CSC 578
Neural Networks and Deep Learning

6-1. Convolutional Neural Networks (1)

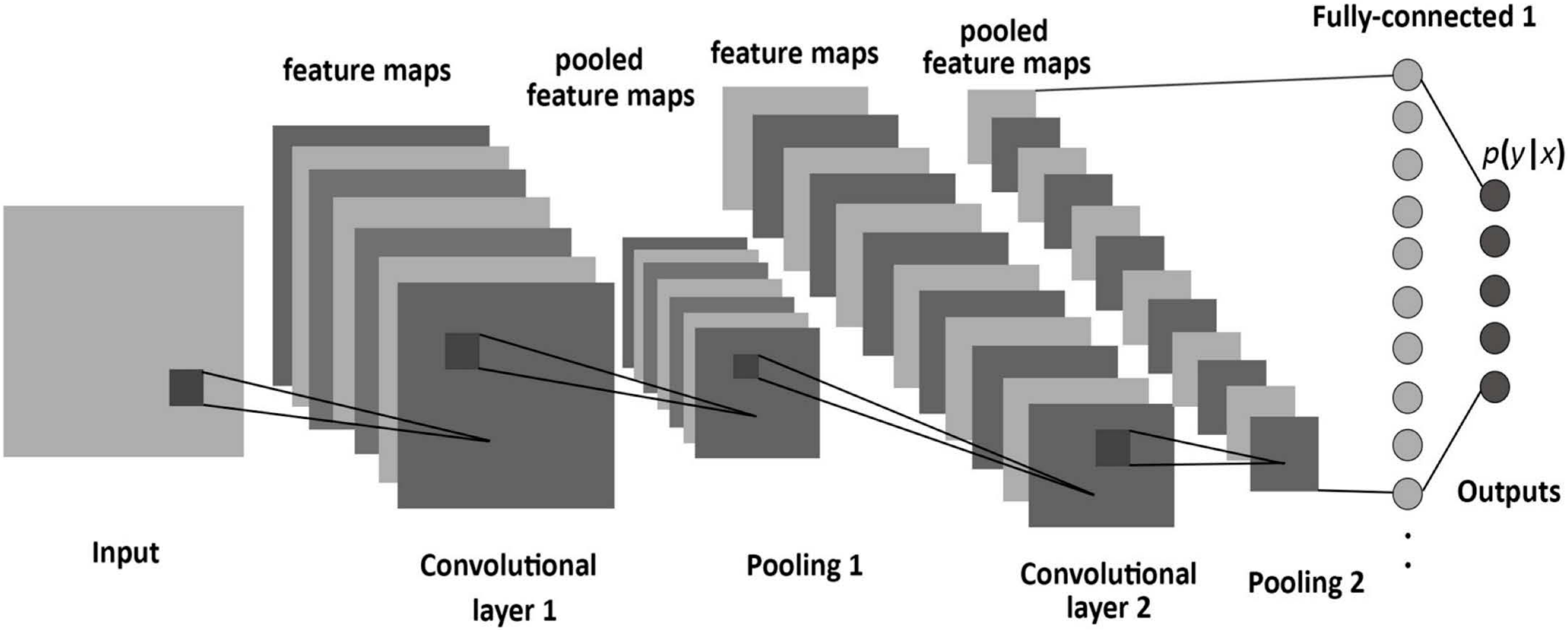
(Some figures adapted from NNDL book)

