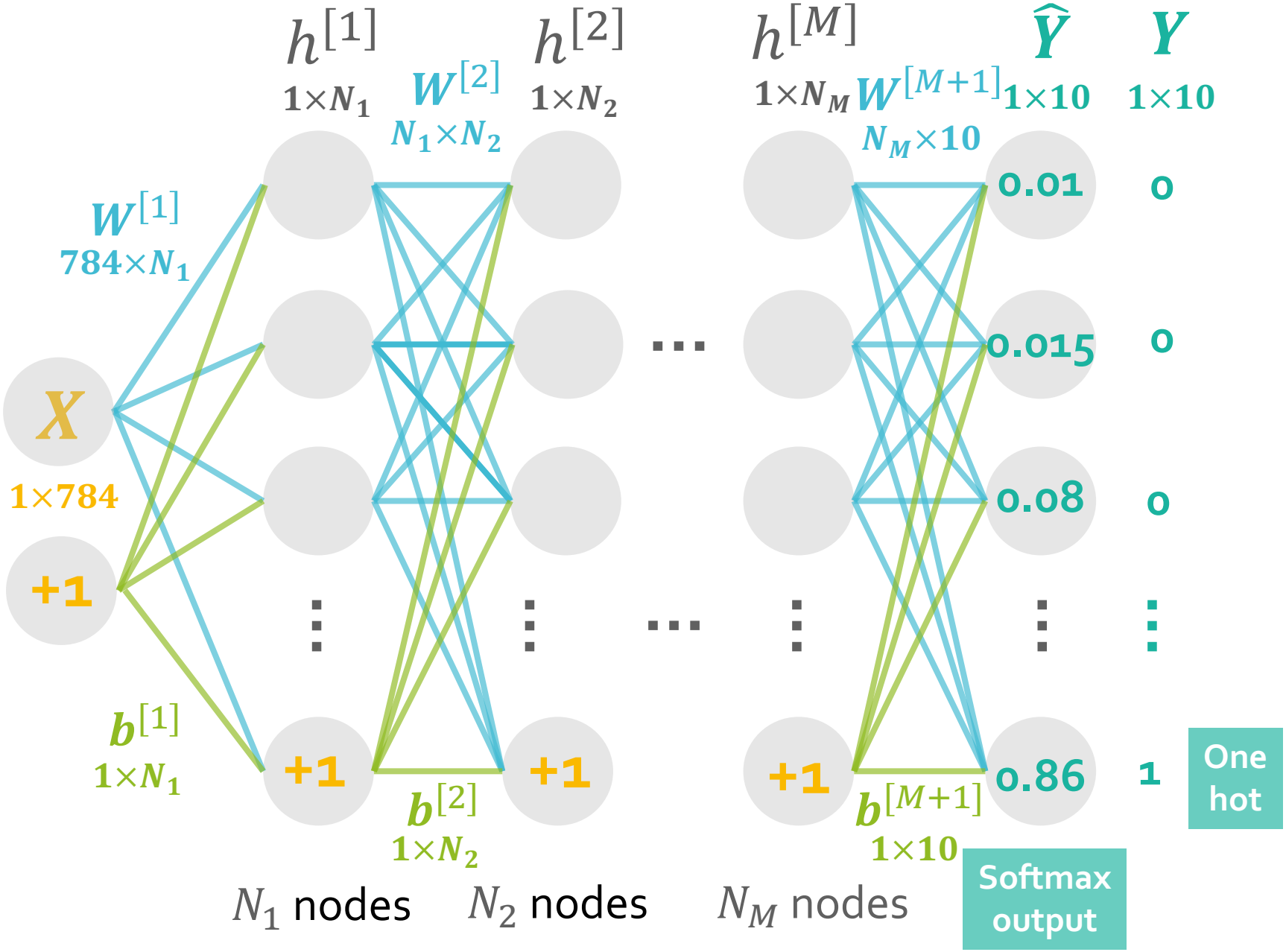
Basic RL.7

Judy Tutorial

MNIST dataset

[LeCun 1998]

Multiple Labels
$y \in \{0,1,2,\ldots,9\}$

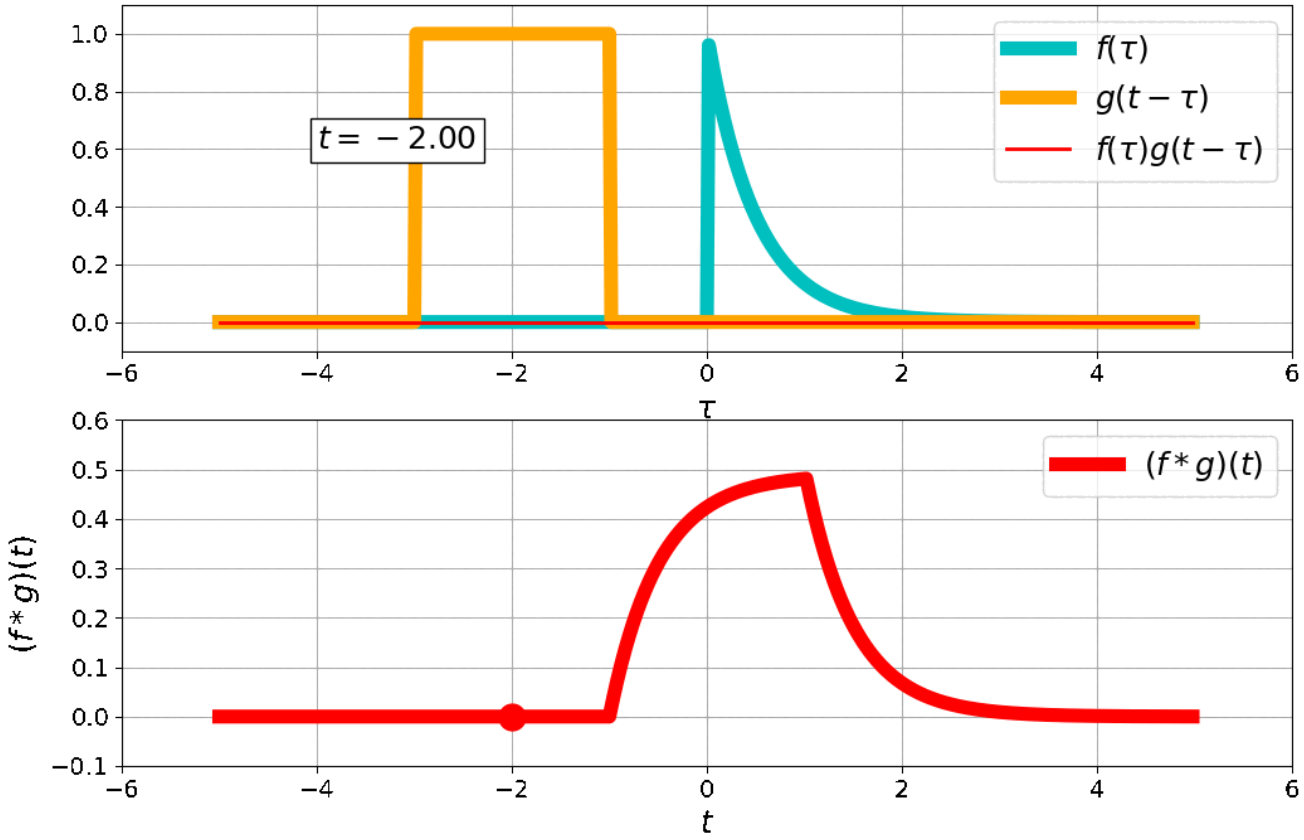

$h^{[1]}$
$1\times N_1$
$W^{[2]}$
$N_1\times N_2$
$h^{[2]}$
$1\times N_2$
$h^{[M]}$
$1\times N_M$
$W^{[M+1]}$
$N_M\times 10$
$\hat{Y}$
1×10
Y
1×10
$W^{[1]}$
$784\times N_1$
X
1×784
+1
$b^{[1]}$
$1\times N_1$
+1
+1
$b^{[2]}$
$1\times N_2$
+1
$b^{[M+1]}$
1×10
0.01
0.015
0.08
0.86
0
0
0
1
One hot
N_1 nodes
N_2 nodes
N_M nodes
Softmax output

Convolutional Neural Networks

[LeCun 1980s]

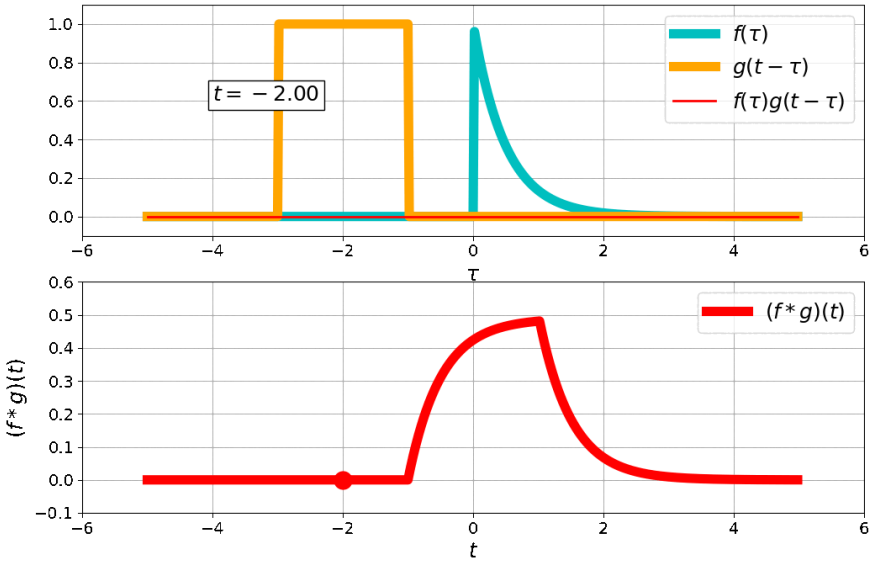
Convolution of two functions

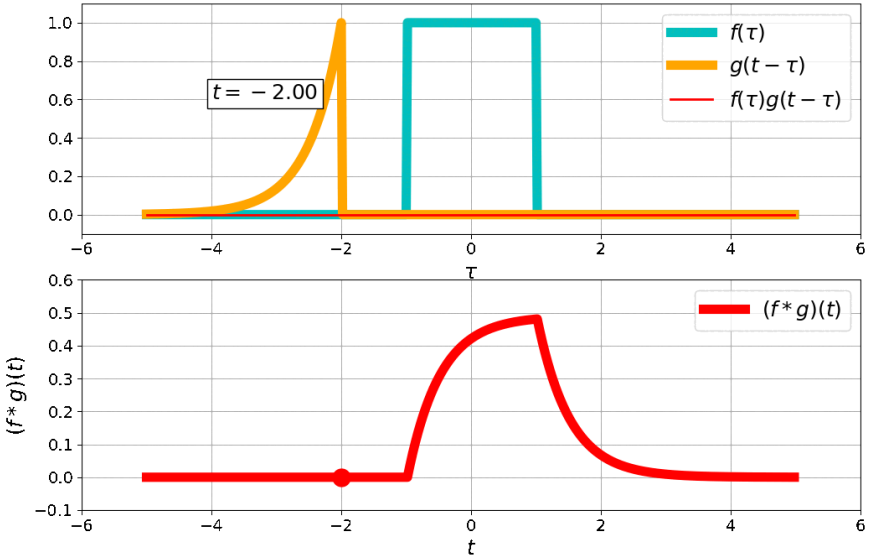
$$(\boldsymbol{f} * \boldsymbol{g})(t) \triangleq \int_{-\infty}^{\infty} \boldsymbol{f}(\tau)\boldsymbol{g}(t-\tau)d\tau$$


