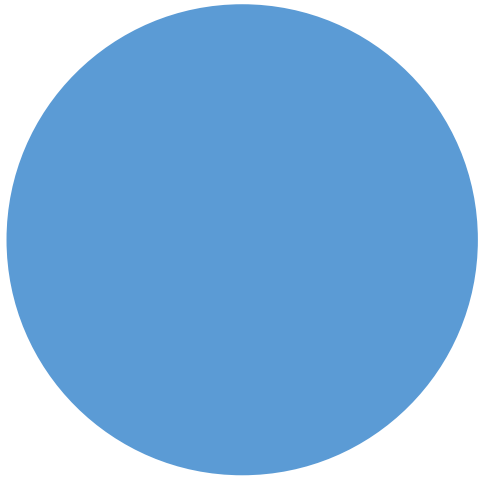
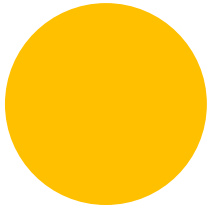
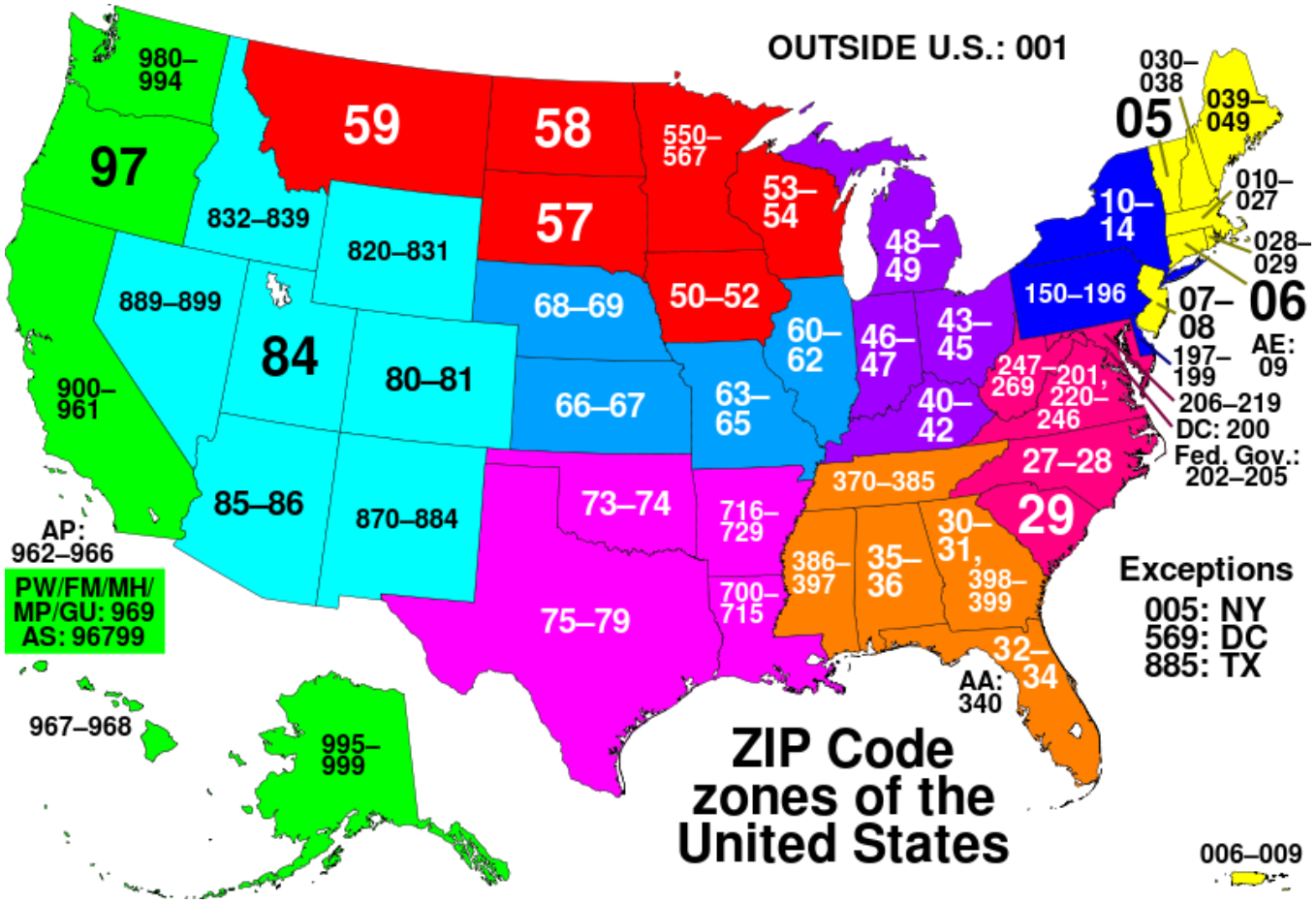
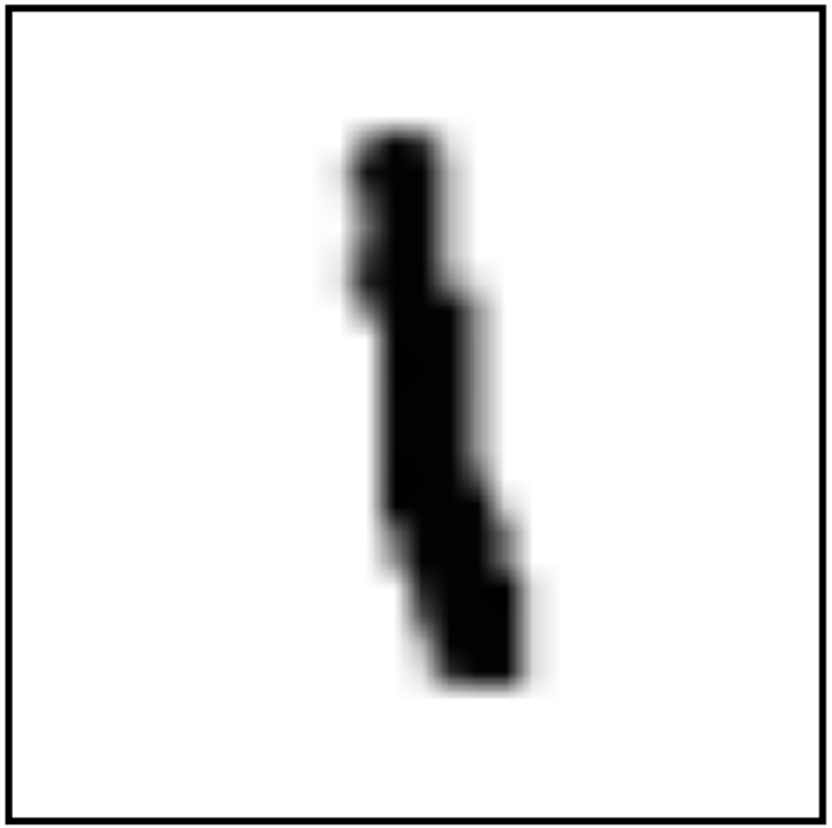



