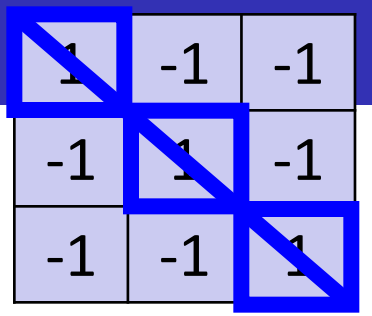
CNNs

CV Jawahar

IIIT-H

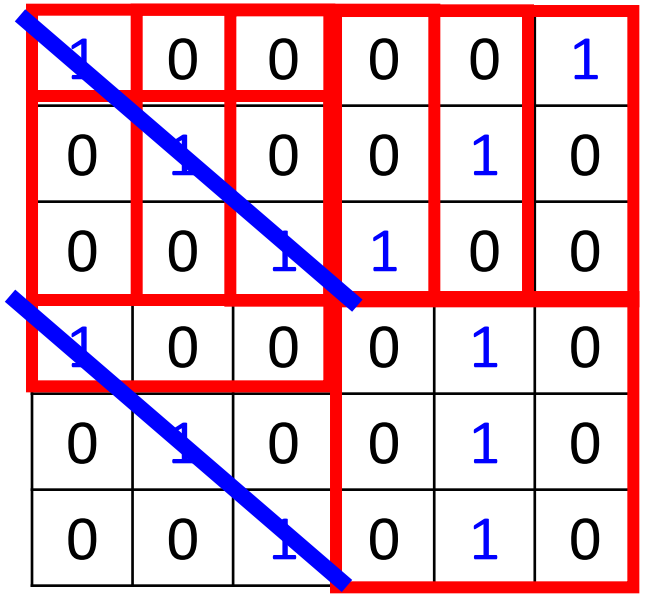
11 April 2025

Convolution

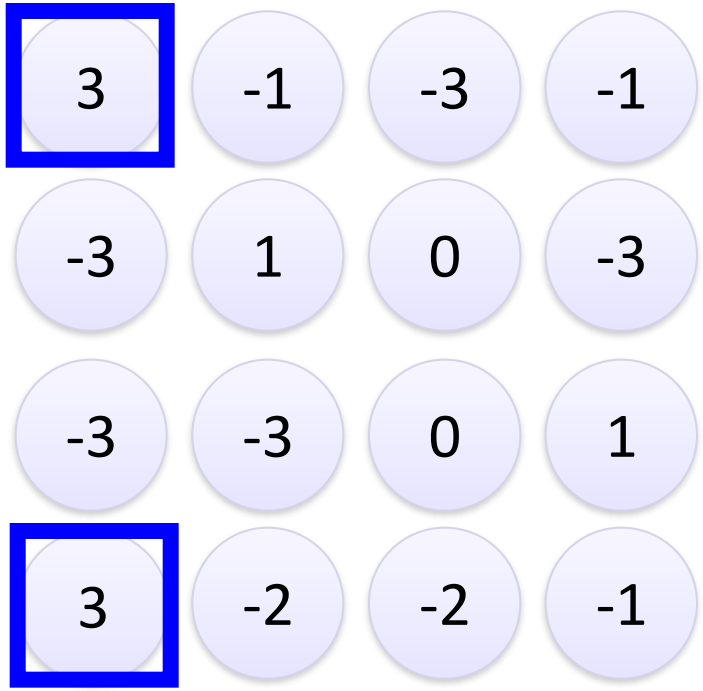


Filter 1

stride=1

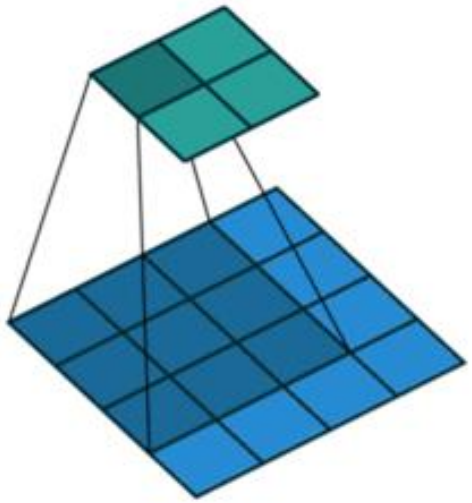


6 x 6 image

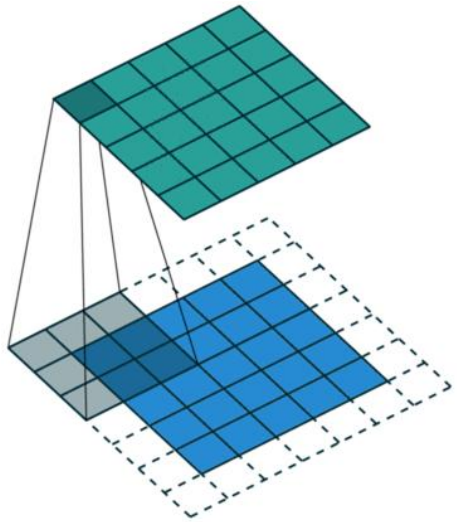


Revisit: Convolution layer

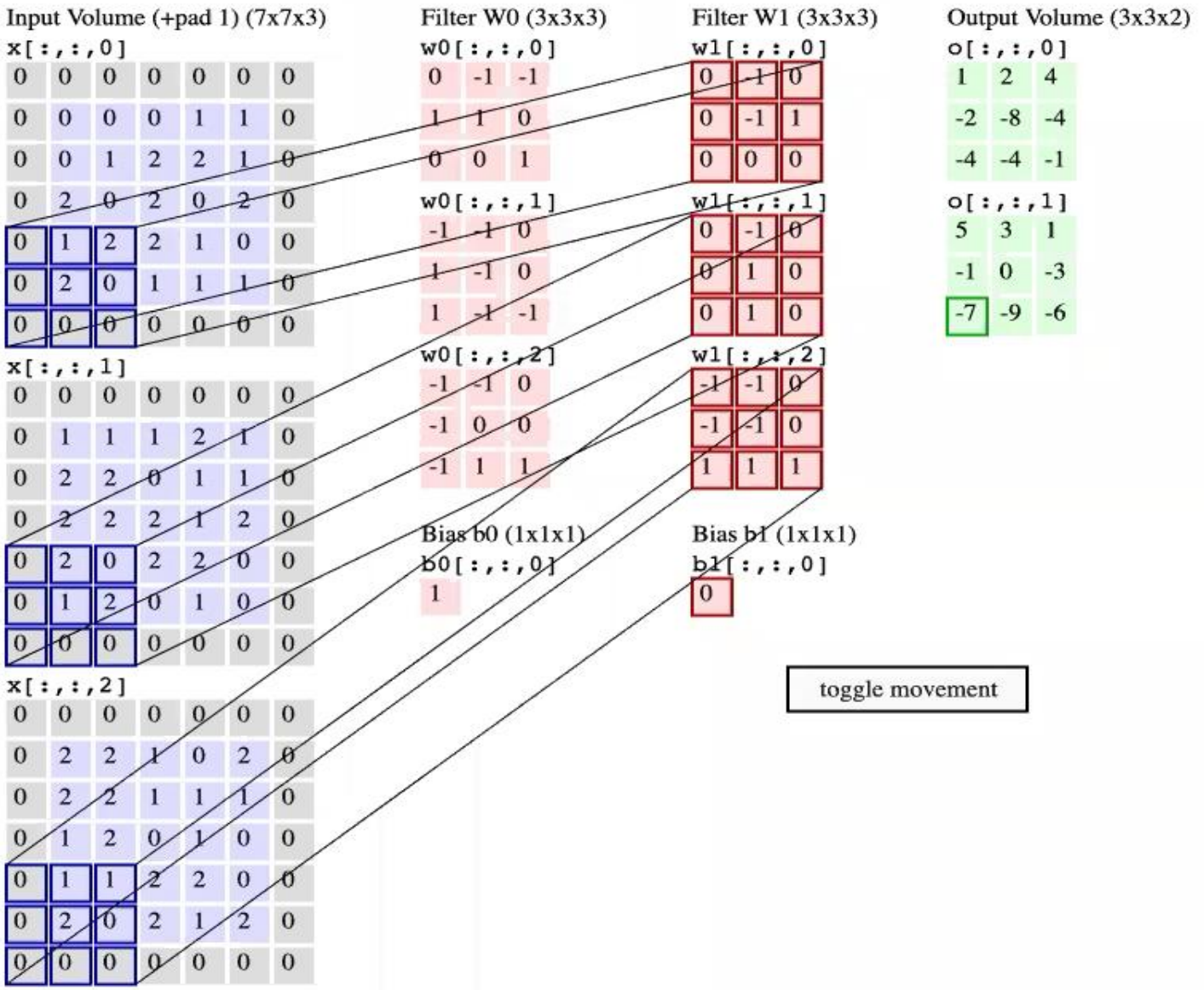
- Window size
- Stride
- Padding
- Pool



Window size: 3x3
Stride: 1
Padding: 0



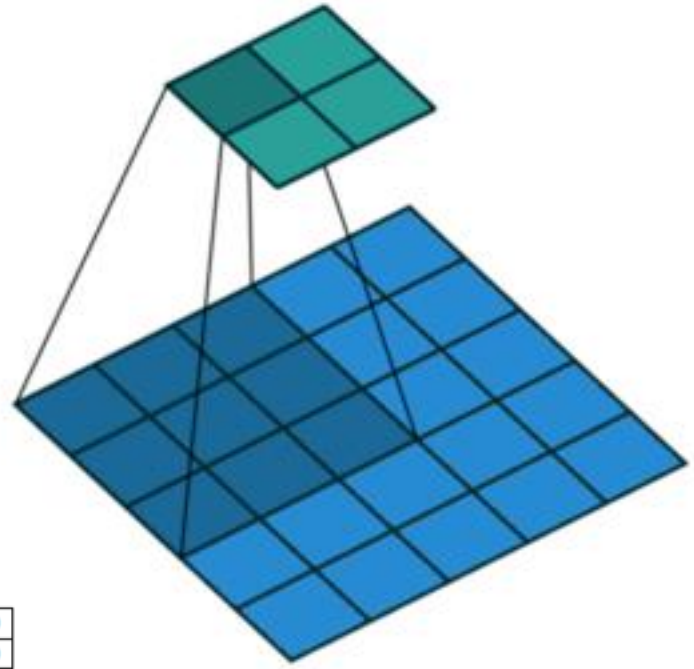
Window size: 3x3
Stride: 1
Padding: 1

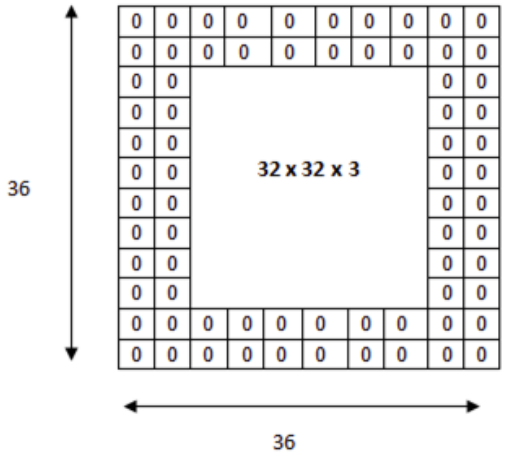

Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 0 0 1 1 0
0 0 1 2 2 1 0
0 2 0 2 0 2 0
0 1 2 2 1 0 0
0 2 0 1 1 1 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 1 1 2 1 0
0 2 2 0 1 1 0
0 2 2 2 1 2 0
0 2 0 2 2 0 0
0 1 2 0 1 0 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 2 2 1 0 2 0
0 2 2 1 1 1 0
0 1 2 0 1 0 0
0 1 1 2 2 0 0
0 2 0 2 1 2 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
0 -1 -1
1 1 0
0 0 1
w0[:,:,1]
-1 -1 0
1 -1 0
1 -1 -1
w0[:,:,2]
-1 -1 0
-1 0 0
-1 1 1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
0 -1 0
0 -1 1
0 0 0
w1[:,:,1]
0 -1 0
0 1 0
0 1 0
w1[:,:,2]
-1 -1 0
-1 -1 0
1 1 1
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
1 2 4
-2 -8 -4
-4 -4 -1
o[:,:,1]
5 3 1
-1 0 -3
-7 -9 -6
toggle movement