Convolution Neural Networks

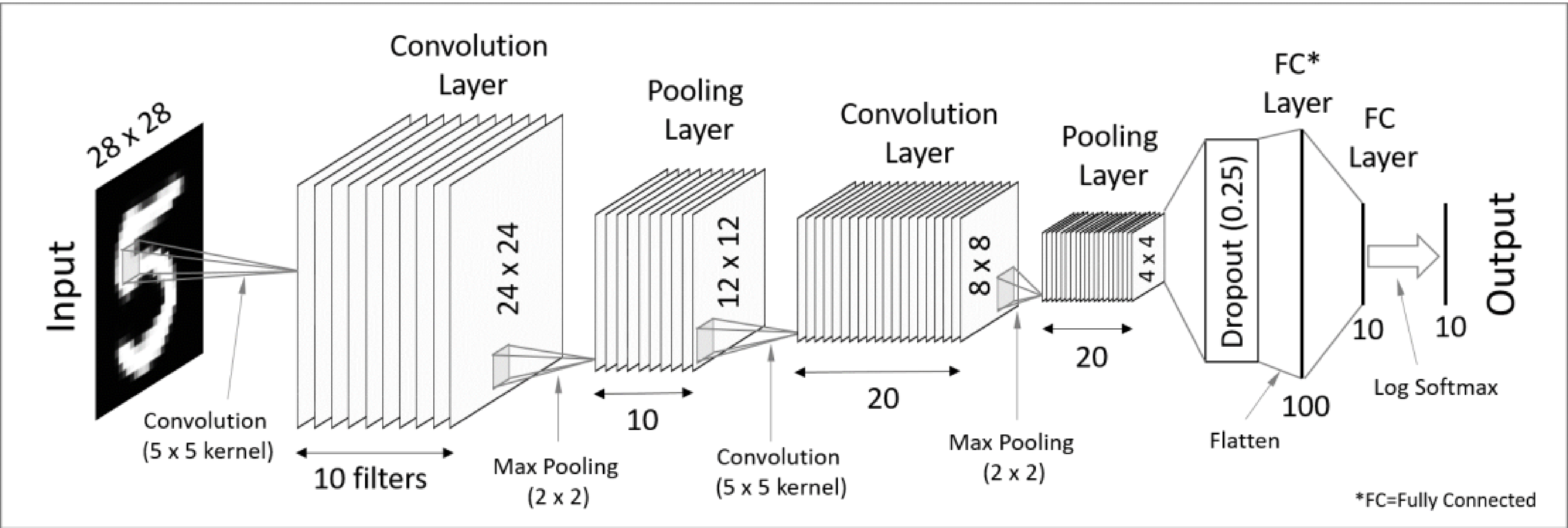
1. Convolutional Neural Networks
   - Convolution, pooling and fully-connected layers
   - Convolution kernel/filter
   - Local receptive field
2. Convolution Kernels
3. Shared Weights and Biases
   - Shift invariance
   - Learned weights
4. Pooling
   - Max, average pooling
5. CNN Learning

1 Convolutional Neural Networks

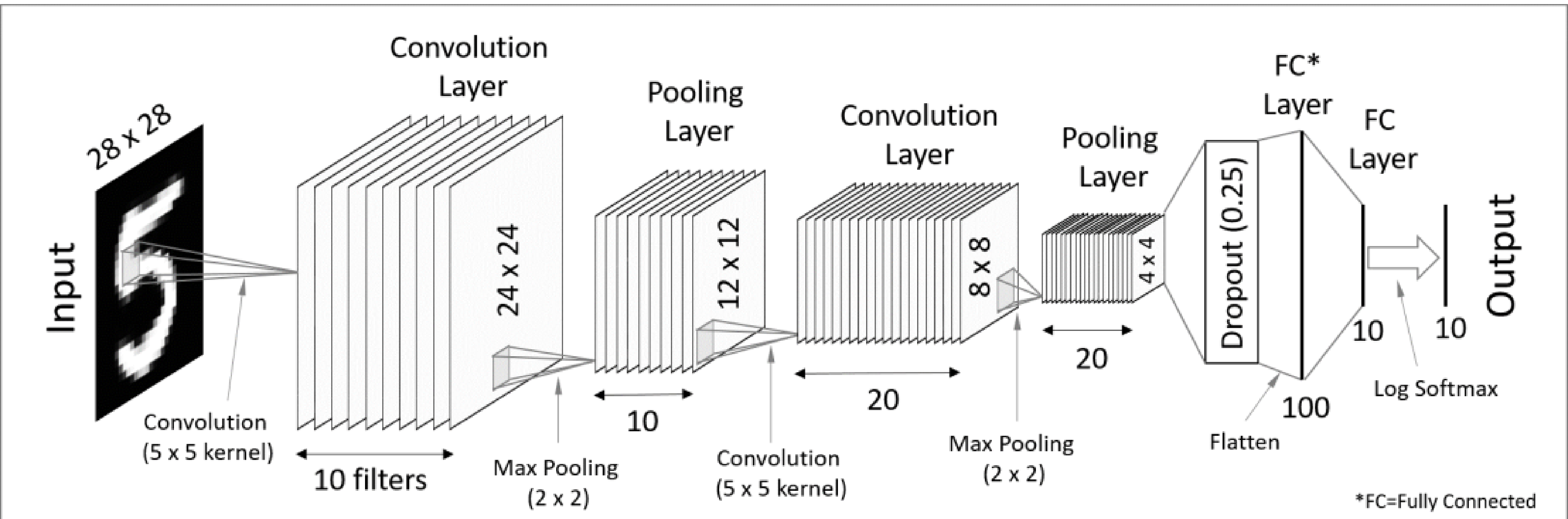
- Convolutional Neural Networks (CNNs) are a variation of a multilayer neural network, typically used for recognizing/classifying 2D images.
- A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of **convolutional layers**, **pooling layers**, and **fully connected layers.**