Neural Network II

MNIST

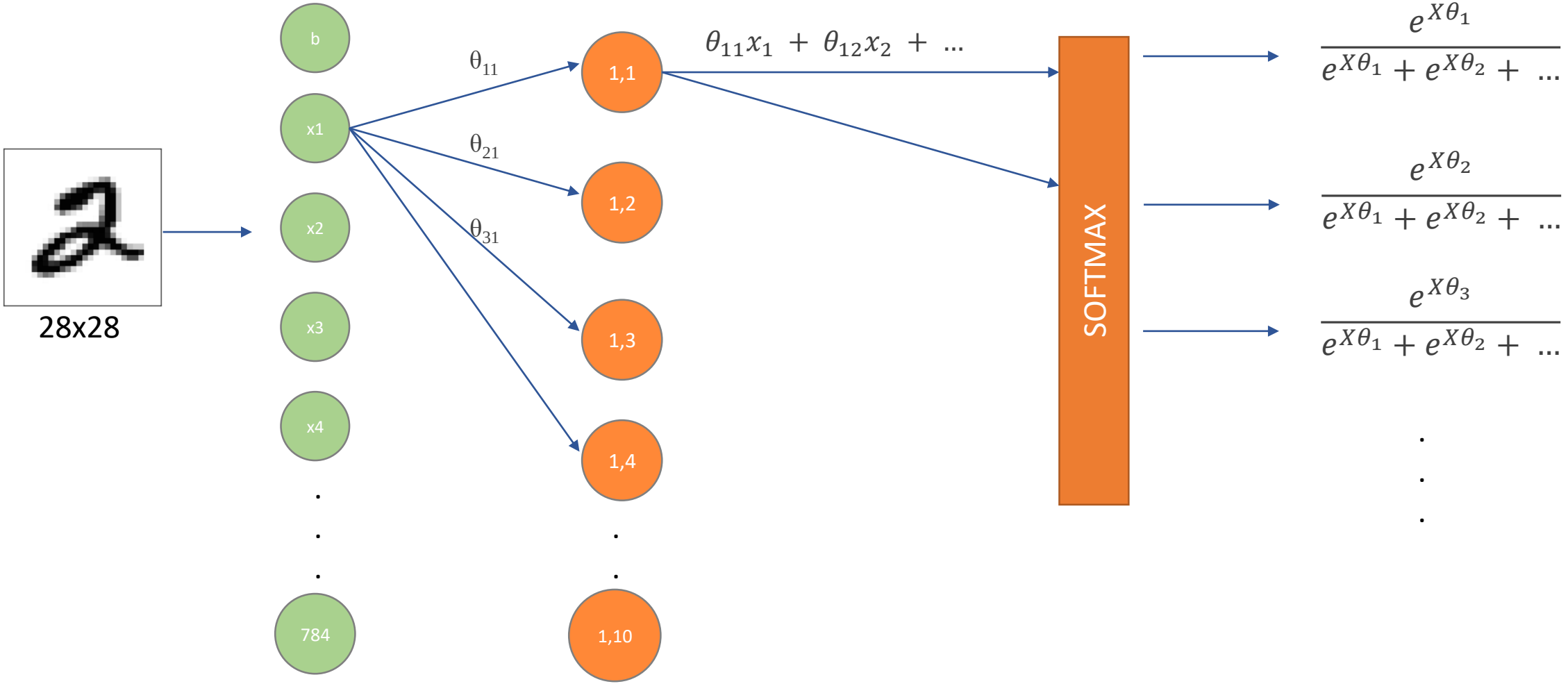

OUTSIDE U.S.: 001
980–994
97
832–839
820–831
59
58
57
550–567
53–54
48–49
030–038
05
039–049
010–027
10–14
028–029
06
07–08
AE: 09
150–196
197–199
206–219
DC: 200
Fed. Gov.: 202–205
889–899
84
80–81
68–69
50–52
60–62
46–47
43–45
247–269
201, 220–246
900–961
66–67
63–65
40–42
27–28
370–385
85–86
870–884
73–74
716–729
30–31, 398–399
29
AP: 962–966
PW/FM/MH/ MP/GU: 969 AS: 96799