$t = -2.00$
$f(\tau)$
$g(t-\tau)$
$f(\tau)g(t-\tau)$
τ
$(f*g)(t)$
t

Convolution is Commutative

$$(f * g)(t) = (g * f)(t)$$





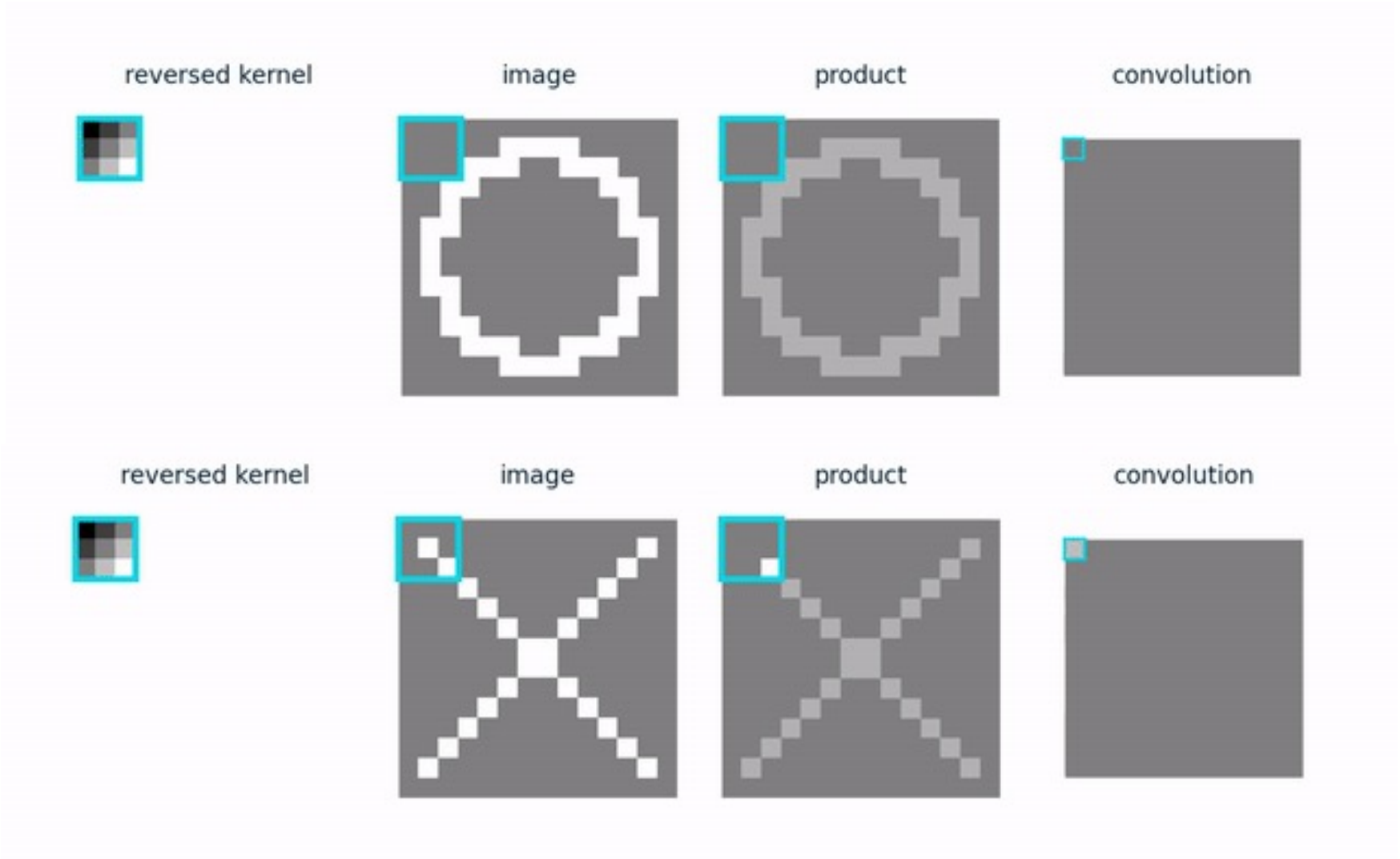
In Image Processing [2D]

Feature Map

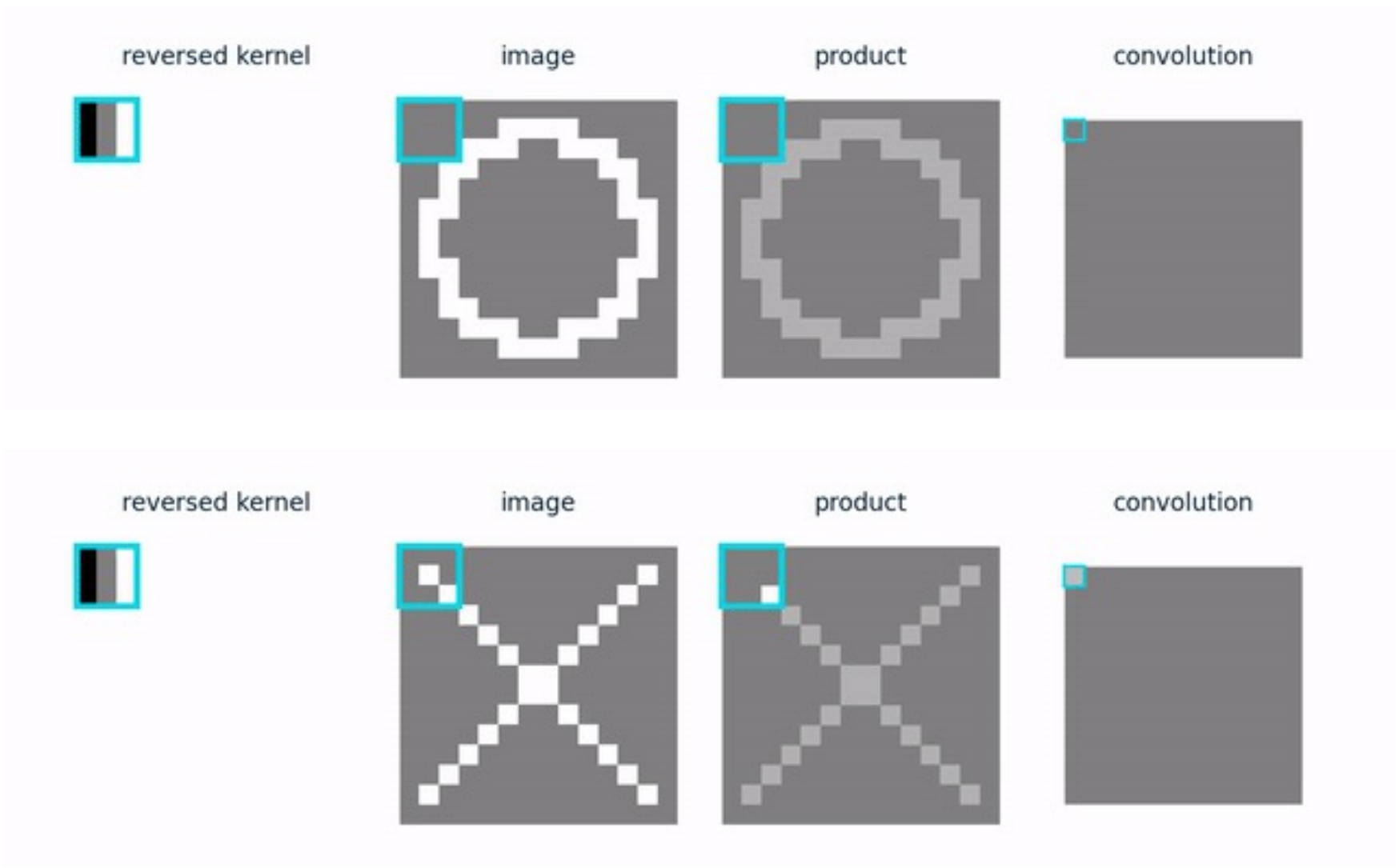
$$FM(i,j) = (I * F)(i,j) \sum_m \sum_n I(m,n) F(i-m, i-n)$$