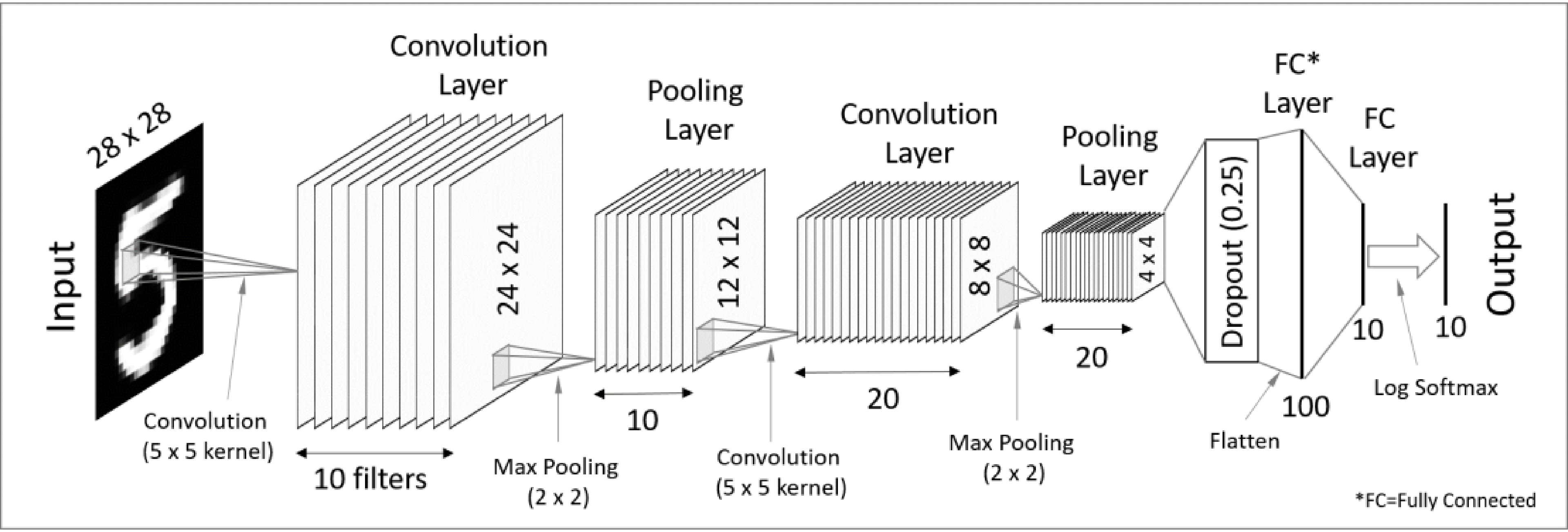
- Convolutional layers apply a [**convolution operation**]() to the input. The operation applies a **filter function/kernel** on a receptive field/window of some size over the input data.
- A receptive field is moved/slid over the input, stopping at every pixel or skipping over a fixed number of pixels (*stride*).
- You can apply as many different filter functions, where each function creates a convolution feature map.



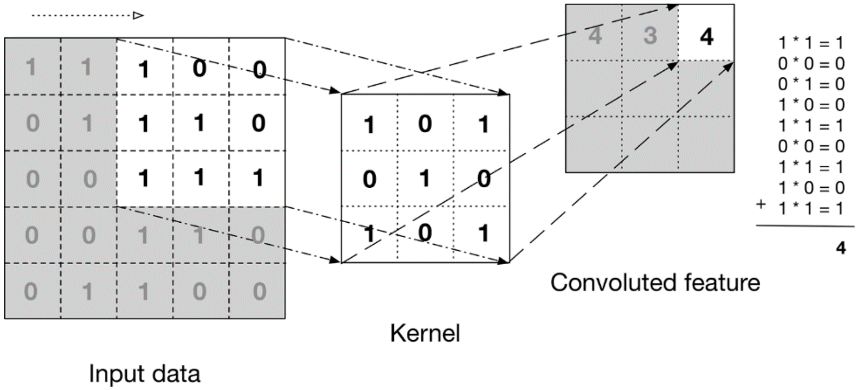
- Next, for each convolution feature map, a **pooling** operation is applied which combines a cluster of convolution features to a single value.
- Common functions are max pooling and average pooling.
- Typically a CNN consists of several convolution and pooling layers.