Back to Top

CNNs

- Strides reduces dimension

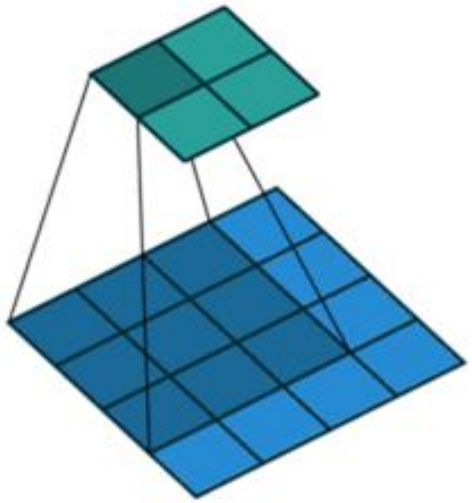




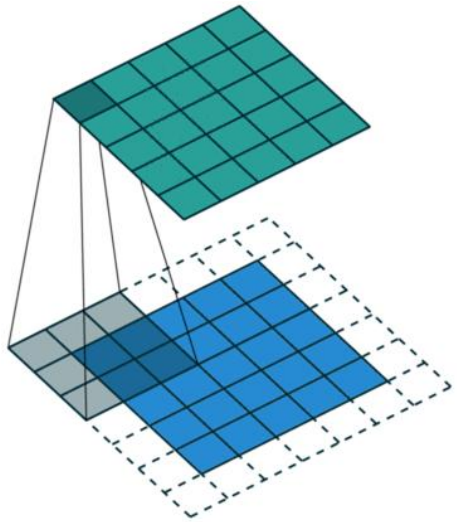
$$O = \frac{(W - K + 2P)}{S} + 1$$

Revisit: Convolution layer

- Window size
- Stride
- Padding
- Pool



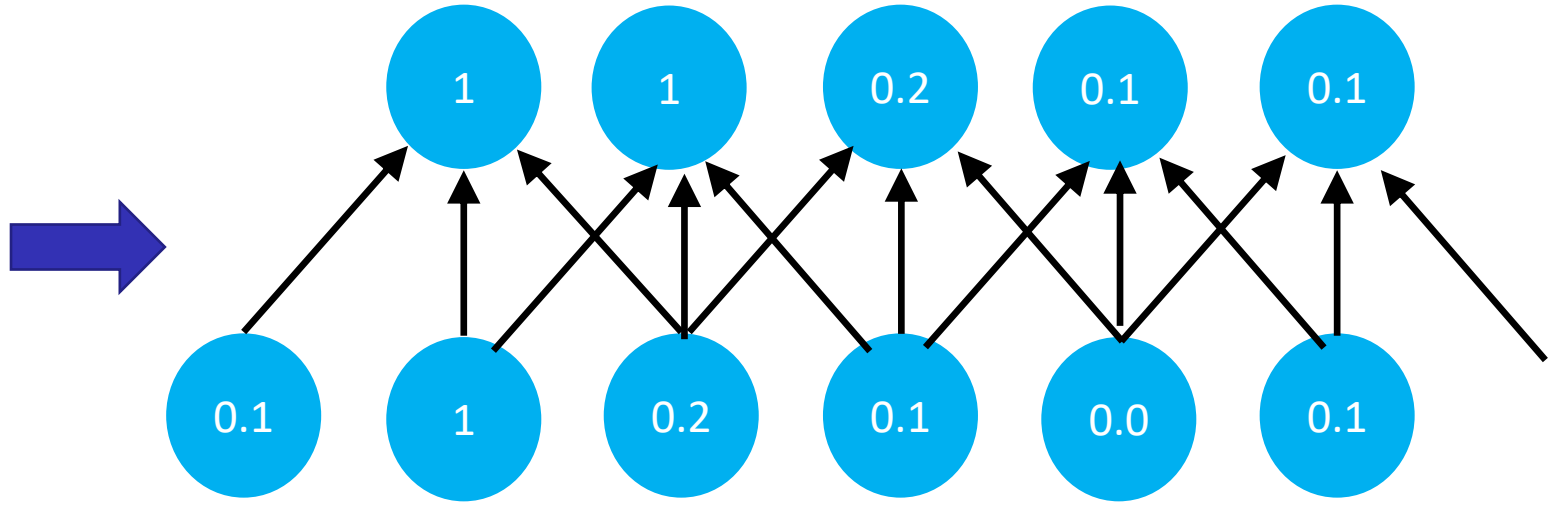
Window size: 3x3
Stride: 1
Padding: 0

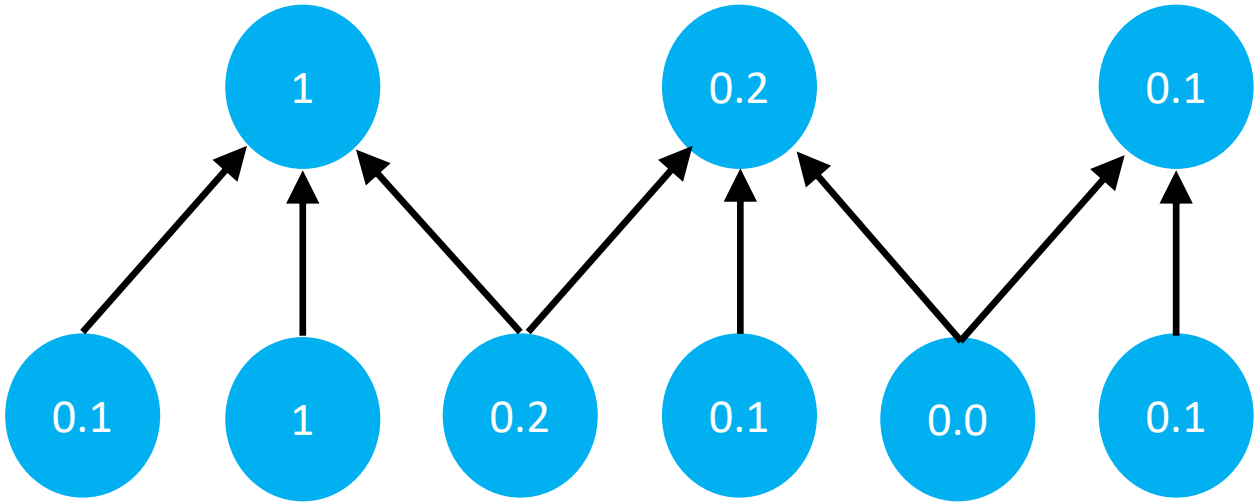


Window size: 3x3
Stride: 1
Padding: 1

Max Pool and Stride