386–397
35–36
700–715
75–79
32–34
AA: 340
Exceptions
005: NY
569: DC
885: TX
967–968
995–999
ZIP Code zones of the United States
006–009

The MNIST dataset



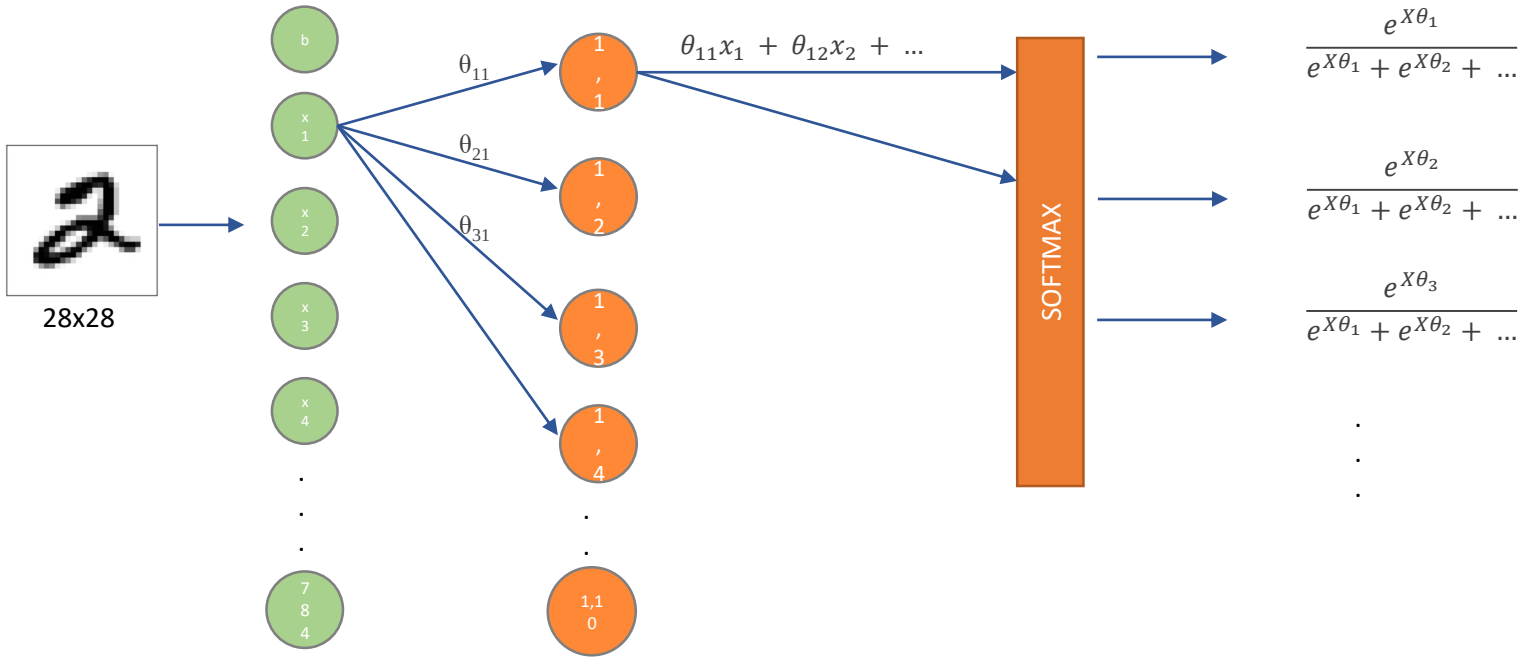
$\approx$

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$


28x28
b
x1
x2
x3
x4
784
θ11
θ21
θ31
1,1
1,2
1,3
1,4
1,10
θ11x1 + θ12x2 + …
SOFTMAX
e^Xθ1 / (e^Xθ1 + e^Xθ2 + …)
e^Xθ2 / (e^Xθ1 + e^Xθ2 + …)
e^Xθ3 / (e^Xθ1 + e^Xθ2 + …)