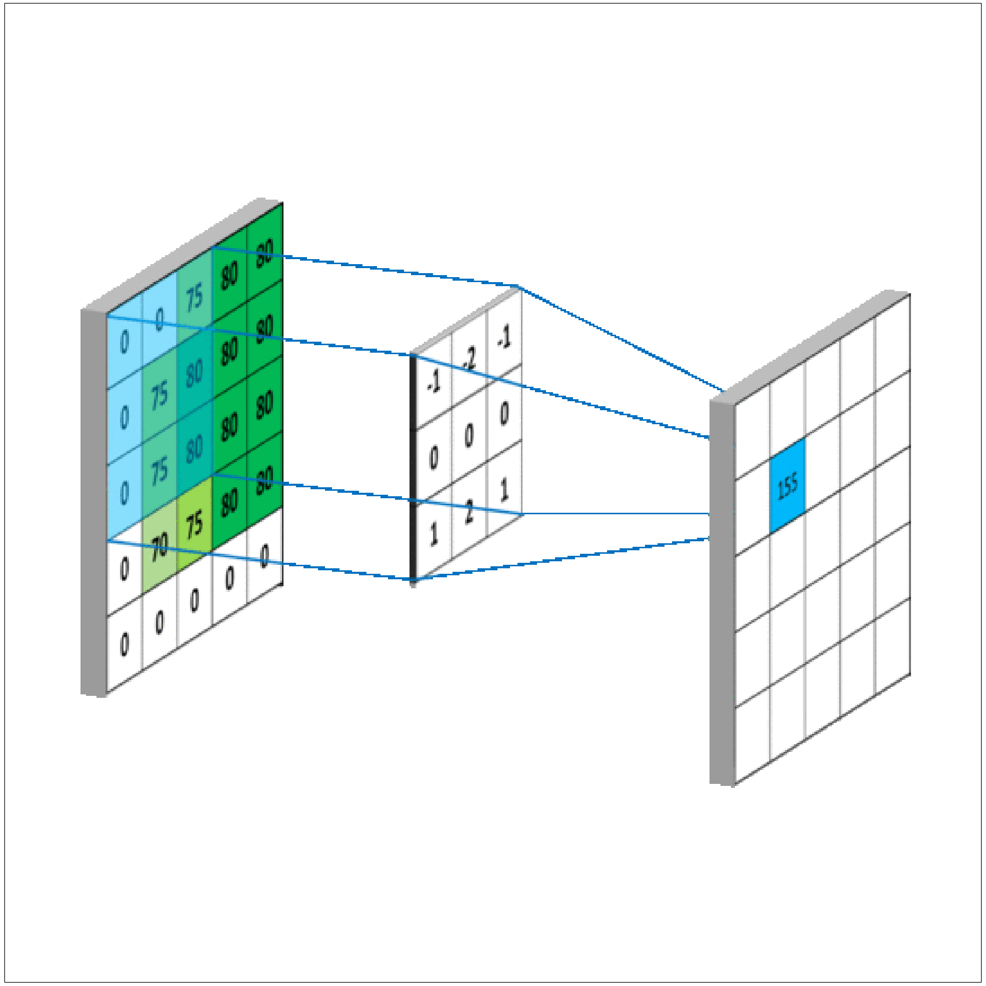


- Then the last pooling layer is flattened to a 1D vector (possibly after dropping some nodes), which gets connected a network of **fully connected layers.** It is in principle the same as the traditional multilayer neural network.
- Common functions are max pooling and average pooling.
- Typically a CNN consists of several convolution and pooling layers.