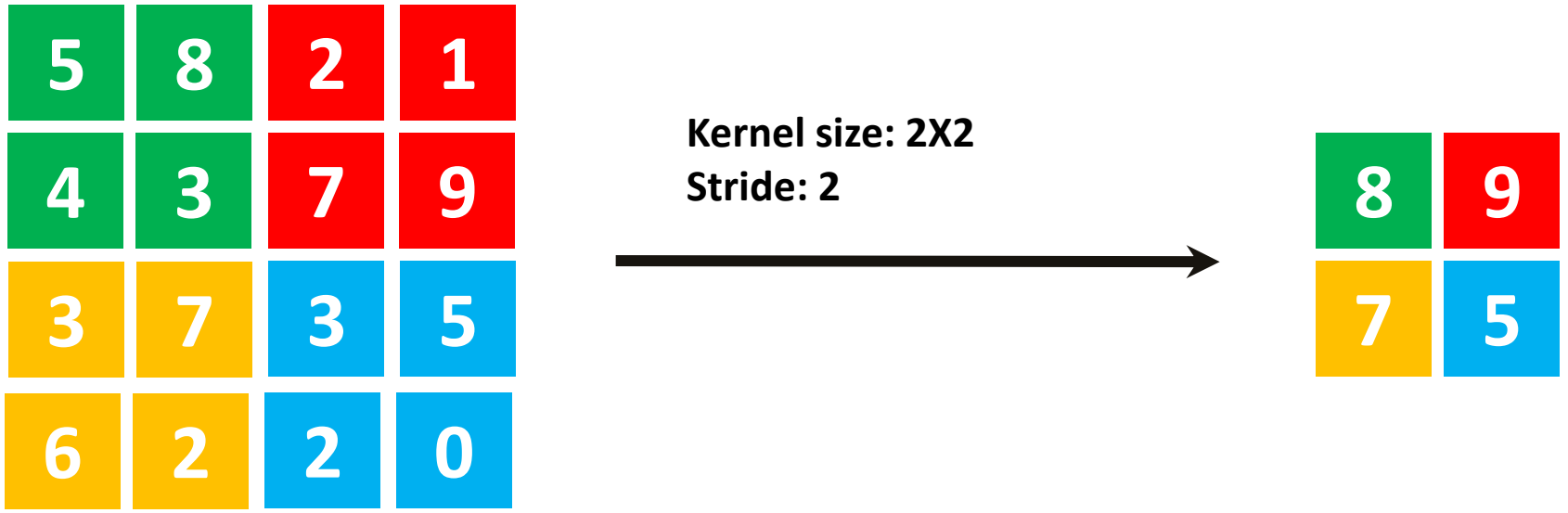
- Window Size = 3
- Stride =1





- Window Size = 3
- Stride =2

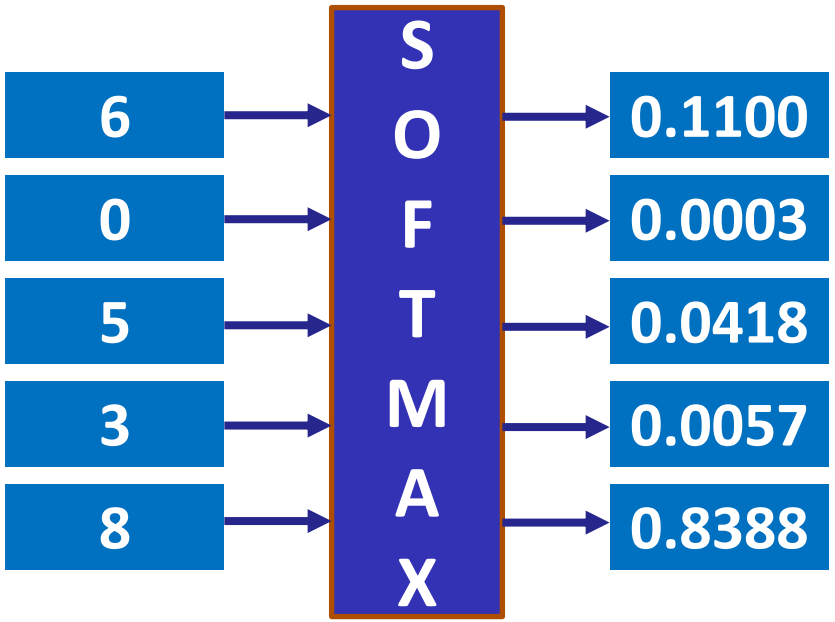
Max pooling in 2-D



Softmax

- Normalizes the output.
- K is total number of classes

$$z_n = \frac{e^{x_n}}{\sum_{i=1}^{K} e^{x_i}}$$



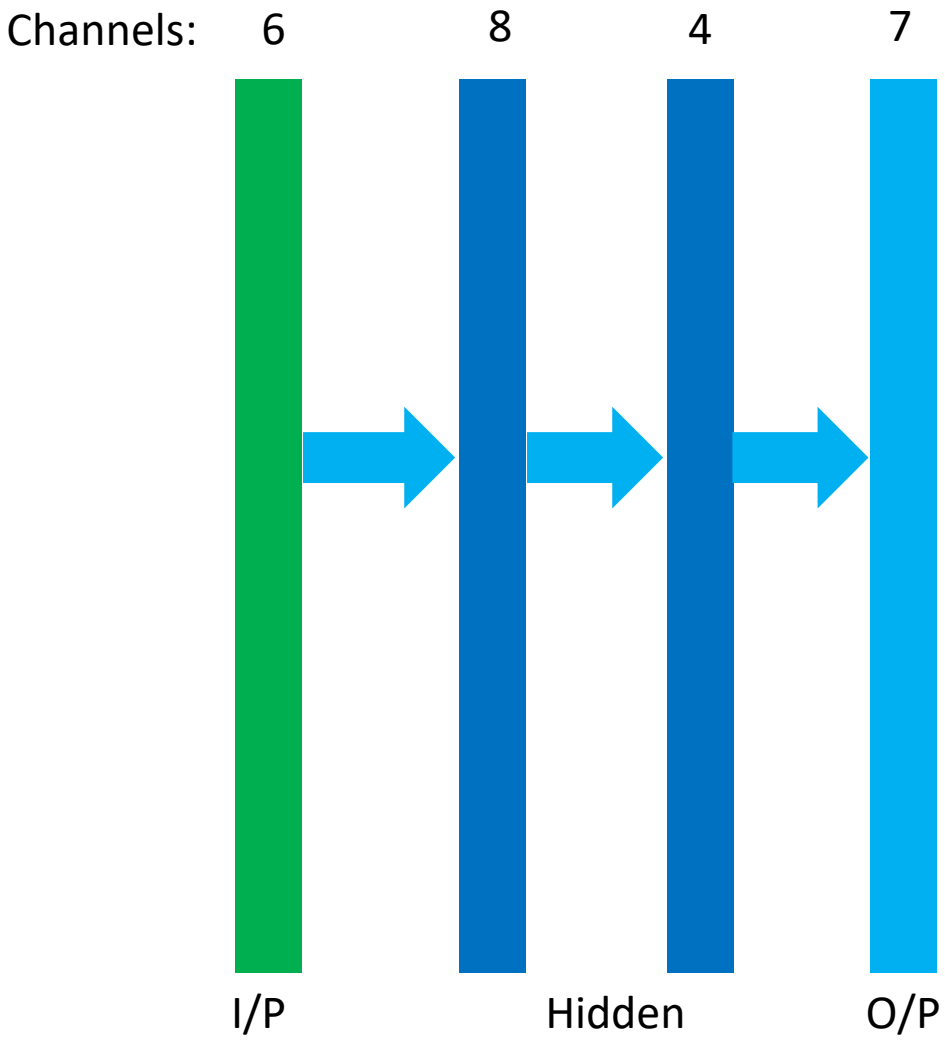
Terminologies

- # Input Channels
- # Output channels

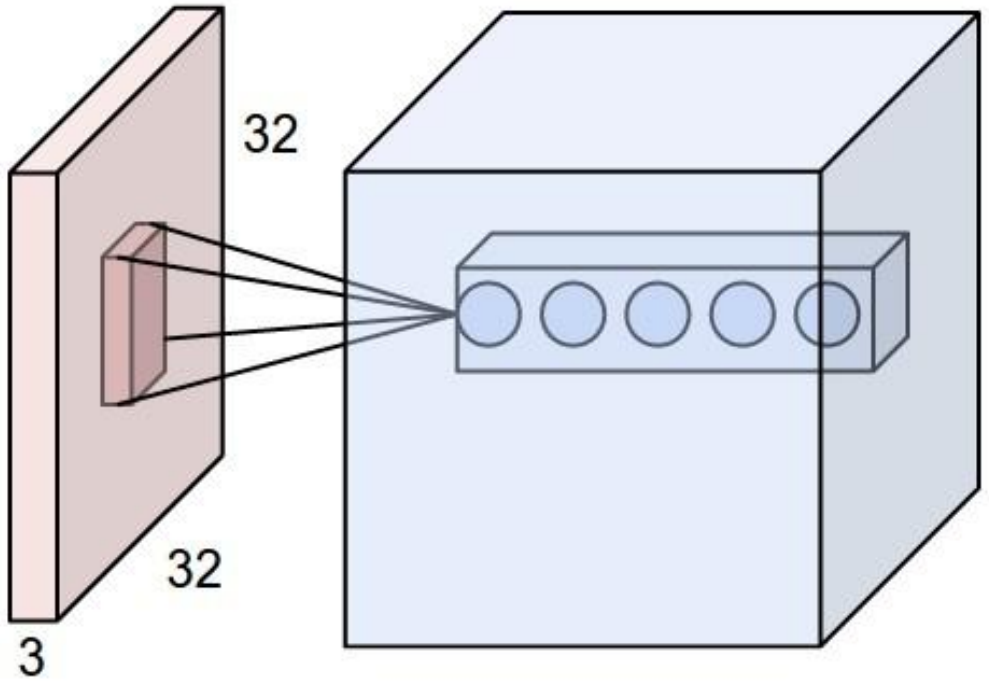
- Feature Maps/Channels
- Filters/Weights