Keras code

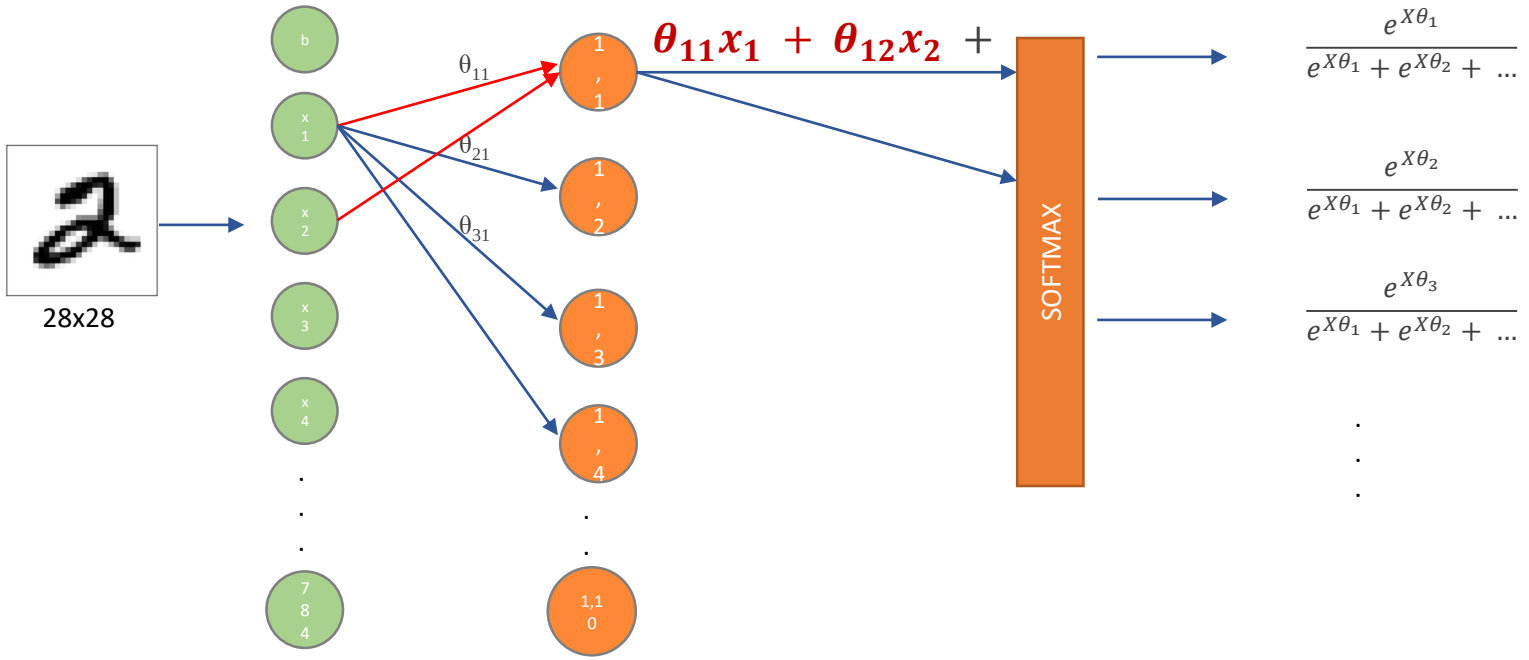
```
def softmax_model():
    model = Sequential()
    model.add(Dense(10, input_shape=(784,)))
    model.add(Activation('softmax'))
    return model


model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])

model.fit(X_train, Y_train,
              batch_size=128,
              epochs=10, verbose=1,
              validation_data=(X_test, Y_test))
```

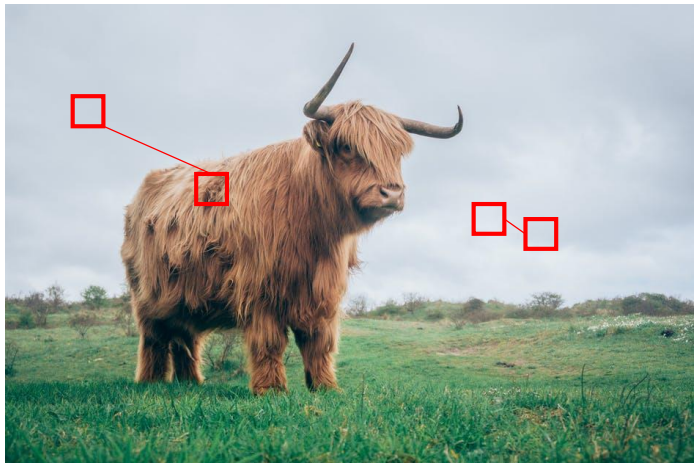


28x28x10 = ~7.6K

256x256x10 = ~650K

256x256x10x10x10 = ~65M


28x28
b
x 1
x 2
x 3
x 4
7 8 4
θ11
θ21
θ31
1 , 1
1 , 2
1 , 3
1 , 4
1,1 0
θ11x1 + θ12x2 +
SOFTMAX
e^Xθ1 / e^Xθ1 + e^Xθ2 + ...
e^Xθ2 / e^Xθ1 + e^Xθ2 + ...
e^Xθ3 / e^Xθ1 + e^Xθ2 + ...