2 Convolution Kernel





0	0	0
0	1	0
0	0	0

Example Kernel

Original

Emboss

1	1	1
1	1	1
1	1	1

Unweighted 3x3 smoothing kernel

0	1	0
1	4	1
0	1	0

Weighted 3x3 smoothing kernel with Gaussian blur

0	-1	0
-1	5	-1
0	-1	0

Kernel to make image sharper

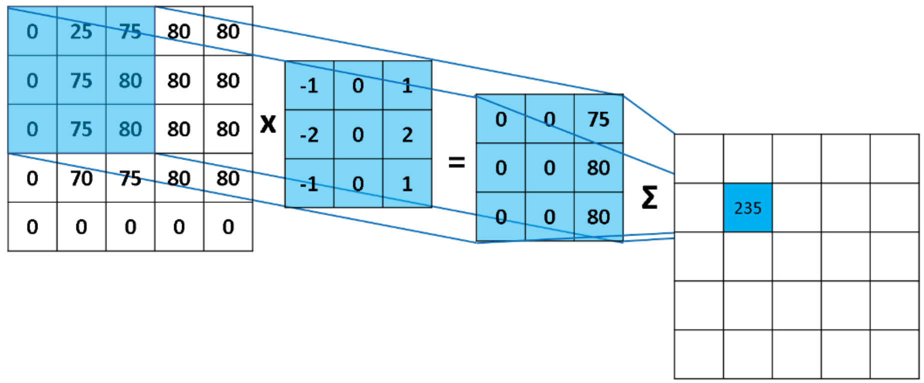
-1	-1	-1
-1	9	-1
-1	-1	-1

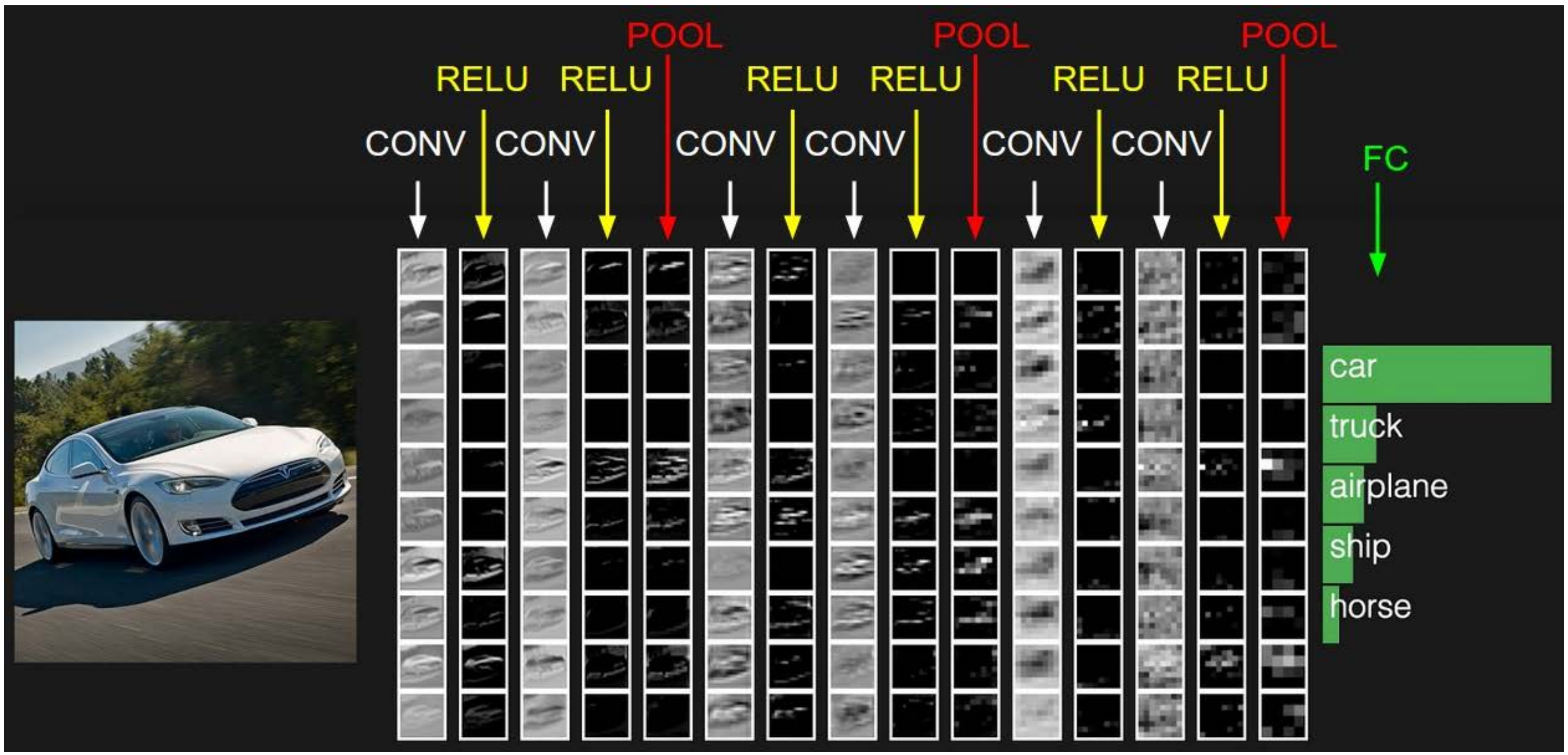
Intensified sharper image

Gaussian Blur

Sharpened image

- Sometimes an activation function (applied after kernel) is considered a separate layer.