- Filter Size/Window Size

- Stride
- Pooling (Max/Average)
- Fully Connected Layer
- Soft-Max
- Normalization
- Flattening
- Convolution Layer

Layer wise abstraction



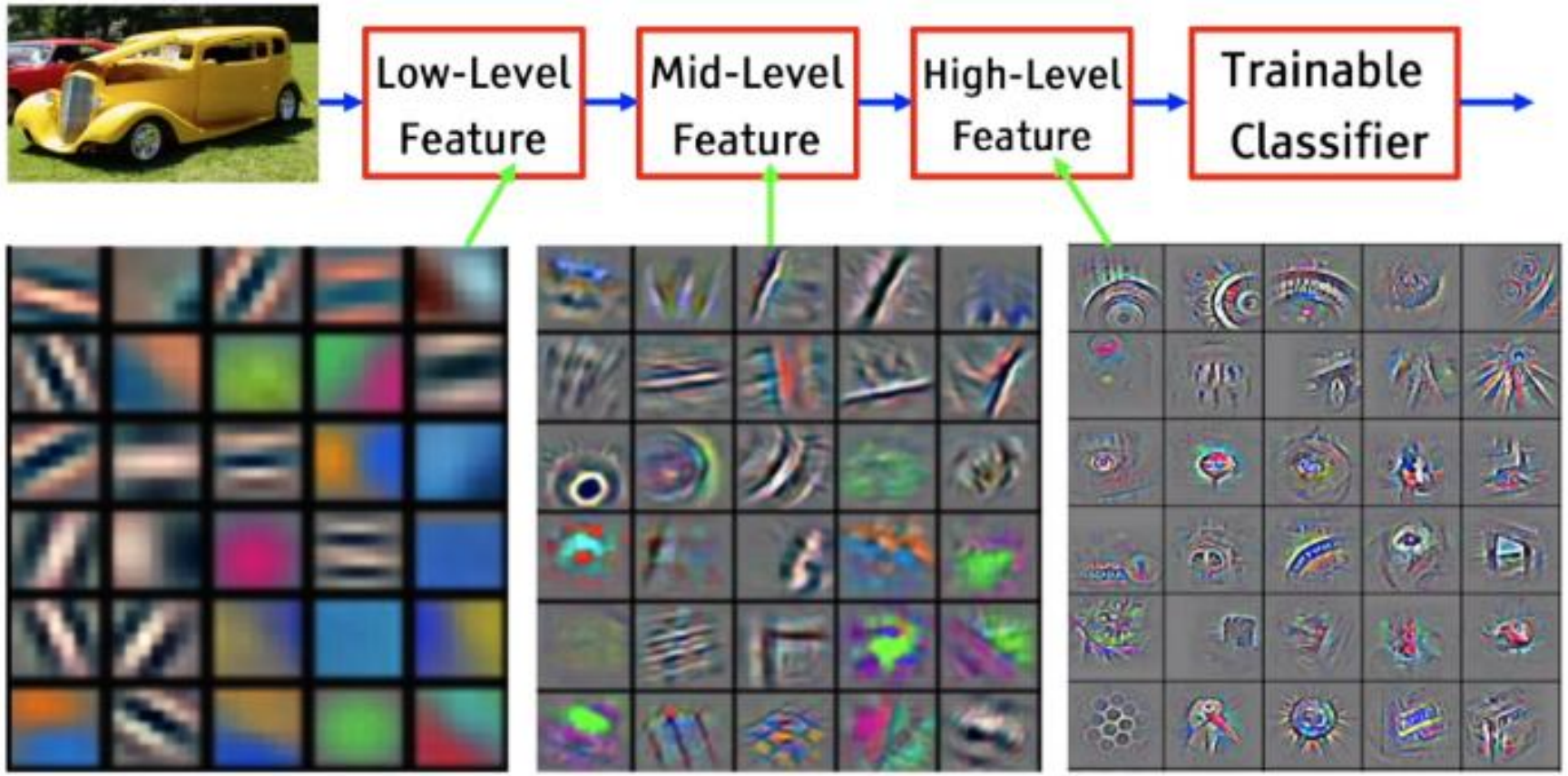
1-D Convolution



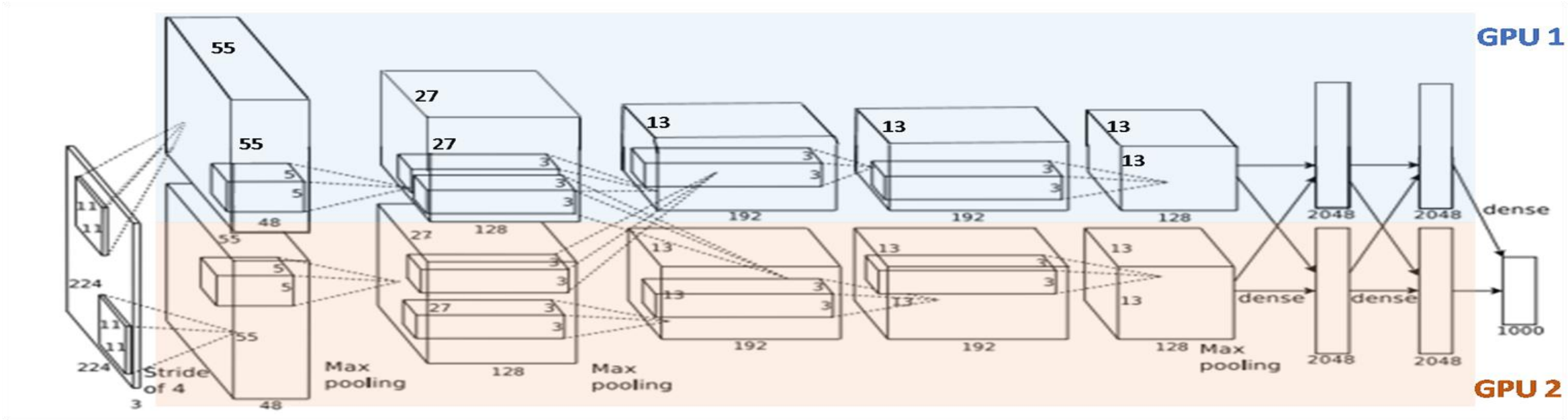
2-D Convolution

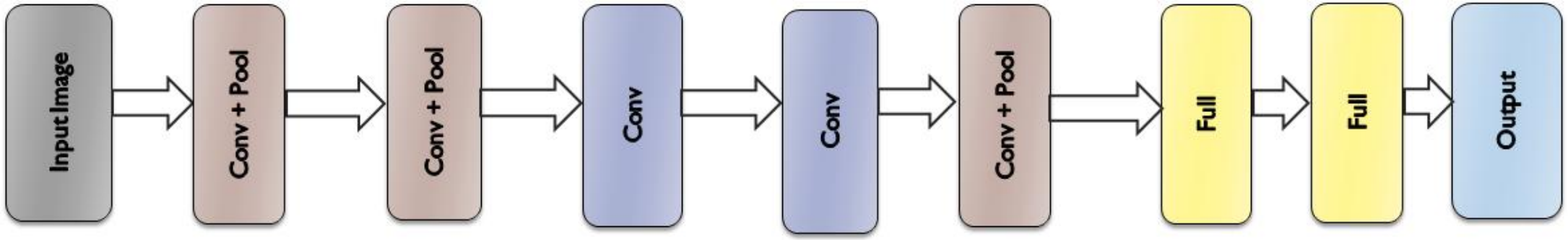
Deep Learnt Features

- It's deep if it has more than one stage of non-linear feature transformation.