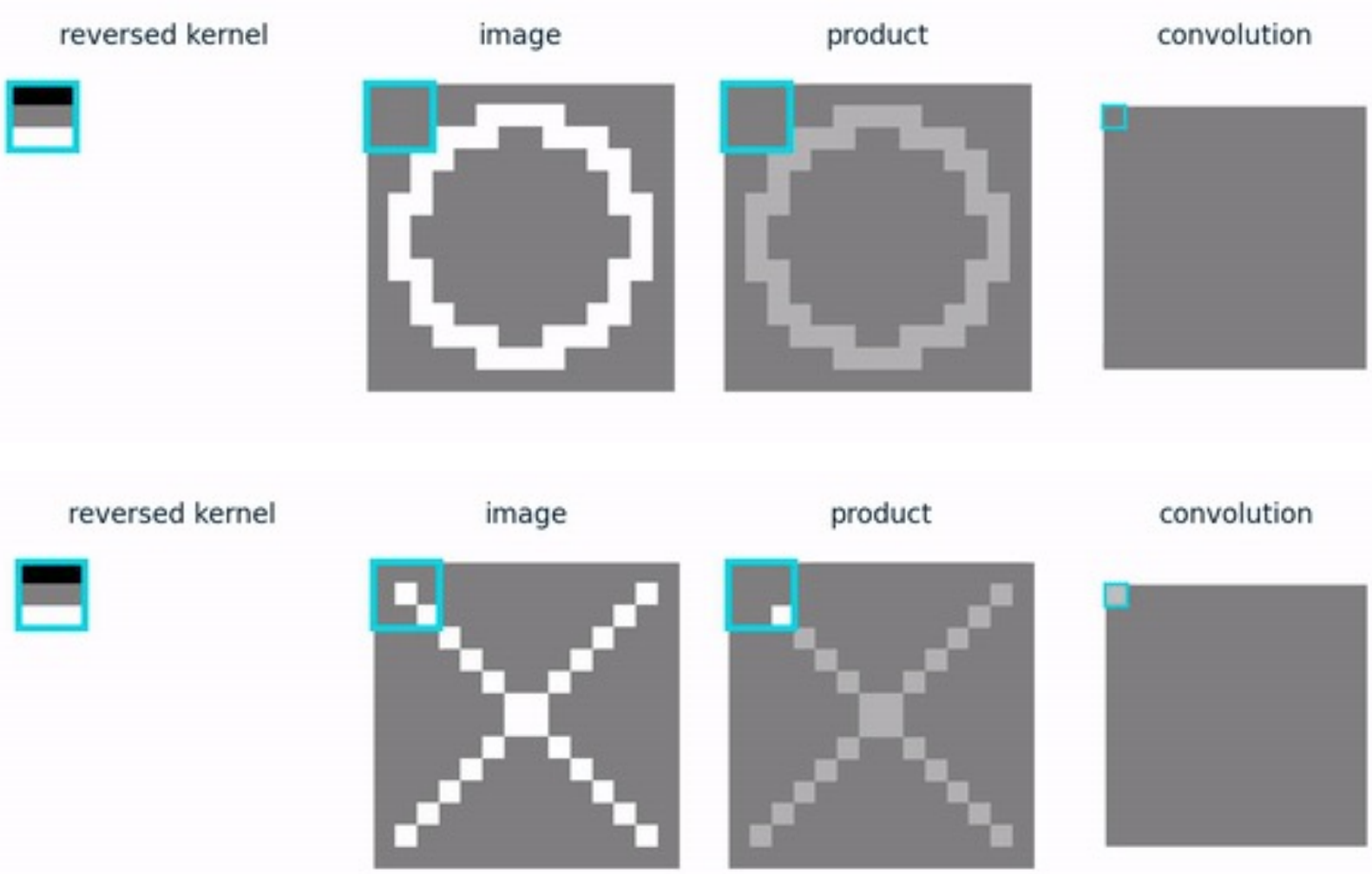
Input image

Filter/ kernel

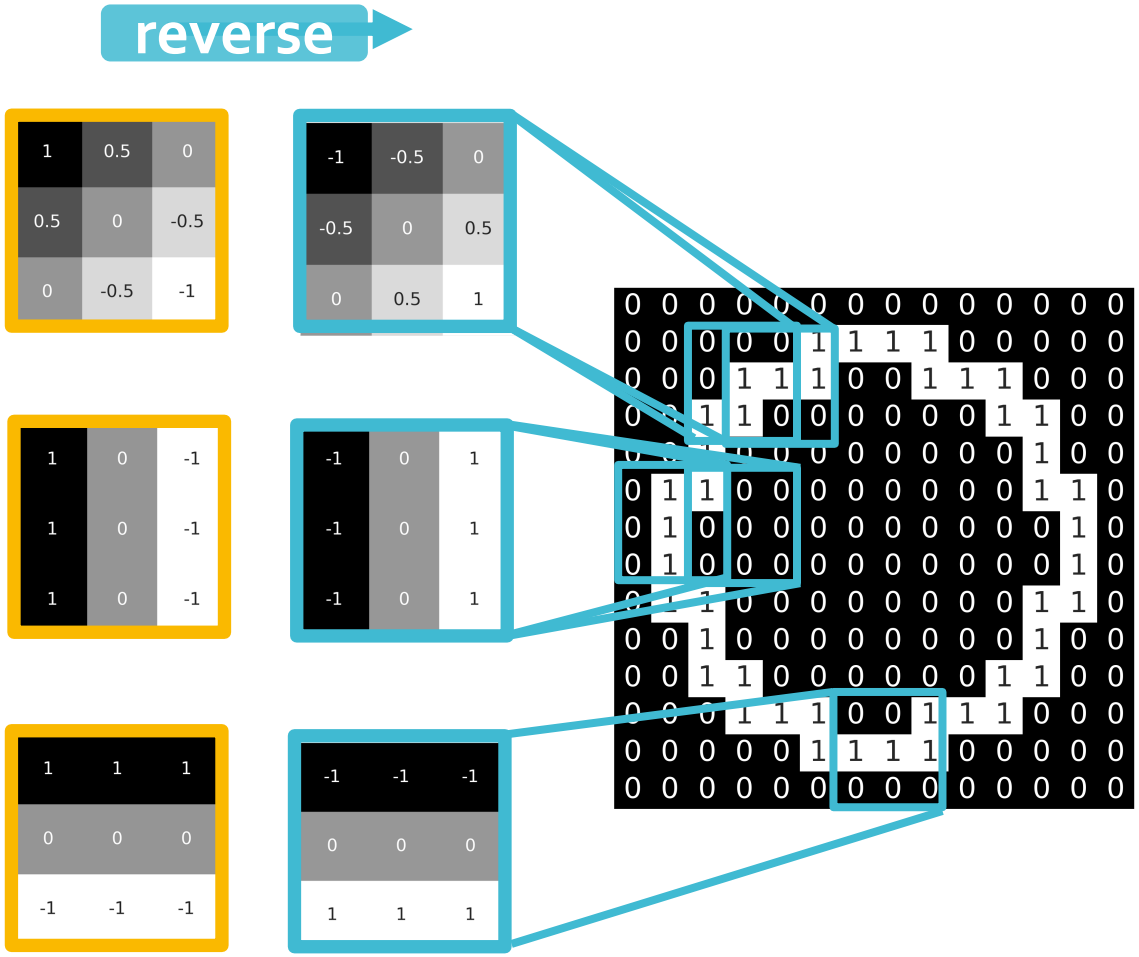


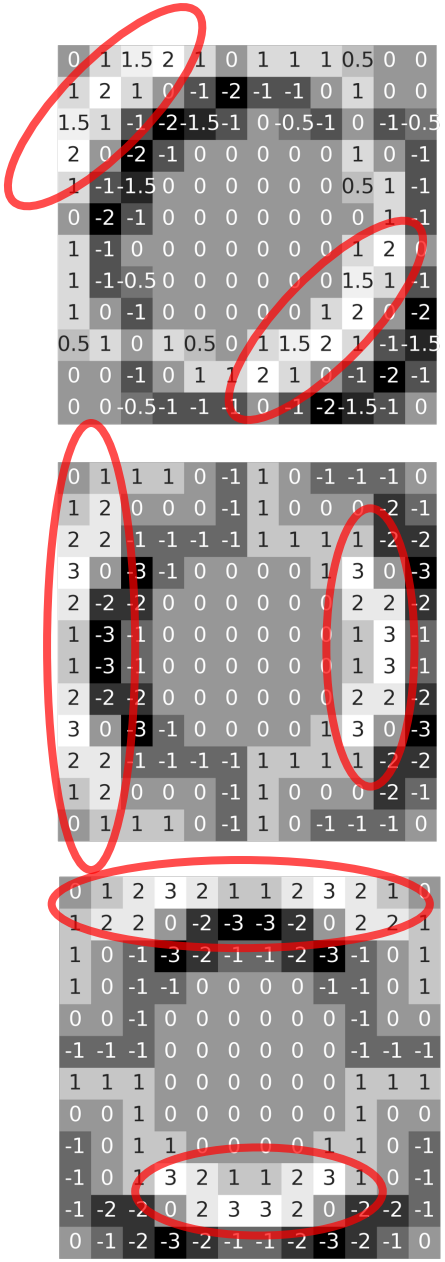
[Brandon Rohrer E2Eml]
https://gitlab.com/brohrer/convolution-2d-animation.git


reversed kernel
image
product
convolution
reversed kernel
image
product
convolution


reversed kernel
image
product
convolution
reversed kernel
image
product
convolution