Conv Nets

122	134	124	25	45
153	210	112	47	96
66	48	68	45	45
11	123	45	78	128

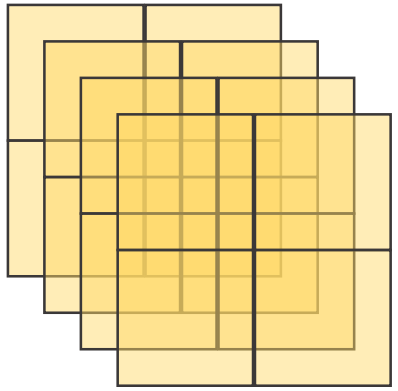
1	-1
1	0

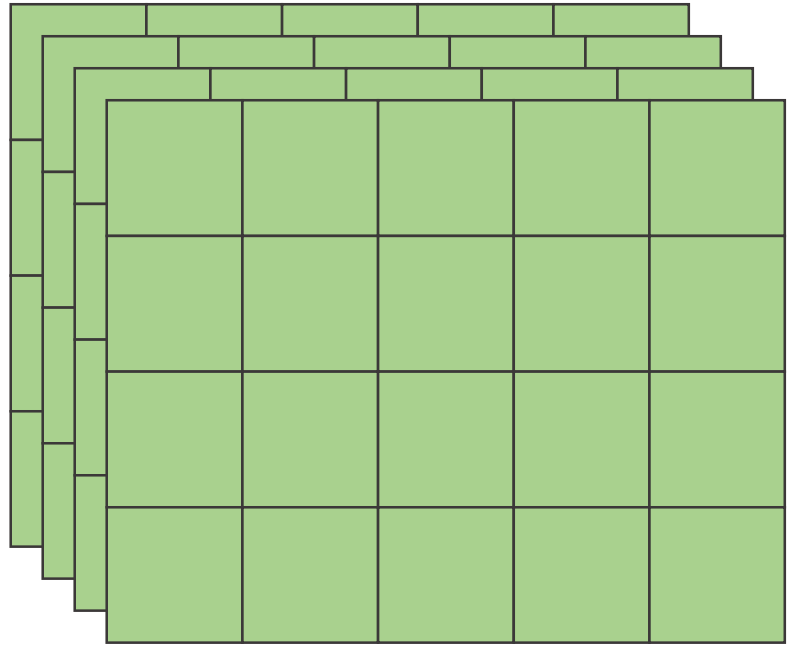
141	220	…	…	…
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

https://www.youtube.com/watch?v=SQ67NBCLV98

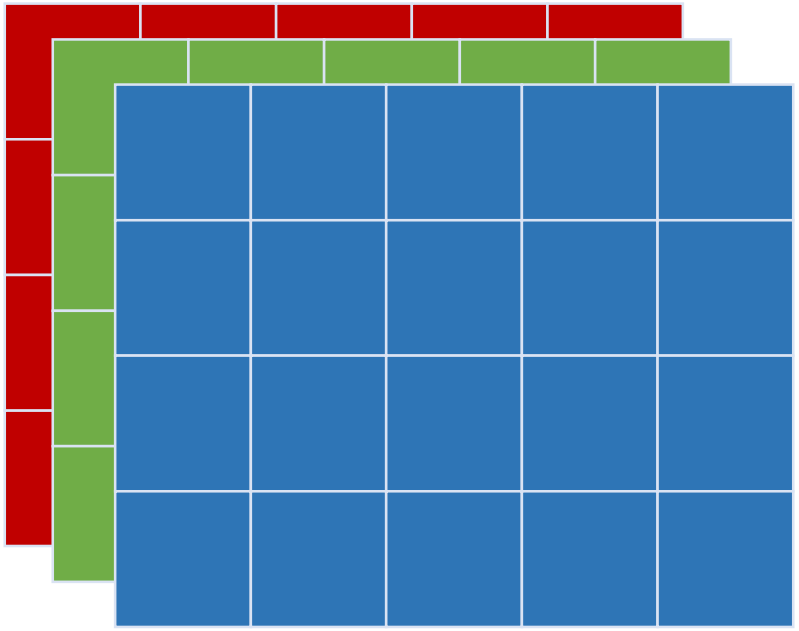
Conv Nets

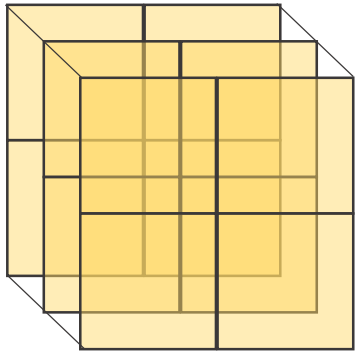
122	134	124	25	45
153	210	112	47	96
66	48	68	45	45
11	123	45	78	128