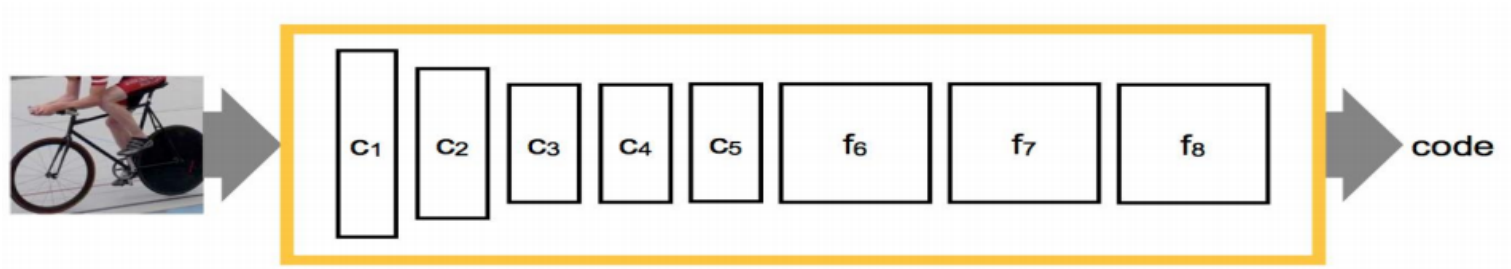


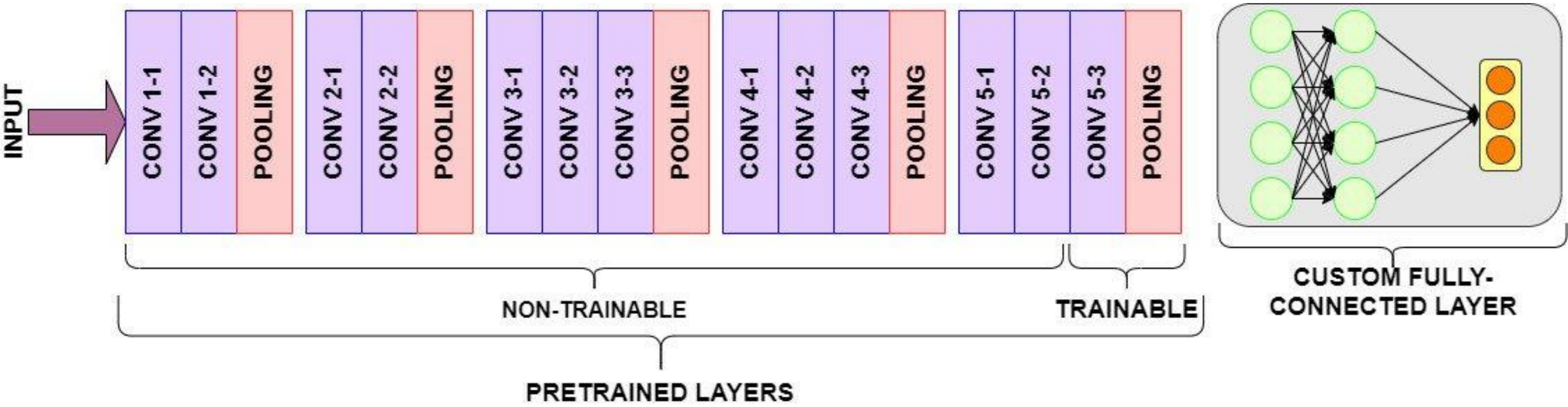
AlexNet Architecture



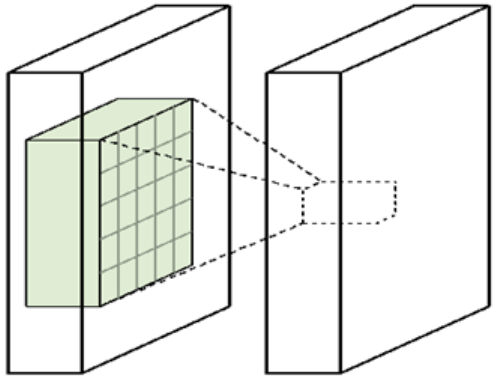


Learned Representations: Pre-Train and Fine Tune



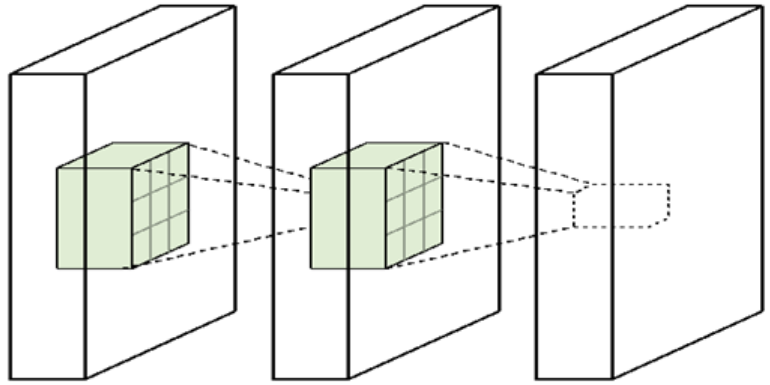


Design Guidelines: Smaller Convolutions and Deeper nw



5 × 5 filters
+ ReLU

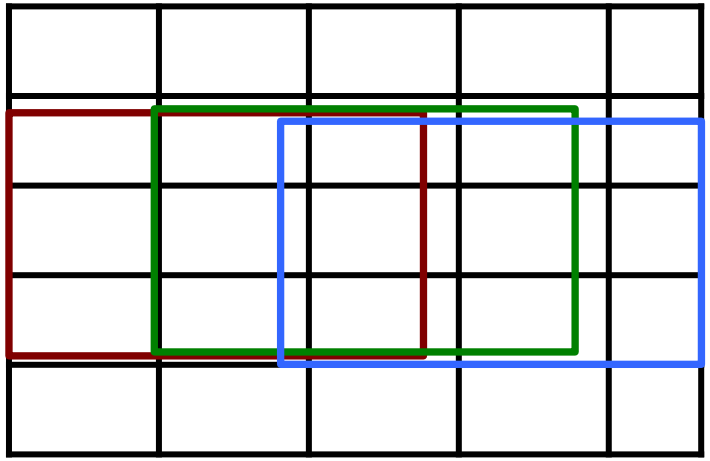
prefer



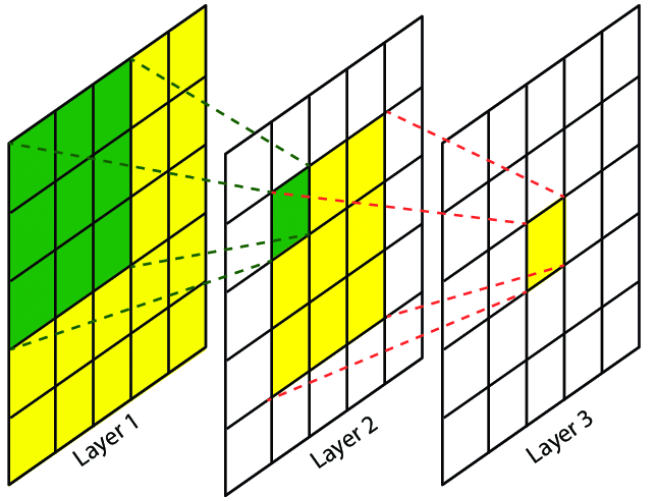
3 × 3 filters
+ ReLU

3 × 3 filters
+ ReLU

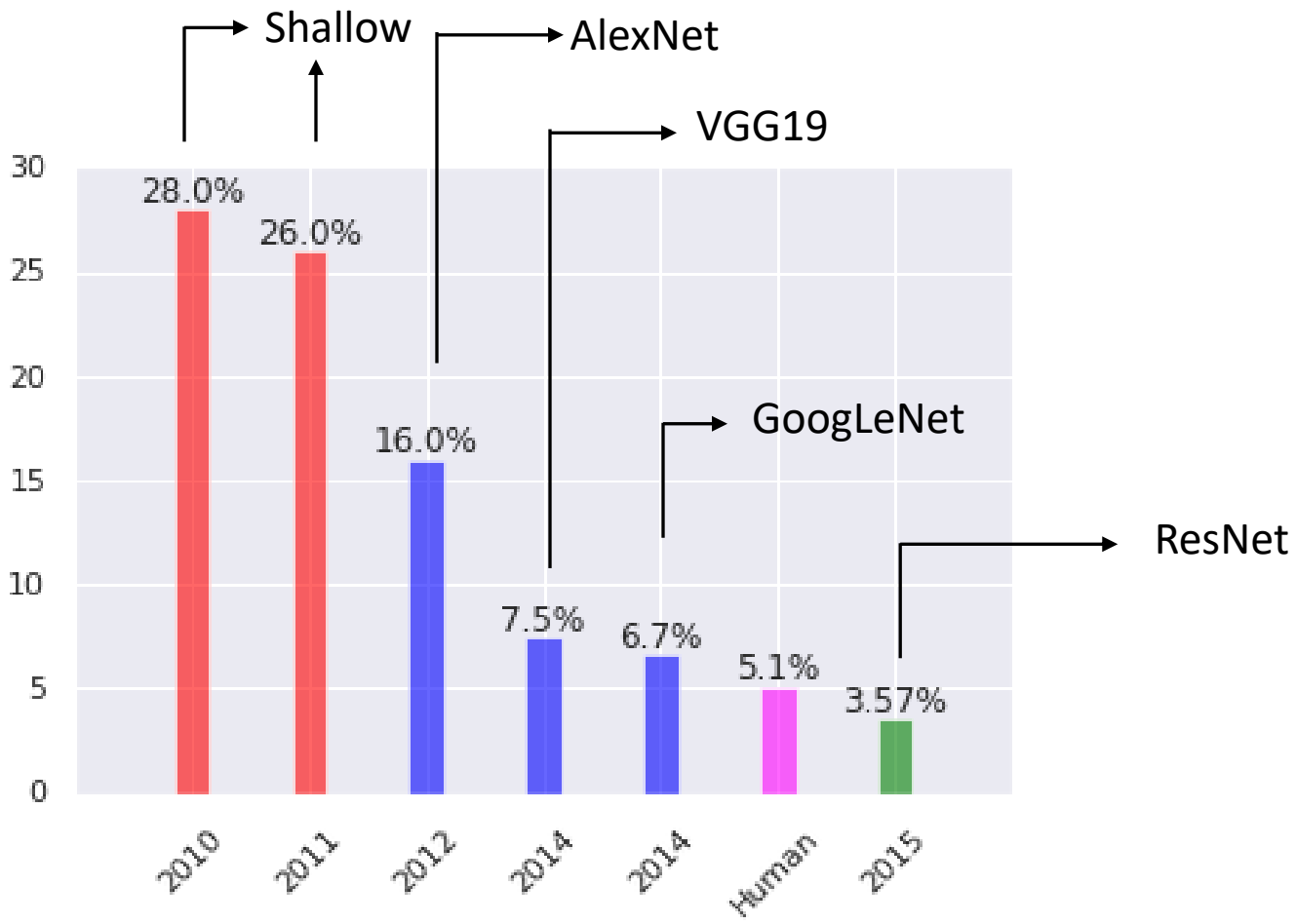
5 X 5 C > 2 X 3 X 3 C



1. Less Parameters; Faster