Convolution layer





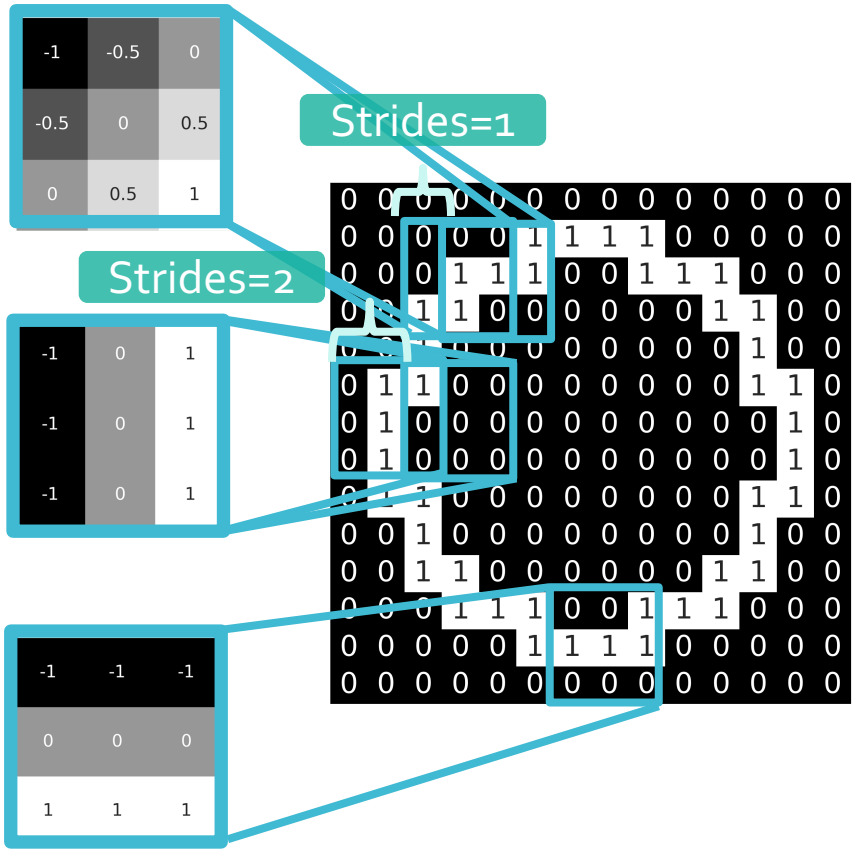
F_{size}: 3×3×3

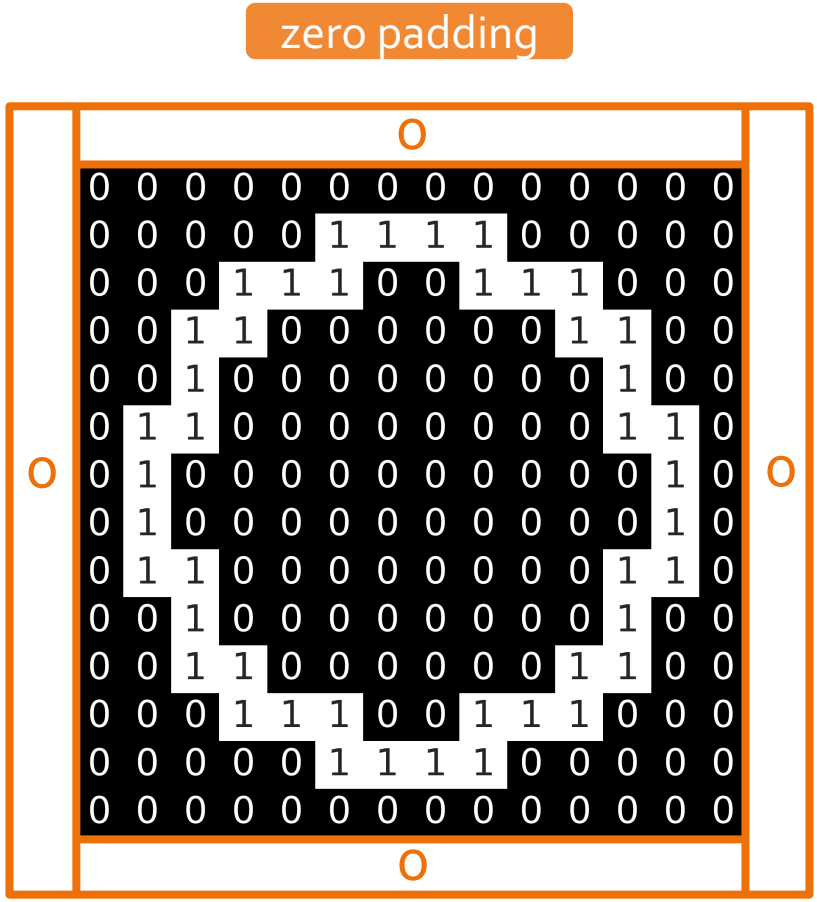
I_{size}: 14×14×1

FM_{size}: 12×12×3

$$FM_{size} = (I_{size} - F_{size} + 2P)/S + 1$$

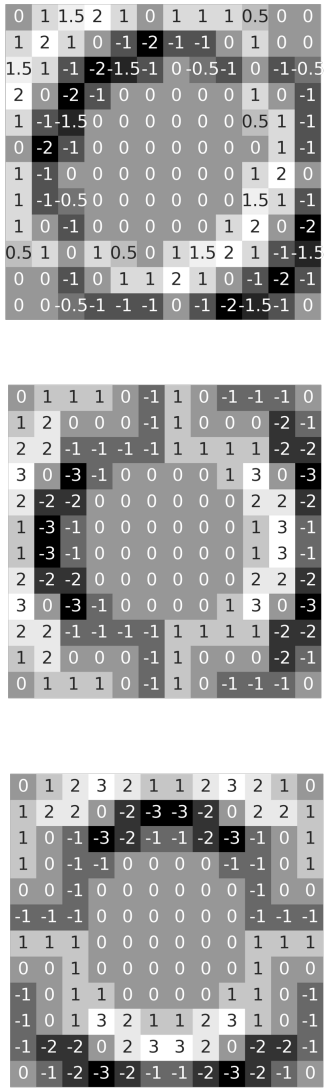
Convolution layer



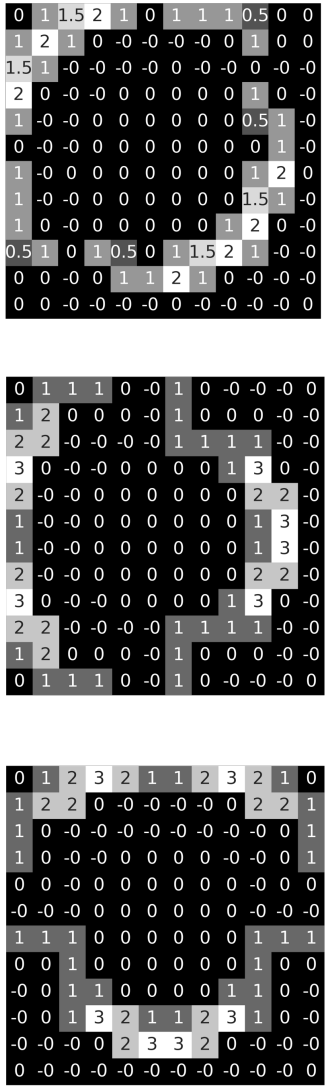


$$FM_{size} = (I_{size} - F_{size} + 2P)/S + 1$$

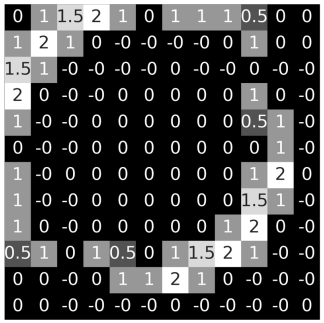
Convolution layer

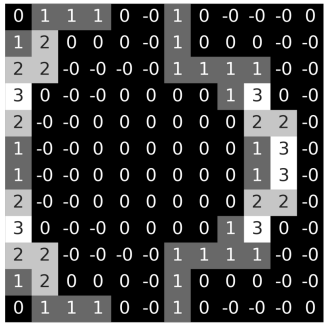


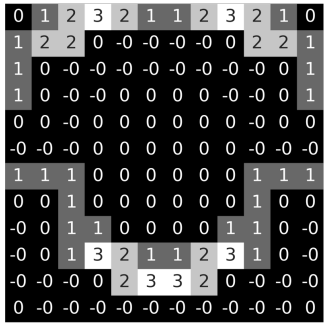
ReLU



Pooling layer



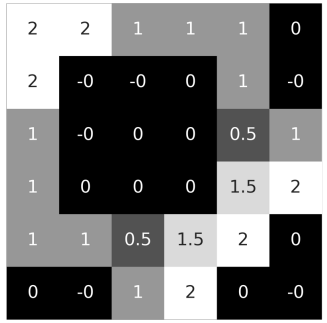


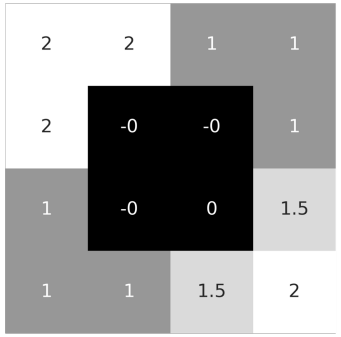


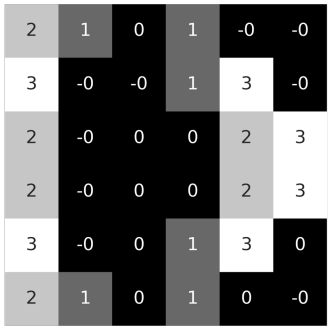
[2x2]
or
[6x6]

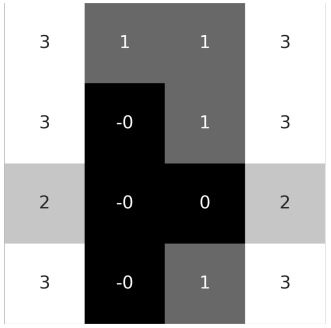
Strides=2

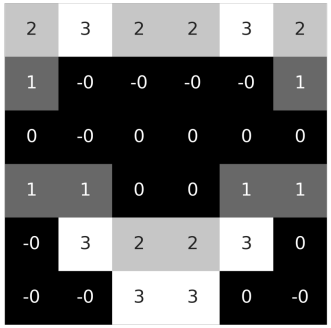
[2x2] [6x6]