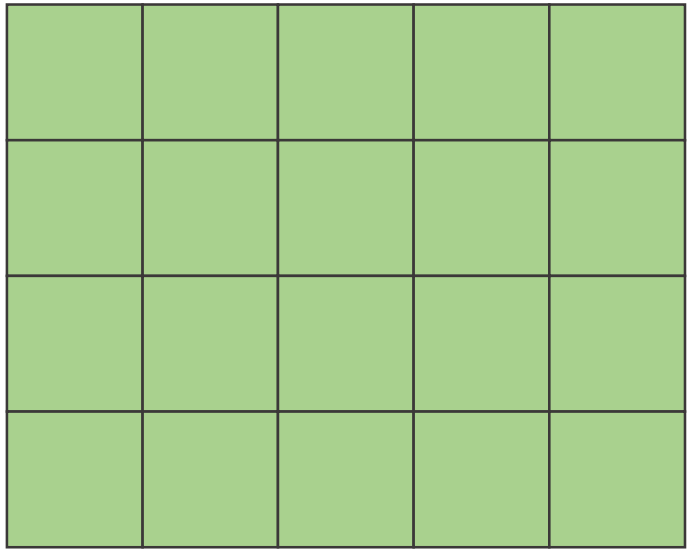




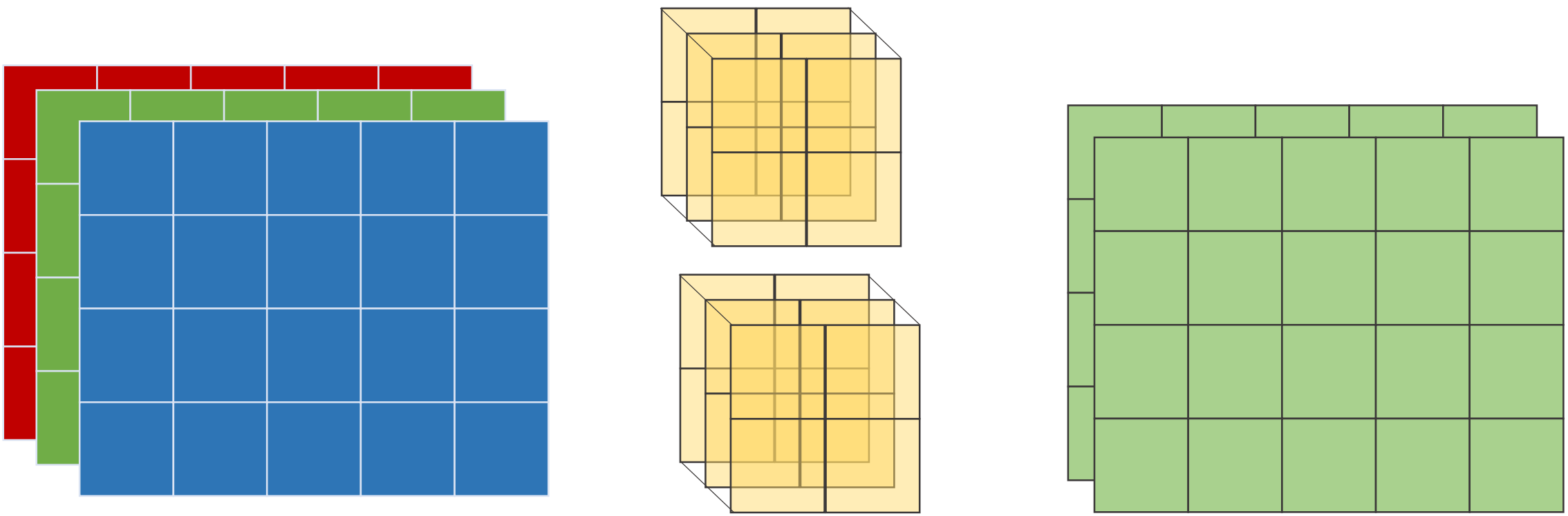
Conv Nets



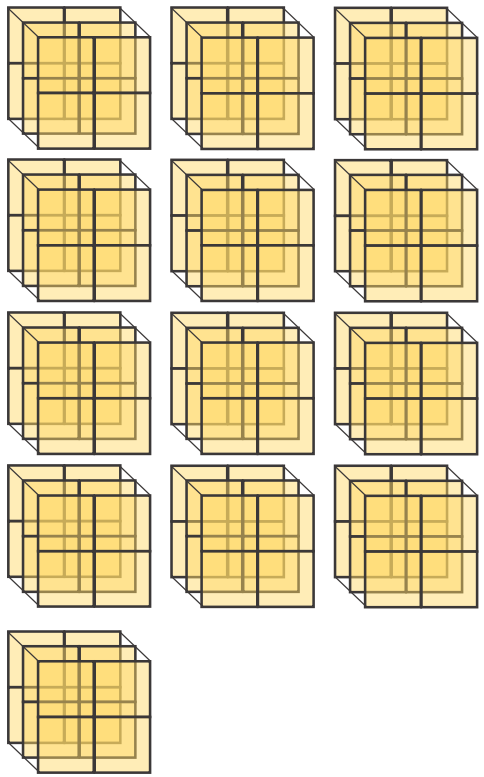


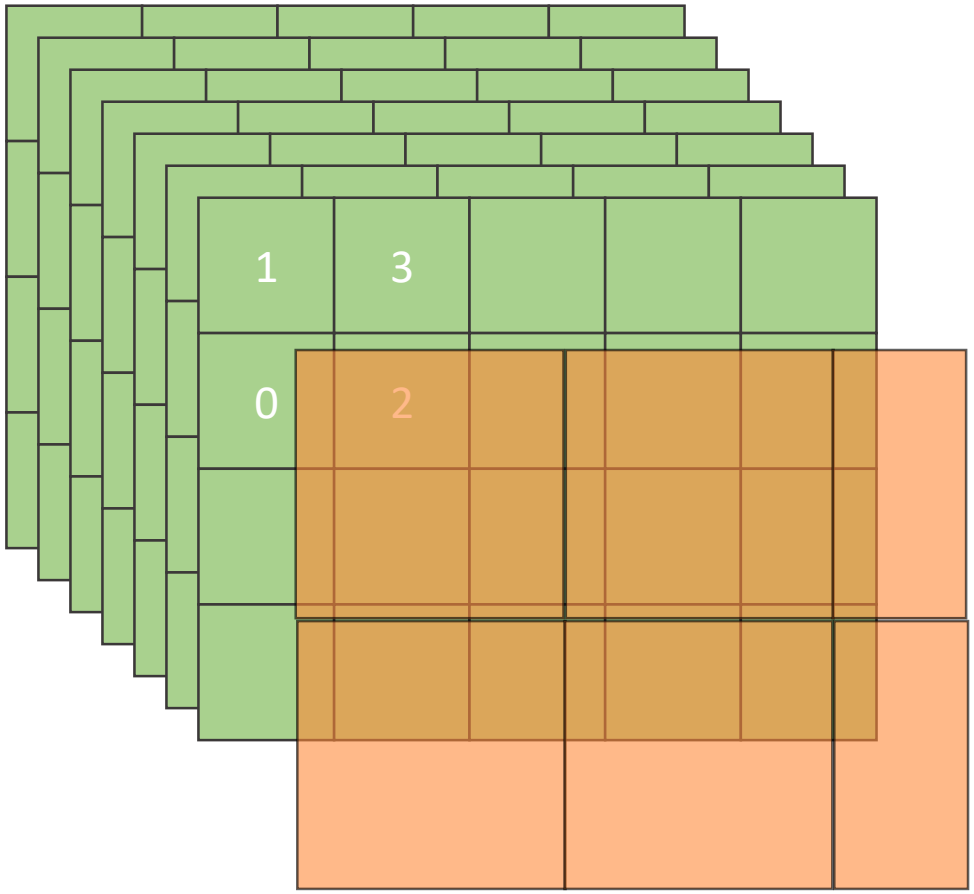