2. Same Receptive Field
3. More nonlinearities (2 ReLU)



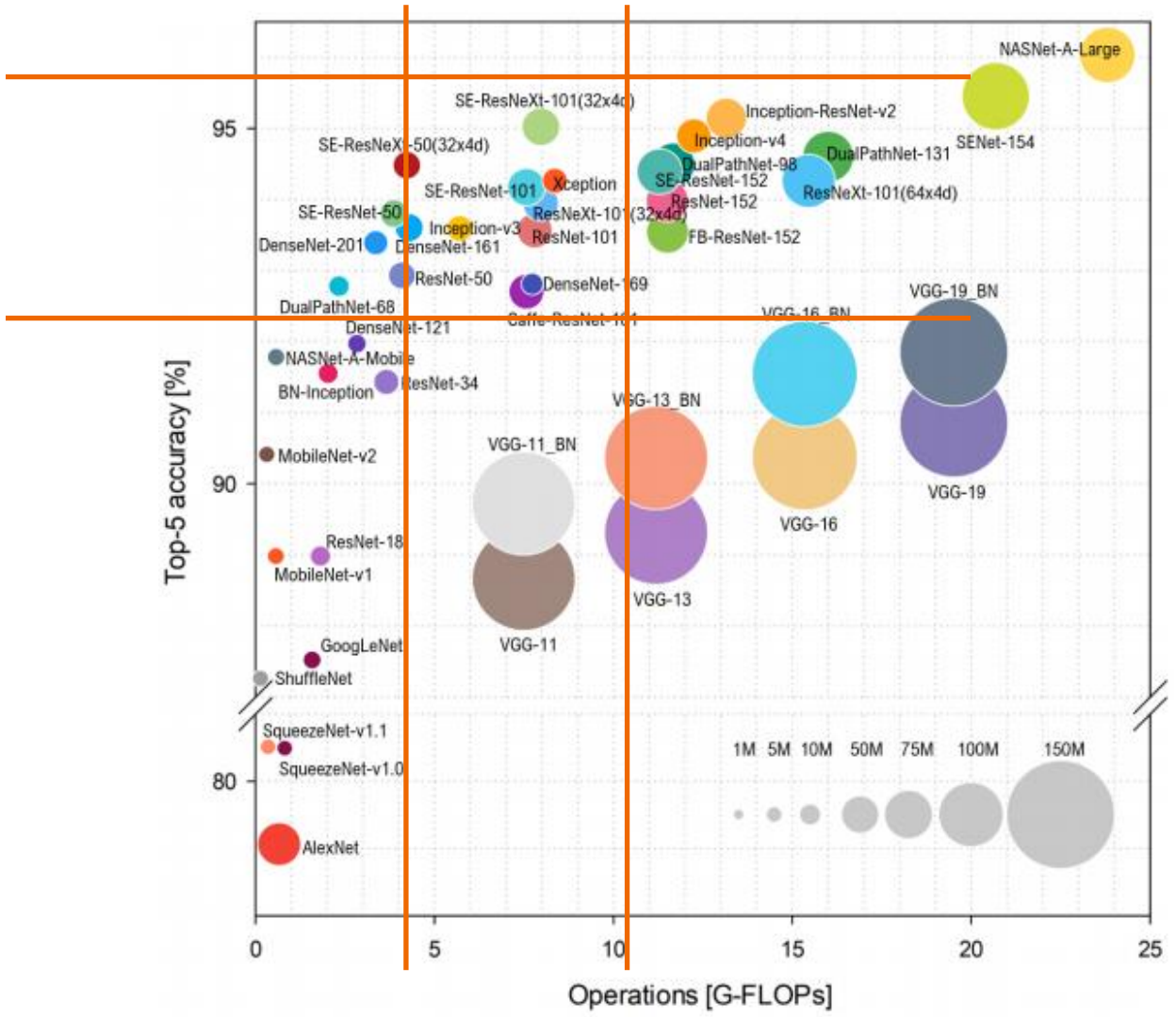
Performance over ImagenetBenchmark



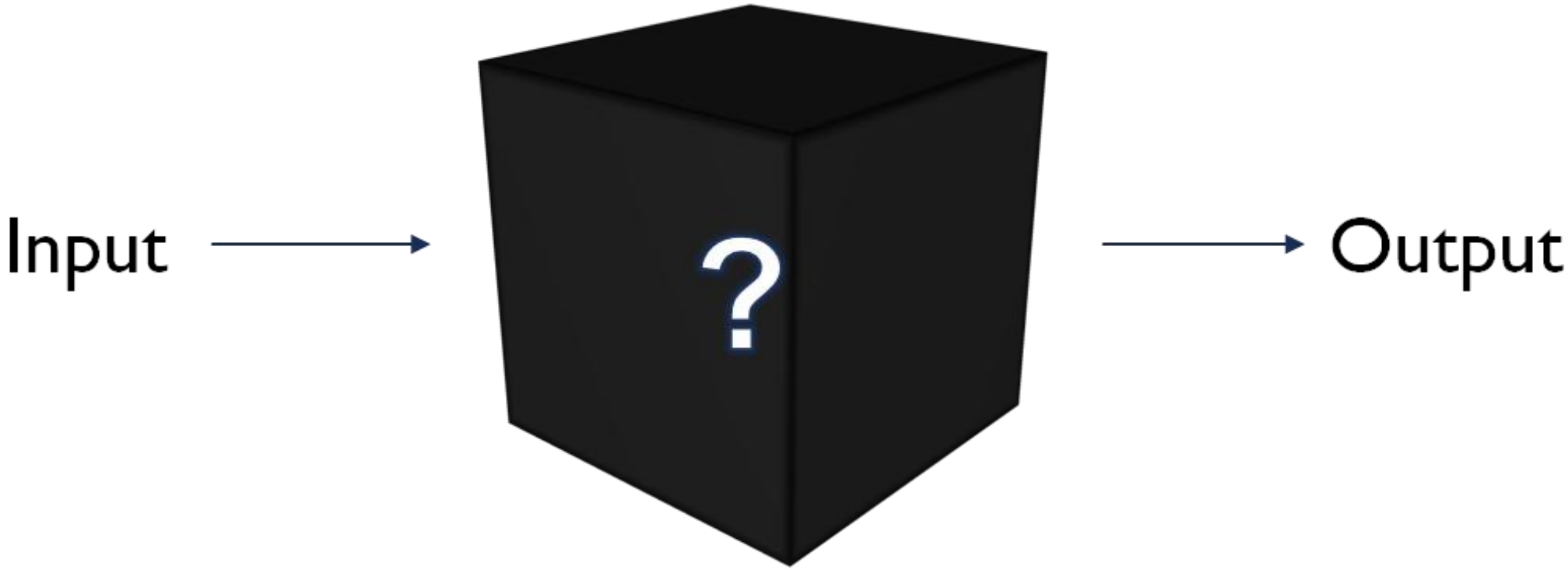
[He *et. al.*, CVPR, 2015]

Accuracy vs Model complexity Vs comput. complexity

- Size of point denotes the Model complexity
- The band around 95% accuracy has varying complexity of 4-25 G-FLOPs
- The band between 10-15 G-FLOPs have high variance in both Model Complexity (size of the point) and accuracy
- Recognition accuracy is not only dependent on the model or computational complexity