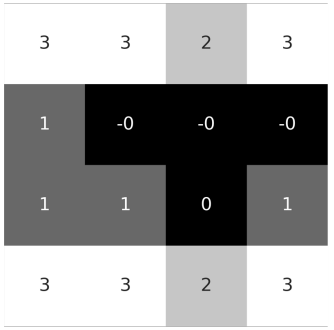








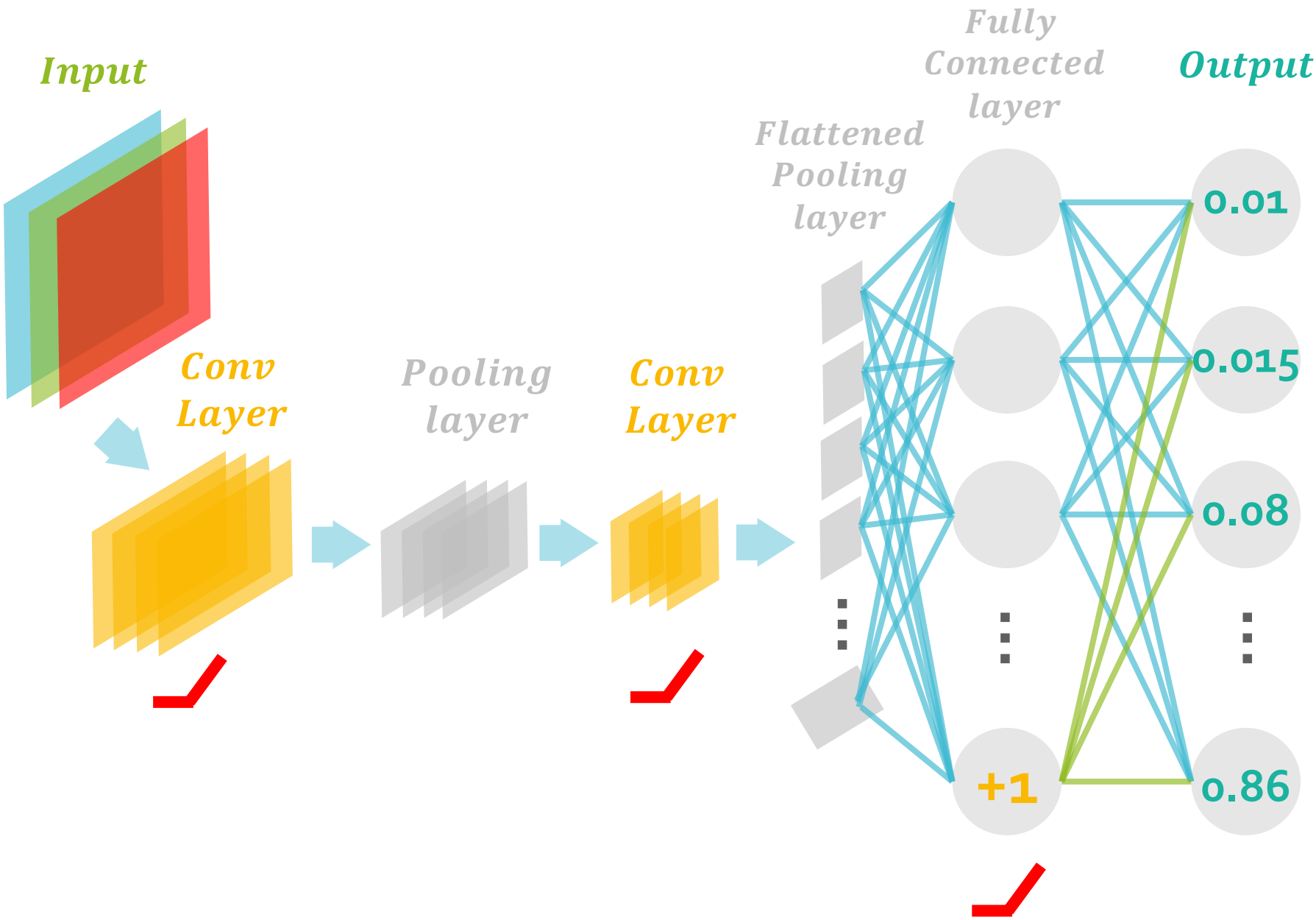




FM_{size}: 12×12×3

MP_{size}: 6×6×3 4×4×3

$MP_{size} = (FM_{size} - MPF_{size})/S + 1$


Input
Conv Layer
Pooling layer
Conv Layer
Flattened Pooling layer
Fully Connected layer
Output
0.01
0.015
0.08
0.86
+1

summary

Convolution layer

- Sparse interactions:
- kernel smaller than the input to lowerize parameter size
- detect small meaningful features
- Shared weights:
- correlation parts in the same fig
- Equivariant Representations, i.e. invariant to the change of positions

Pooling layer

- Downsampling
- summarize the responses over a whole neighborhood
- invariant to small translations of the input

MNIST dataset

[LeCun 1998]

Multiple Labels
$y \in \{0,1,2,\ldots,9\}$

in python keras

```
cnn=Sequential()
cnn.add(Conv2D(filters=16,kernel_size=(3,3),activation='relu',input_shape=(28,28,1)))
cnn.add(MaxPooling2D((2,2)))
cnn.add(Conv2D(filters=16,kernel_size=(3,3),activation='relu'))
cnn.add(MaxPooling2D((2,2)))
cnn.add(Flatten())
cnn.add(Dense(32,activation='relu'))
cnn.add(Dense(10,activation='softmax'))
cnn.summary()

cnn.compile(loss='categorical_crossentropy',
            optimizer='adam',
            metrics=['categorical_accuracy'])

traj=cnn.fit(x_train,y_train,epochs=20,batch_size=32,verbose=1,
             shuffle=True,validation_data=(x_test, y_test))
```

Build model

Training

x_train: (60000,28,28,1)
y_train: (60000,10) - one hot
x_test: (10000,28,28,1)
y_test: (10000,10) – one hot

summary

```
Model: "sequential_17"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_61 (Conv2D)           (None, 26, 26, 16)        160

 max_pooling2d_32 (MaxPoolin  (None, 13, 13, 16)        0
 g2D)

 conv2d_62 (Conv2D)           (None, 11, 11, 16)        2320

 max_pooling2d_33 (MaxPoolin  (None, 5, 5, 16)          0
 g2D)

 flatten_16 (Flatten)         (None, 400)               0

 dense_32 (Dense)             (None, 32)                12832

 dense_33 (Dense)             (None, 10)                330

=================================================================
Total params: 15,642
Trainable params: 15,642
Non-trainable params: 0
_________________________________________________________________
```

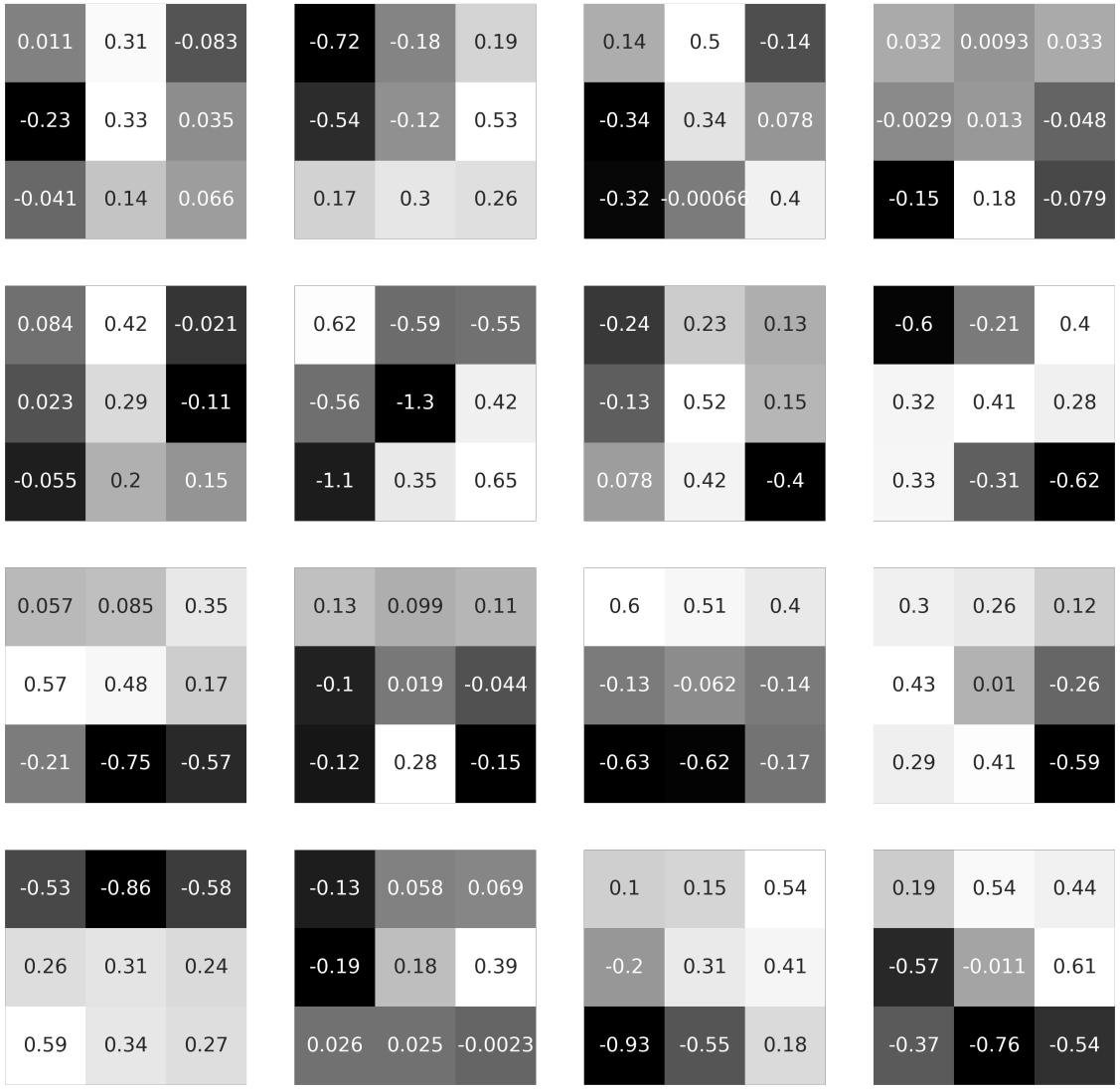
layer.0

layer.1

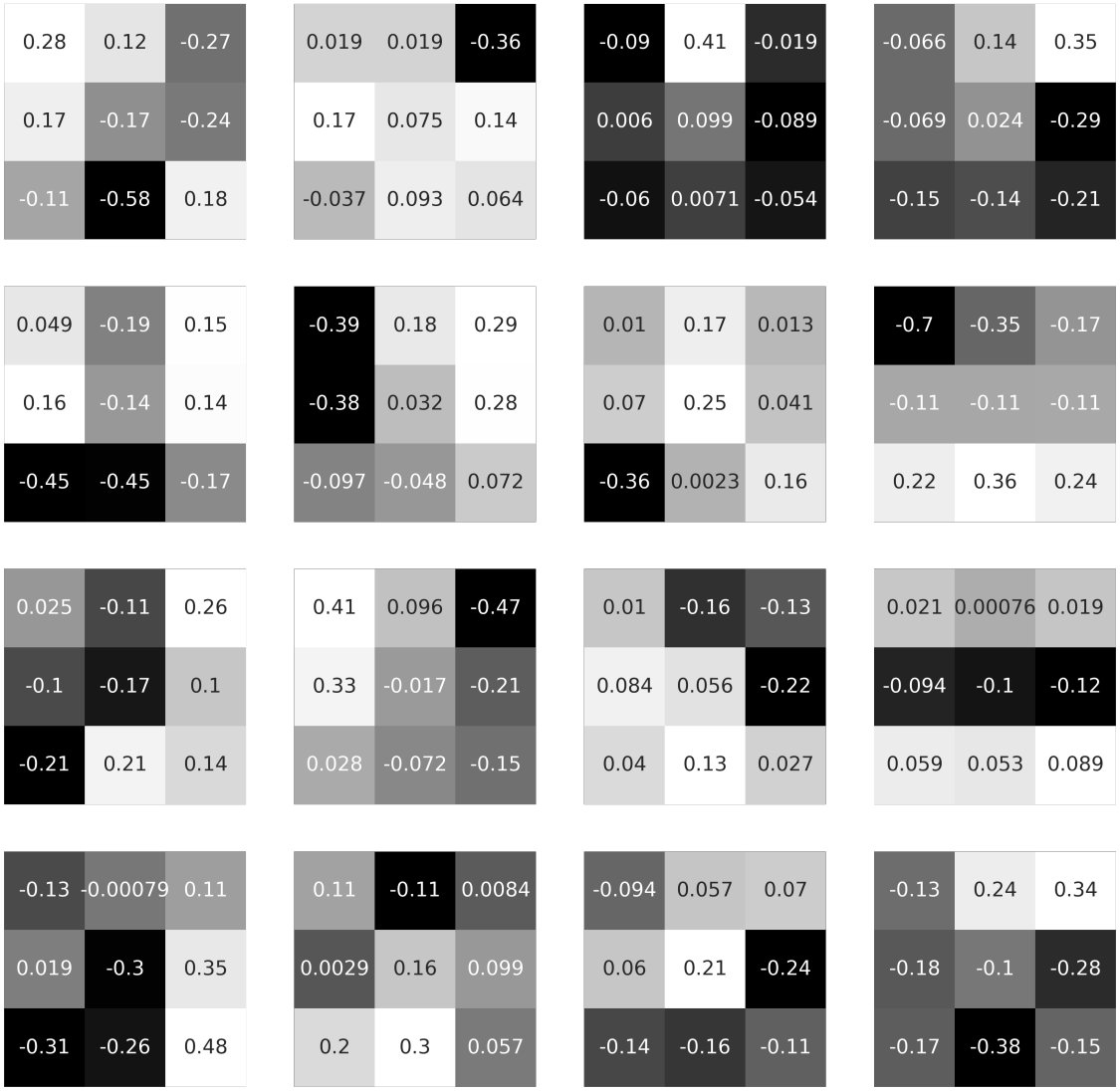
layer.2

layer.3

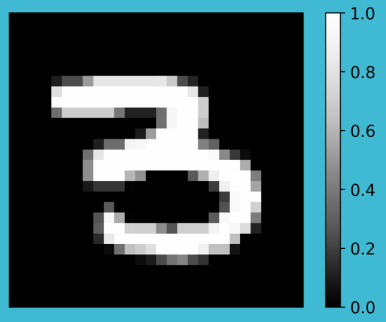
Filter Weights [layer.0]