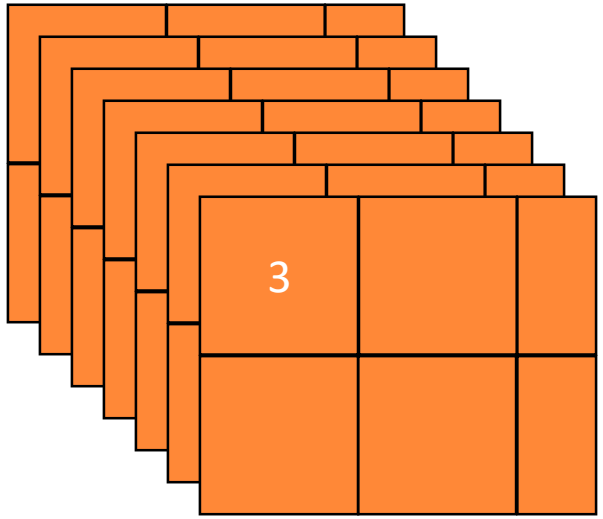
Conv Nets



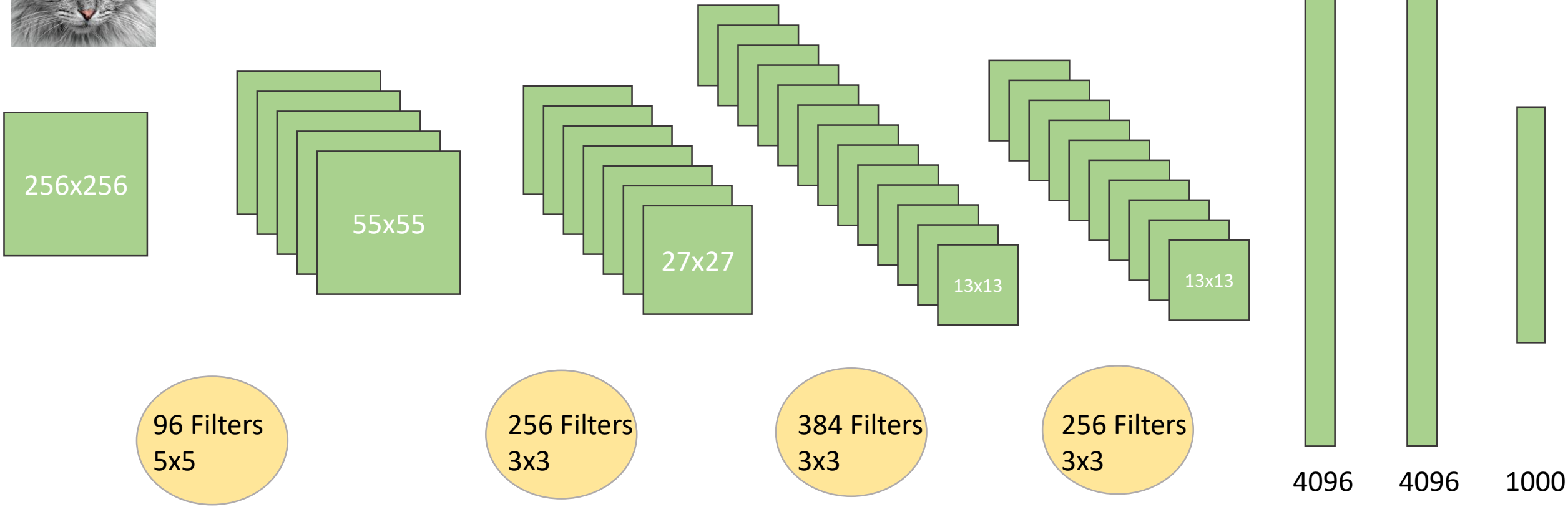
Pooling Layers



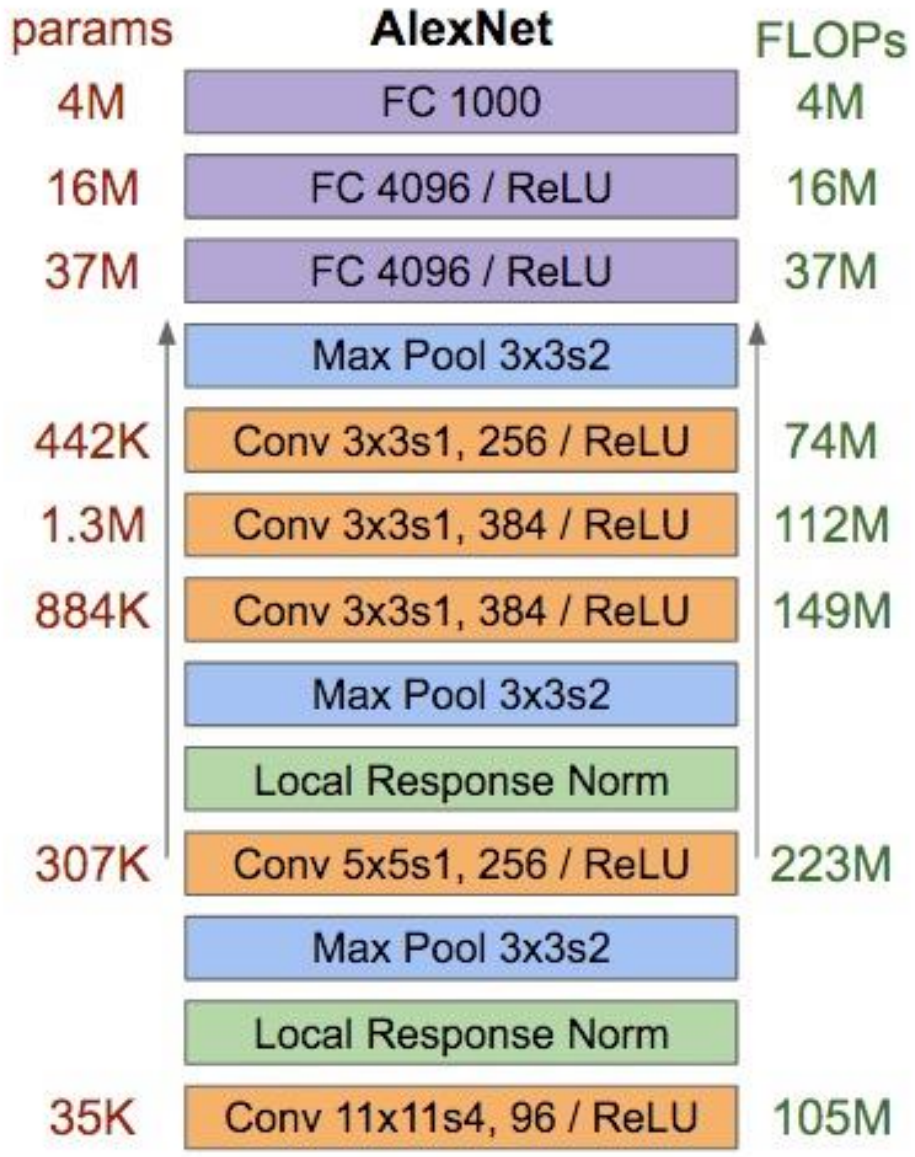




AlexNet



ImageNet Classification with Deep Convolutional Neural Networks
Alex Krizhevsky and Sutskever, Ilya and Hinton, Geoffrey E, 2012



AlexNet

- AlexNet has 60 million parameters and 500,000 neurons