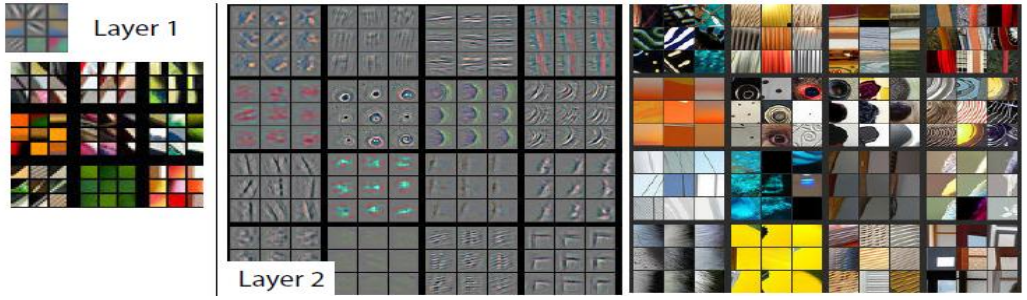


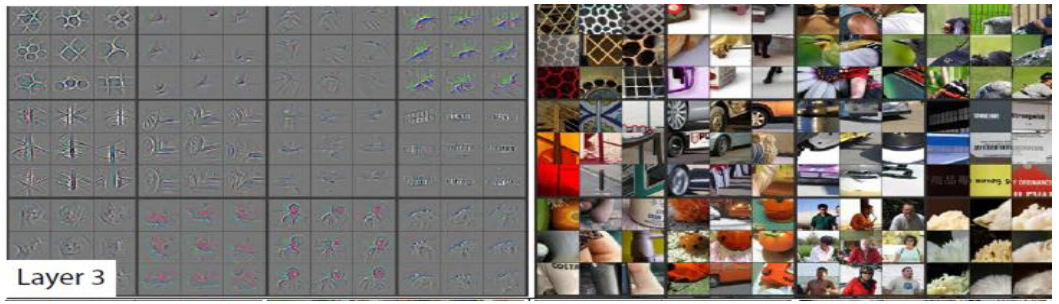
What goes on inside a convnet?

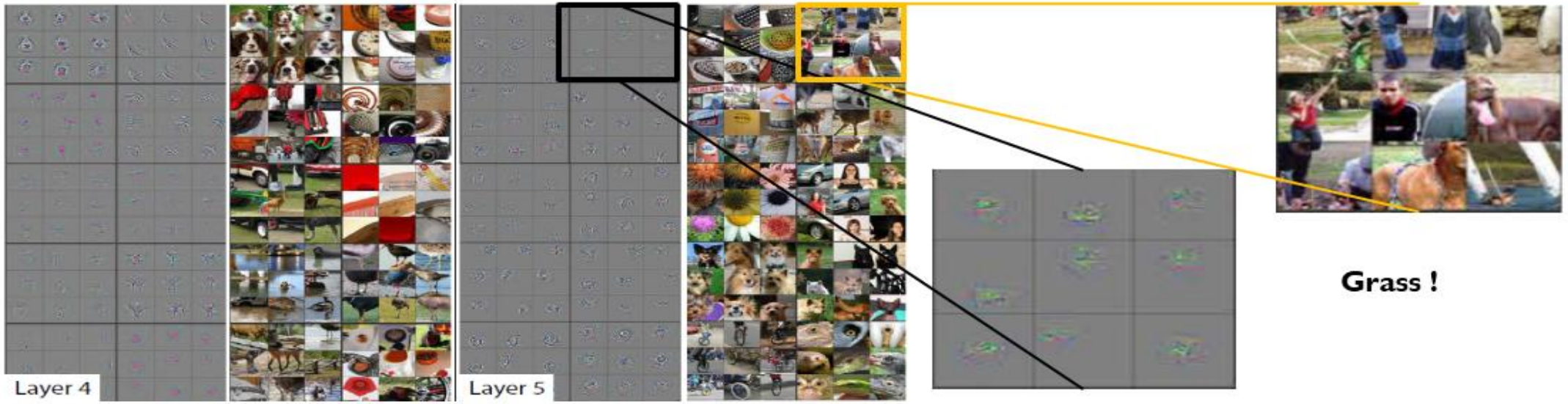


Visualizing CNNs

A. How do I interpret the learned filters?







Source: Zeiler e.t. al. ECCV'14

Zeiler and Fergus , Visualizing and Understanding Convolutional Networks,. ECCV 2014

Early Layers Converge Faster

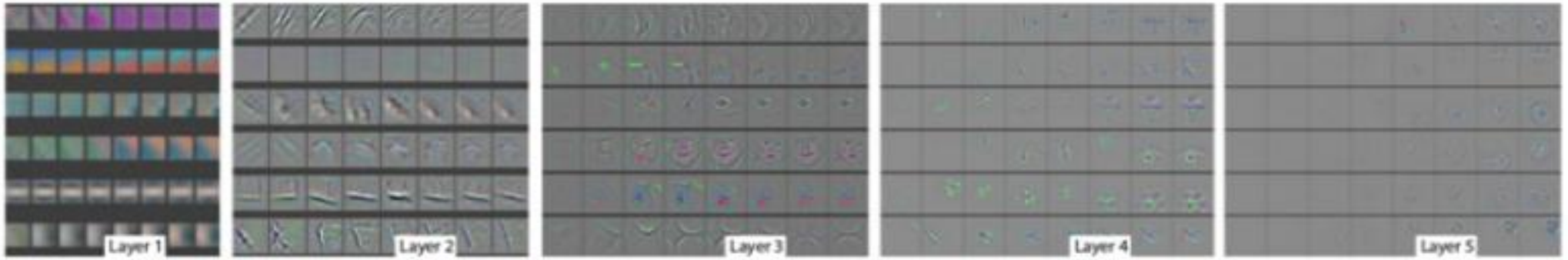
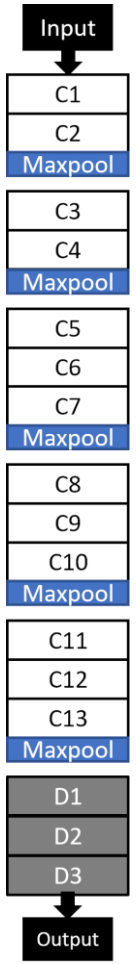


Figure: Evolution of randomly chosen subset of model features generated using deconvnet through training at epoch 1, 2, 5, 10, 20, 30, 40, 64.

CNN Visualizations



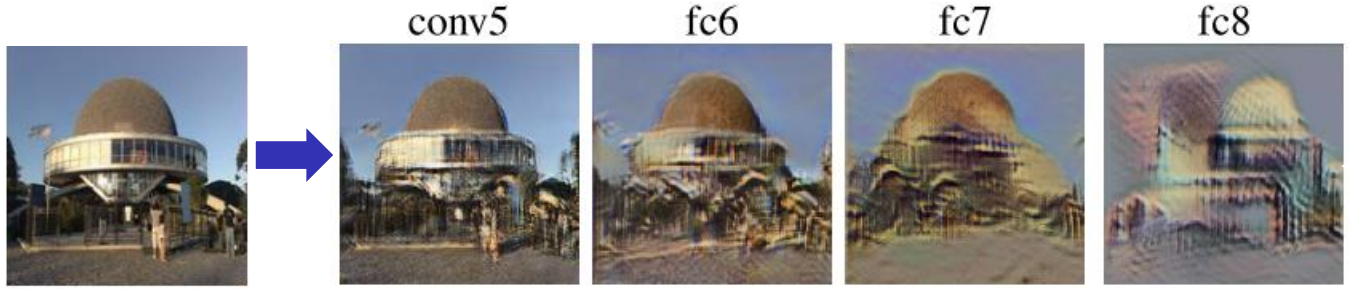
Saliency

What is in this image?

Cat Dog

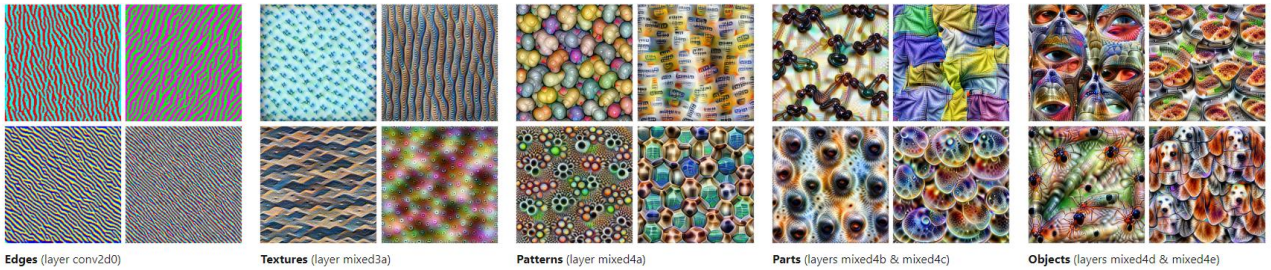
Which part of the image explains the classification?

Feature Inversion



How much information is retained/discarded at each layer of a deep network?