Filter Weights [layer.2]



Activations of each layer

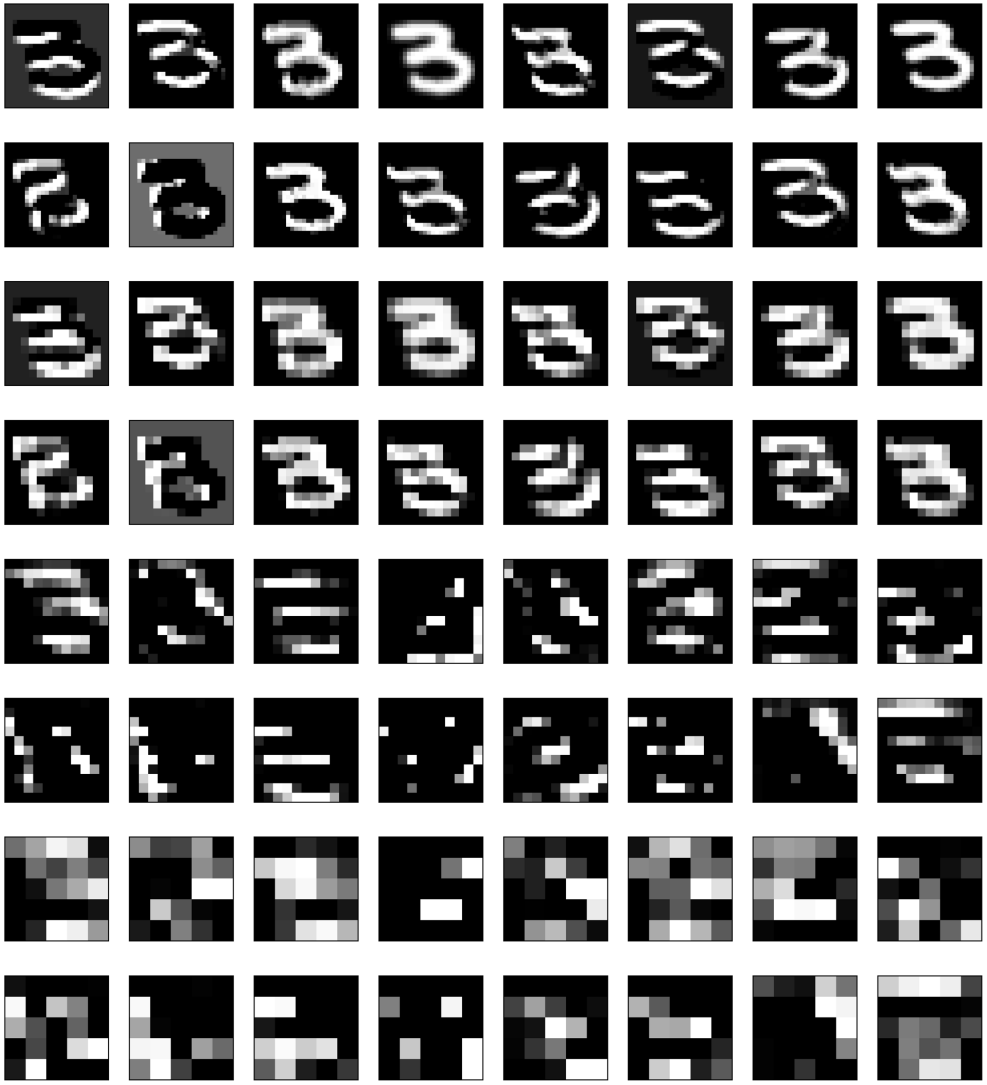


layer.0
26x26

layer.1
13x13

layer.2
11x11

layer.3
5x5



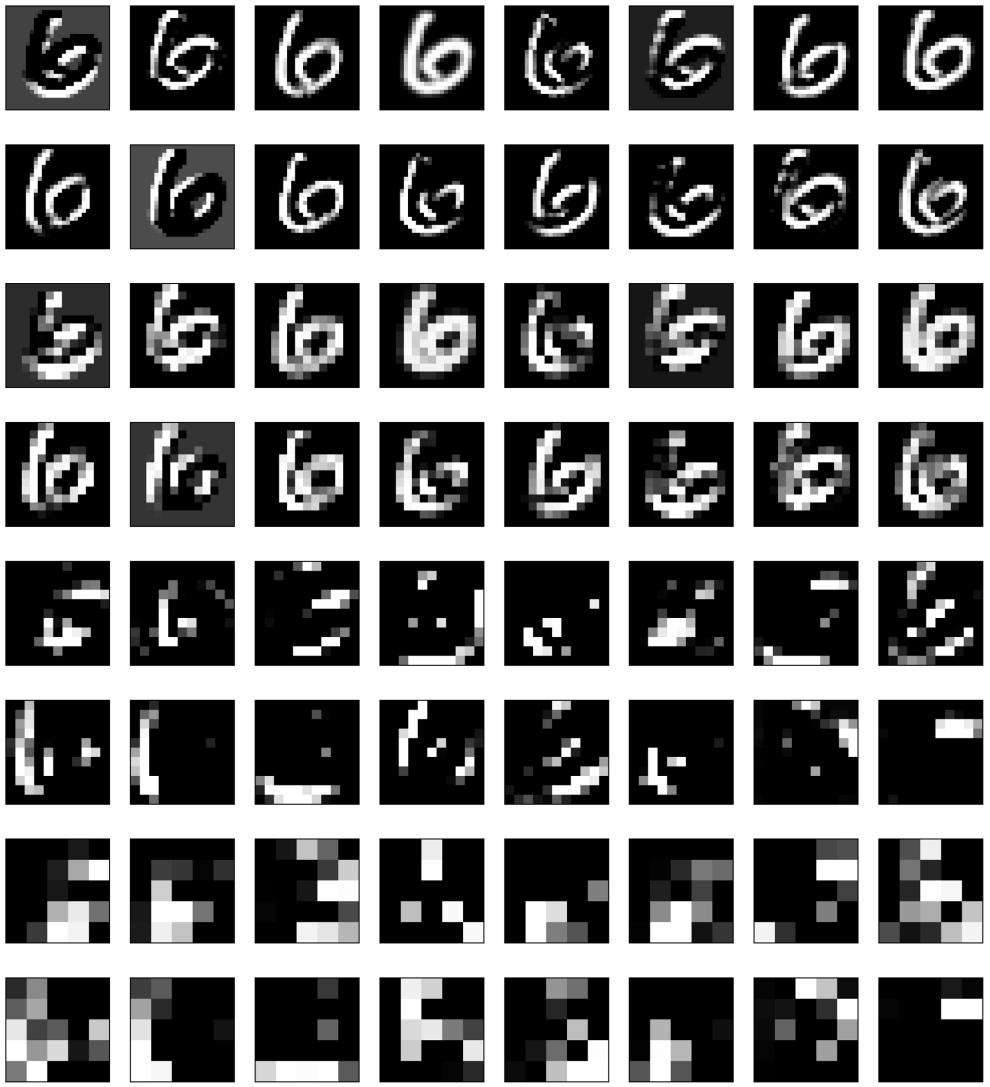
Activations of each layer

layer.0
26x26

layer.1
13x13

layer.2
11x11

layer.3
5x5



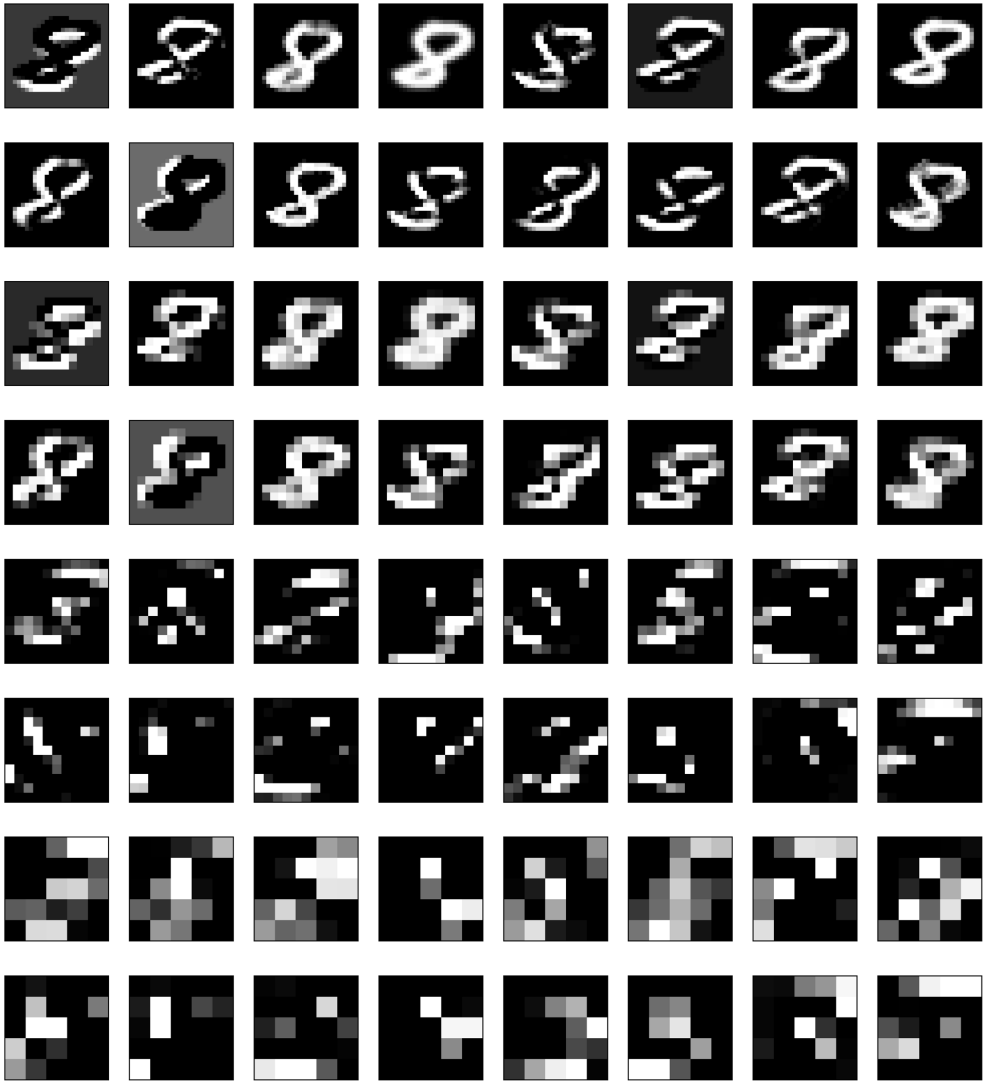
