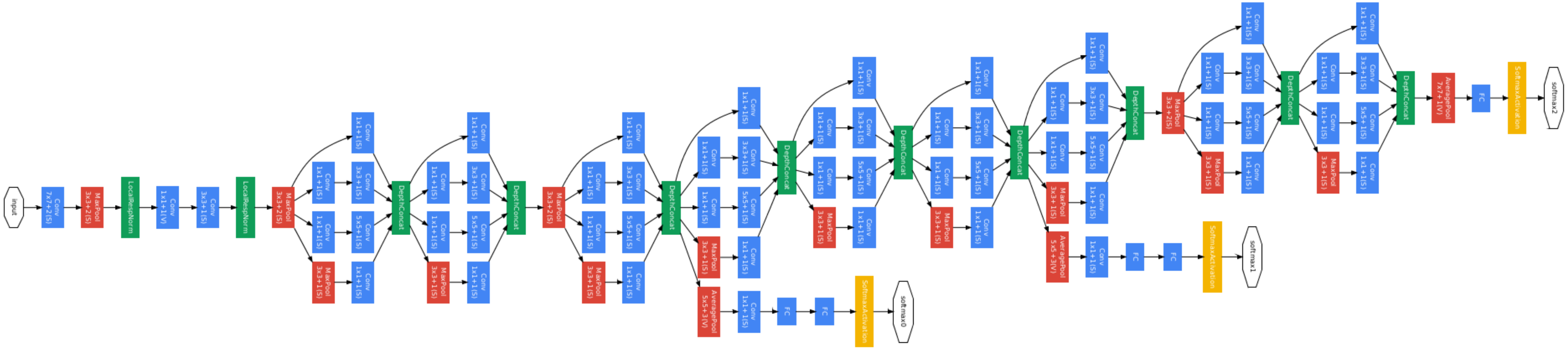
Top-5:



Keras: https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3

Simonyan and Zisserman, 2014.

GoogLeNet



Keras: https://gist.github.com/joelouismarino/a2ede9ab3928f999575423b9887abd14

Szegedy et al. 2014

ResNet

Sorry, does not fit in slide.

http://felixlaumon.github.io/assets/kaggle-right-whale/resnet.png

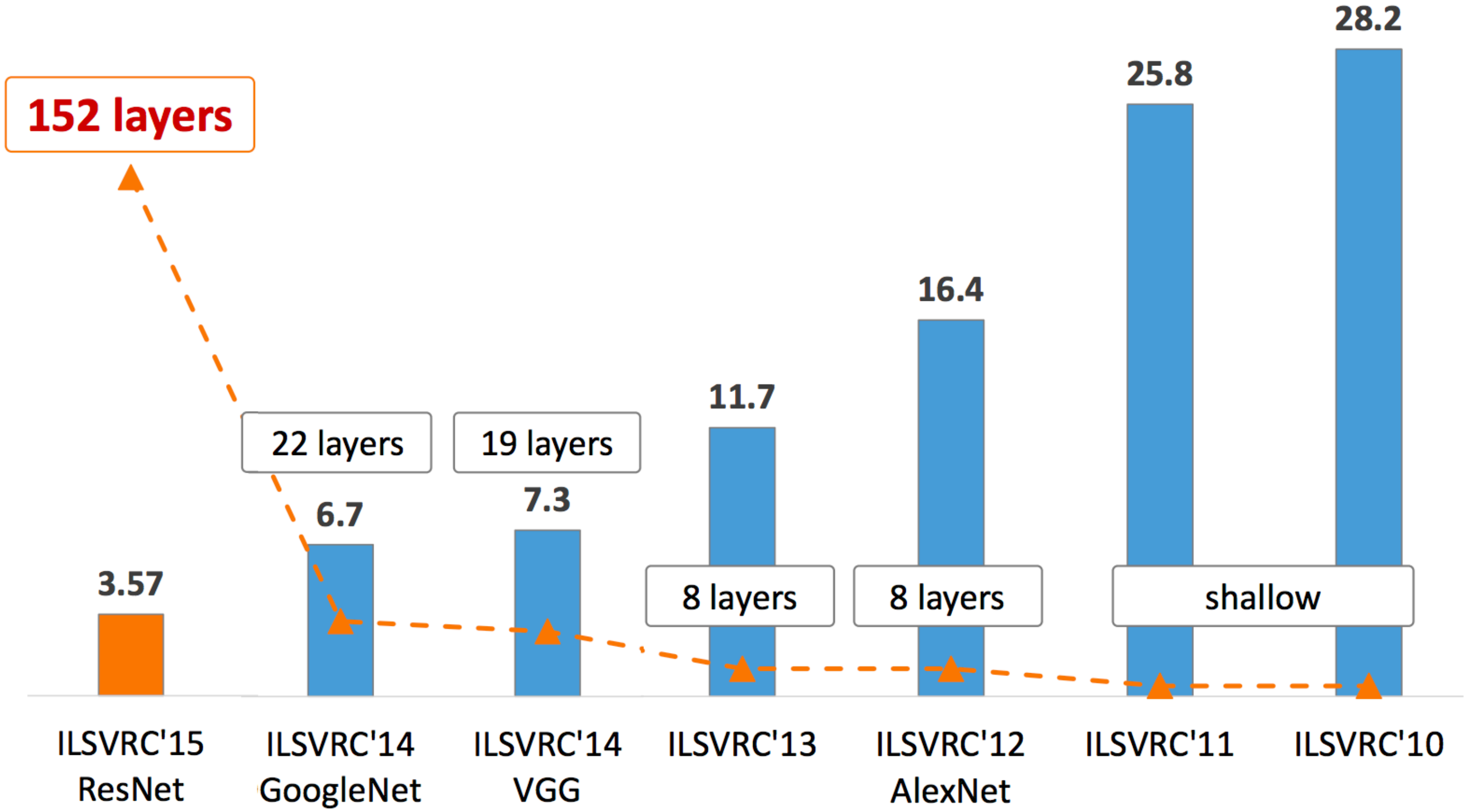
Keras: https://github.com/raghakot/keras-resnet/blob/master/resnet.py

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)



Onward to … Dangers.

Jonathan Hudson
jwhudson@ucalgary.ca
https://pages.cpsc.ucalgary.ca/~hudsonj/

UNIVERSITY OF CALGARY