Feature visualization



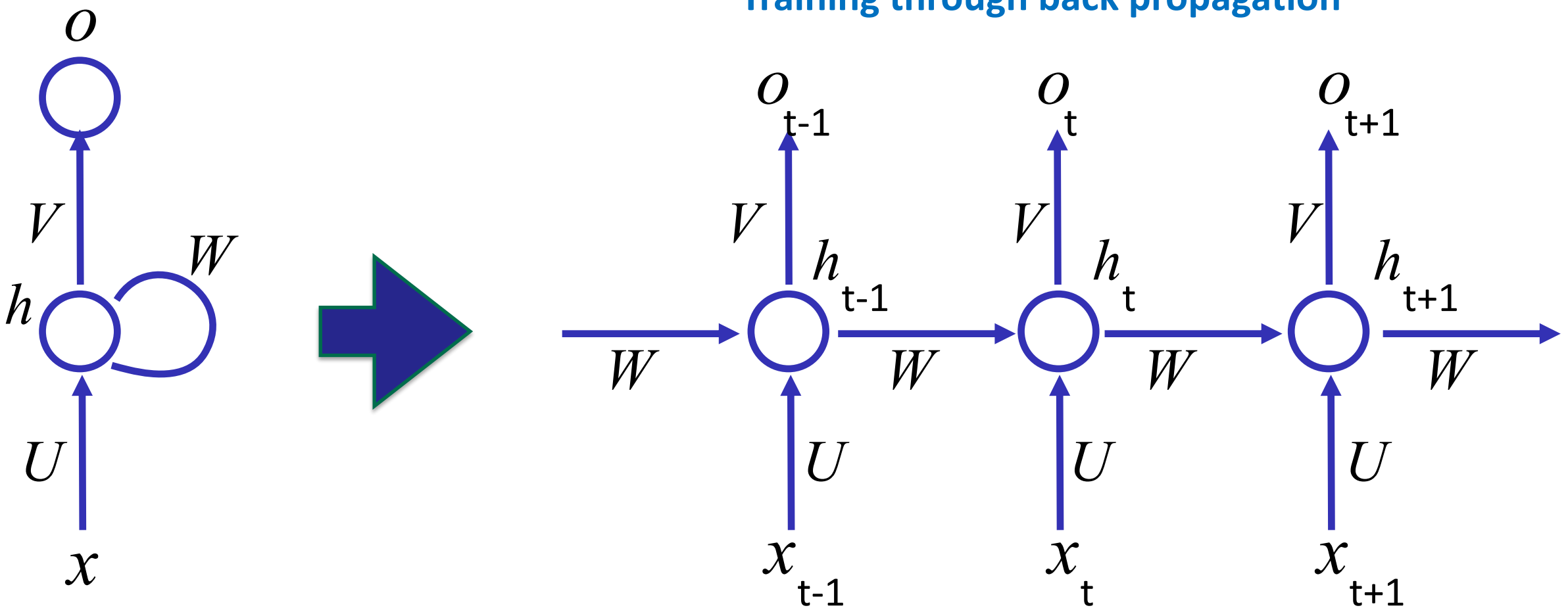
What pattern does each convolutional filter search for?

Monitoring training

How well is the CNN getting trained?

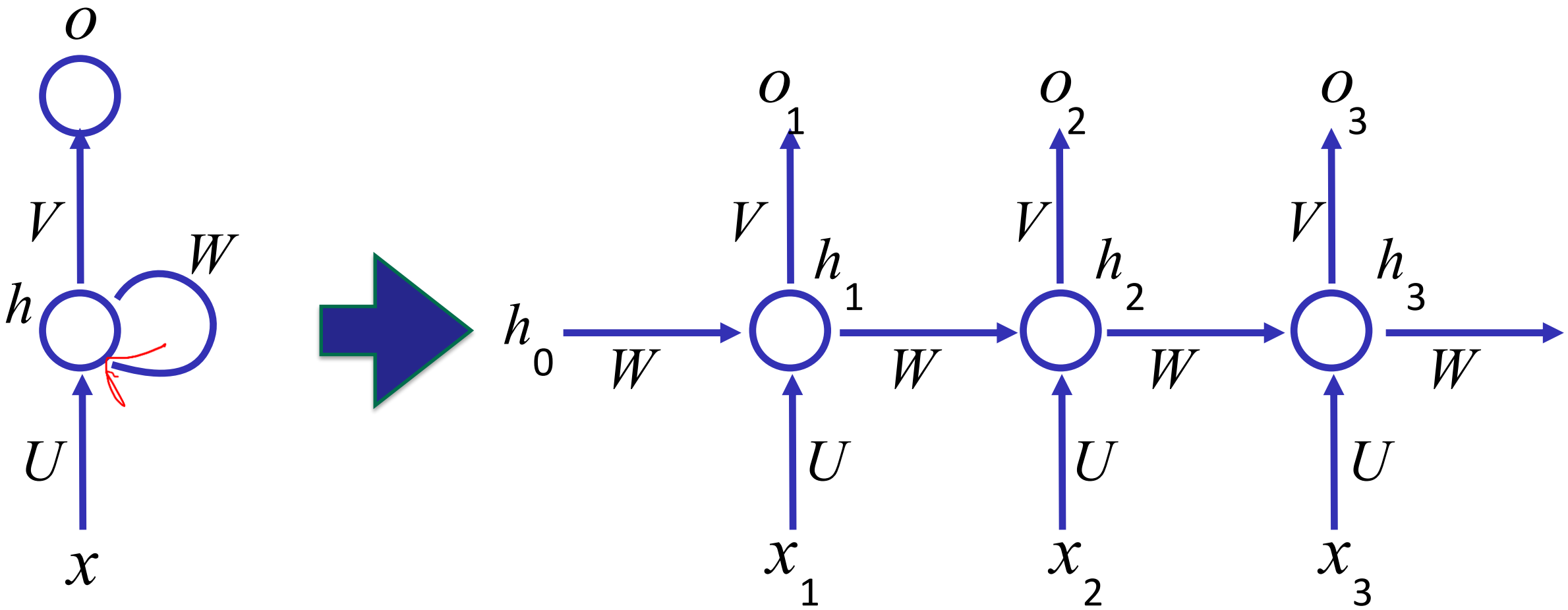
RNNs

RNN basic architecture



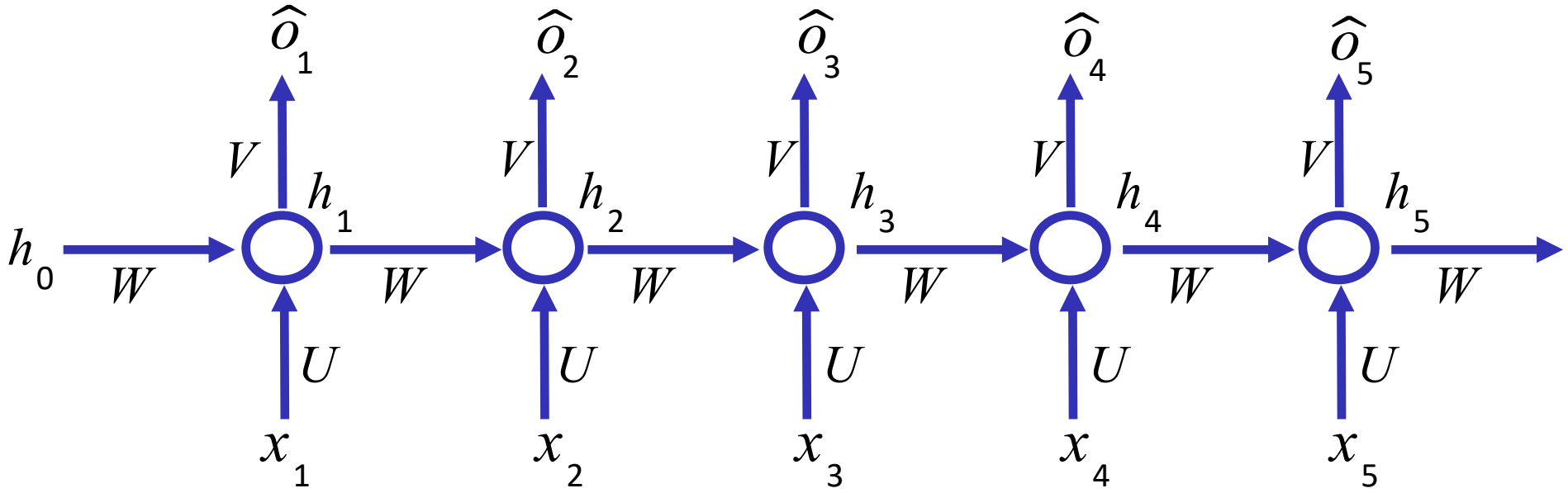
$$h_t = f(Ux_t + Wh_{t-1}) \qquad o_t = \mathrm{softmax}(Vh_t)$$

RNN basic architecture



$$h_t = f(Ux_t + Wh_{t-1}) \qquad o_t = \text{softmax}(Vh_t)$$

Forward Pass, Loss



$$h_t = f(Ux_t + Wh_{t-1})$$

$$\hat{o}_t = \text{softmax}(Vh_t)$$

$$E(o, \hat{o}) = \sum_t E_t(o_t, \hat{o}_t)$$

Questions?