Activations of each layer

layer.0
26x26

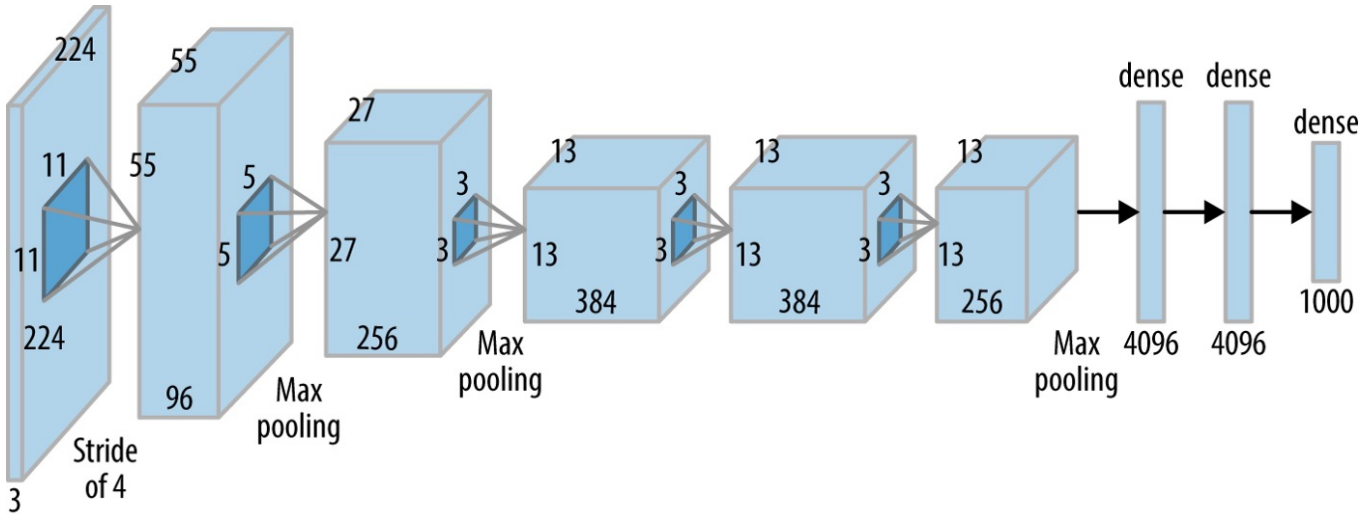
layer.1
13x13

layer.2
11x11

layer.3
5x5



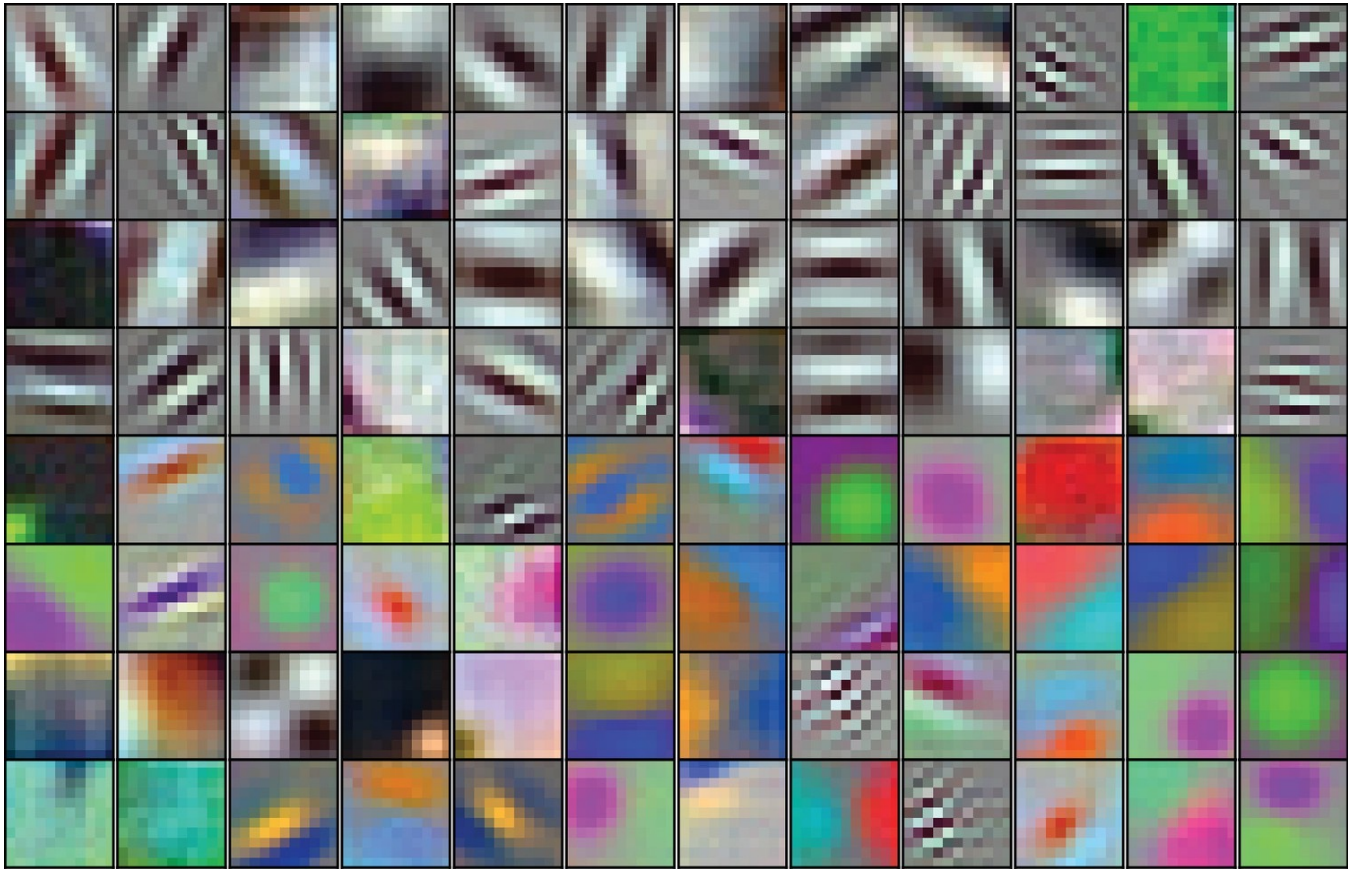
Alexnet
[Krizhevsky et al. 2012]



[ImageNet Large Scale Visual Recognition Challenge 2012]

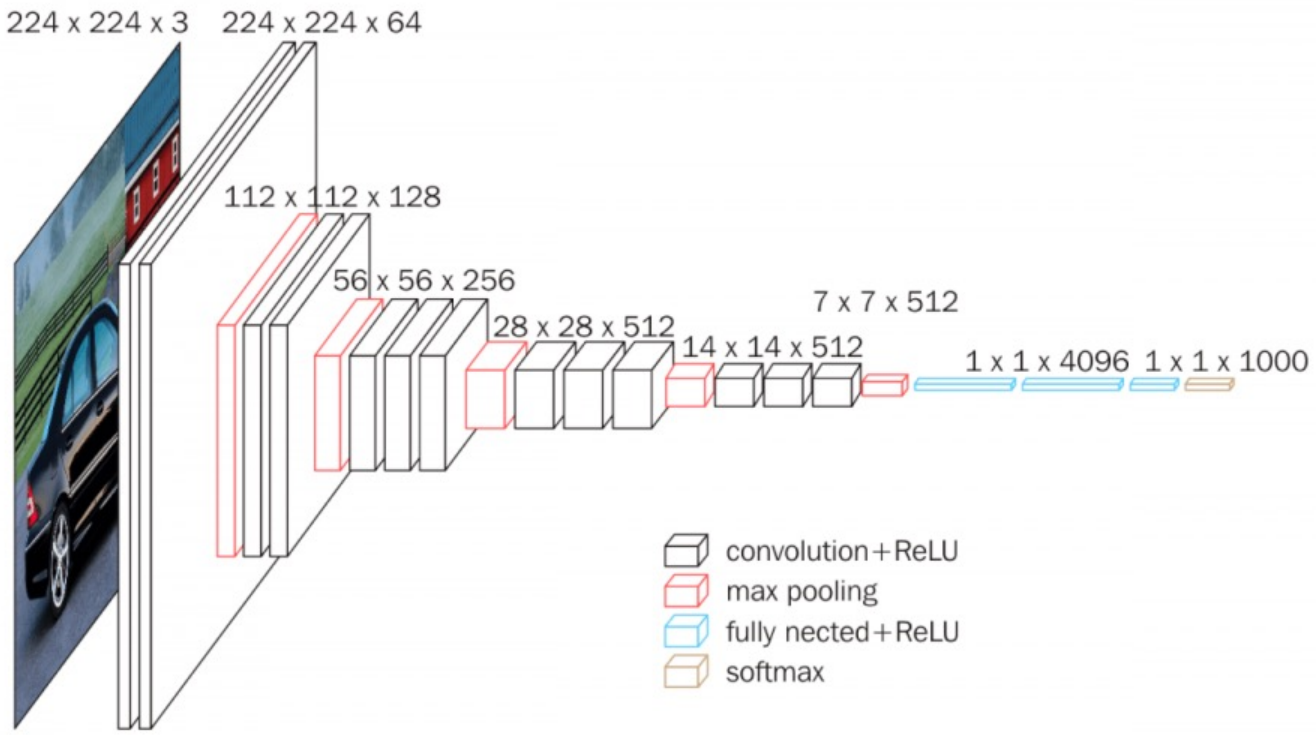
Alexnet

[Krizhevsky et al. 2012]



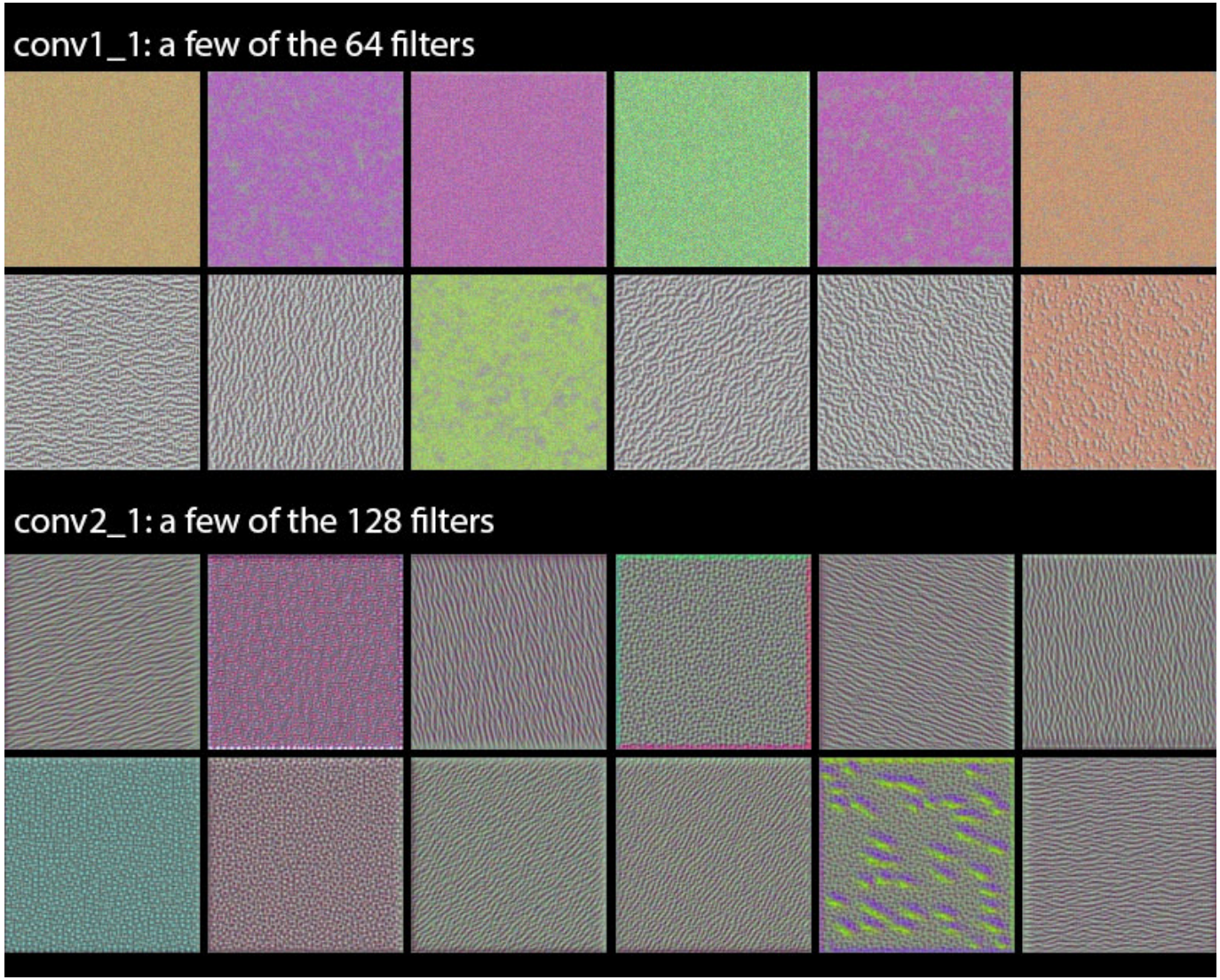
VGG

[Simonyan et al. 2014]



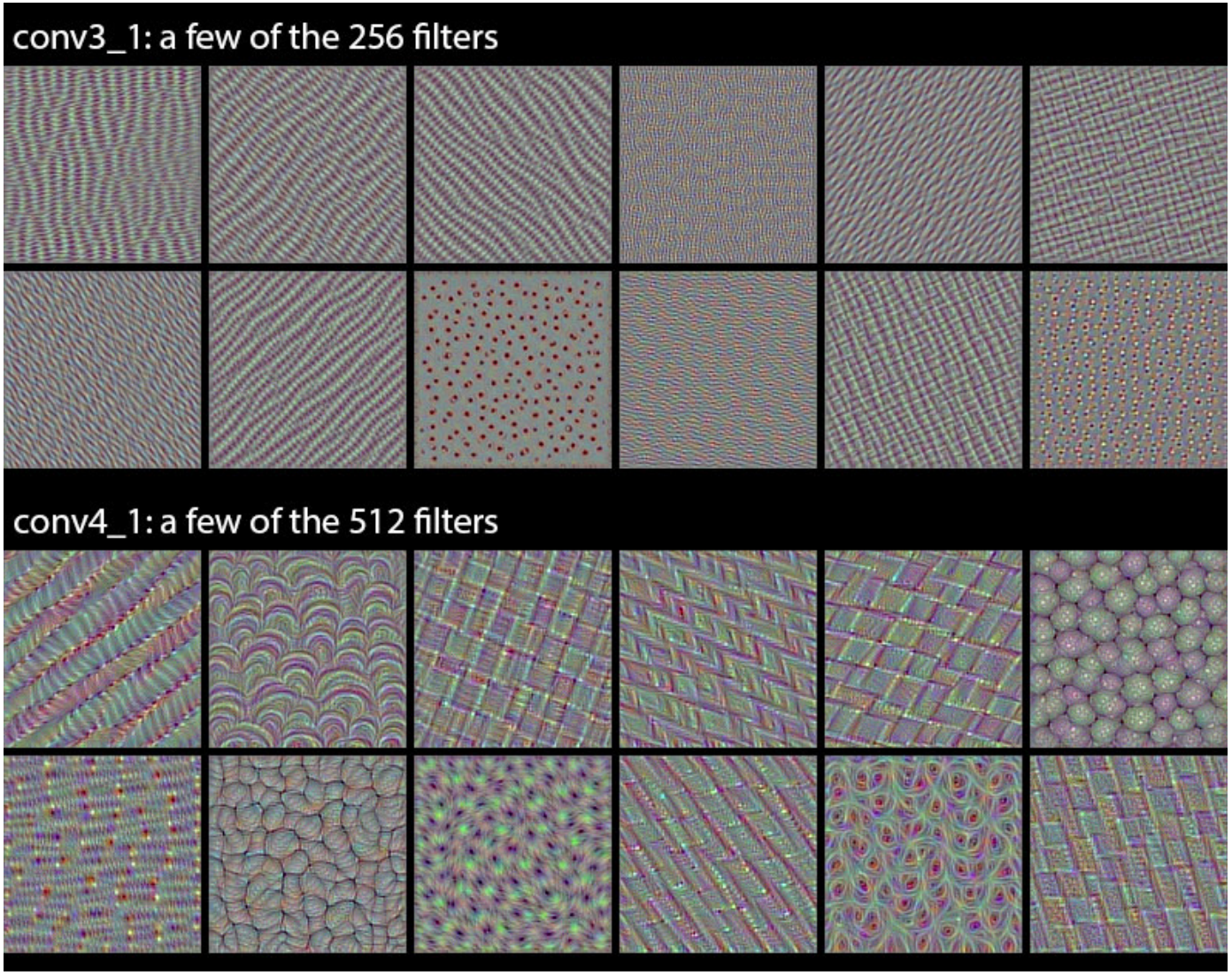
VGG

[Simonyan et al. 2014]



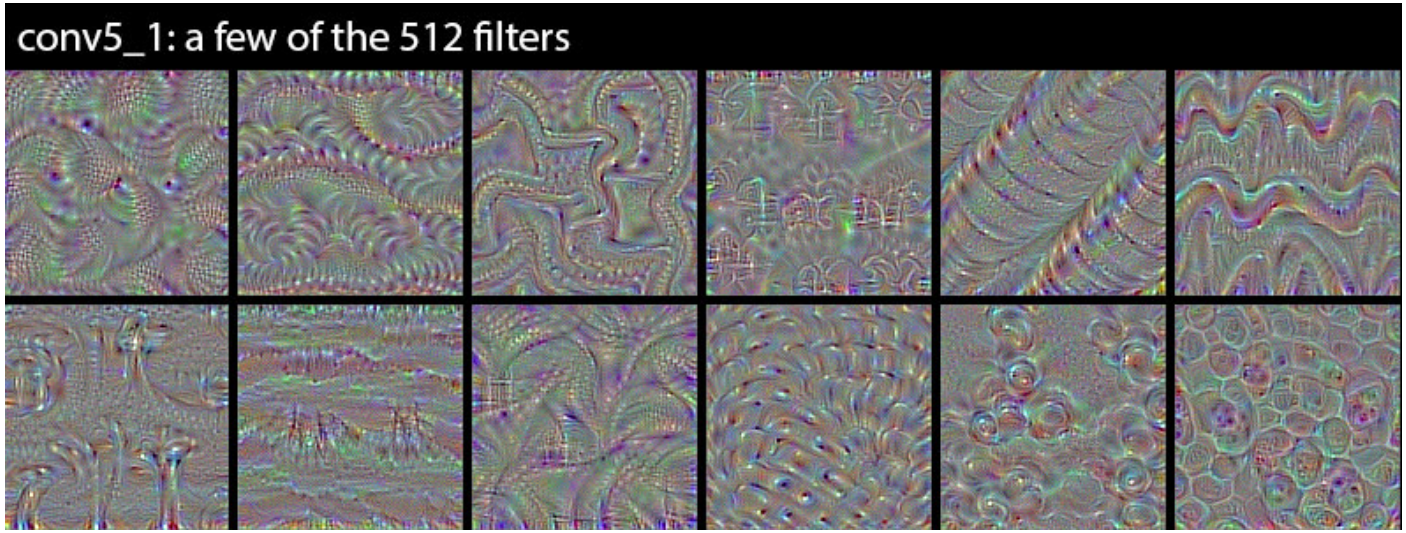
VGG

[Simonyan et al. 2014]



VGG

[Simonyan et al. 2014]



Interesting extensions

- Inceptionism: Going Deeper into Neural Networks

https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

- Understanding Neural Networks Through Deep Visualization

https://yosinski.com/deepvis

- Feature Visualization

https://distill.pub/2017/feature-visualization/