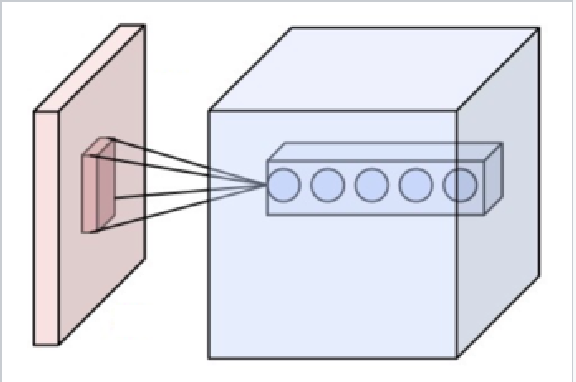




3 Shared Weights and Biases

- Nodes in a receptive field to a node on the convolution layer are connected with weights. There is also a bias. Those weights and bias are shared – **same values are used for a given filter** as a receptive field is moved on the same (input or intermediate convolutional) layer.
- For example, the output of the *jk*th convolution node from a 5x5 receptive field would be

$$\sigma\left(b + \sum_{l=0}^{4}\sum_{m=0}^{4} w_{l,m} a_{j+l,k+m}\right)$$

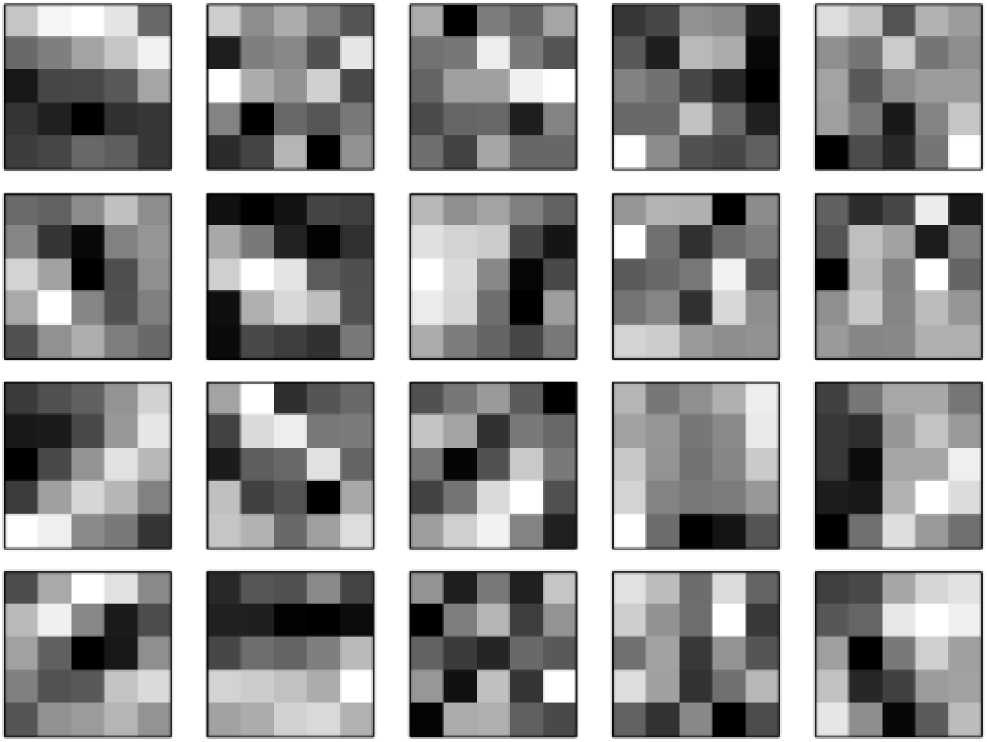


Neurons of a convolutional layer (blue), connected to their receptive field (red)

And those weights are **learned** by training.

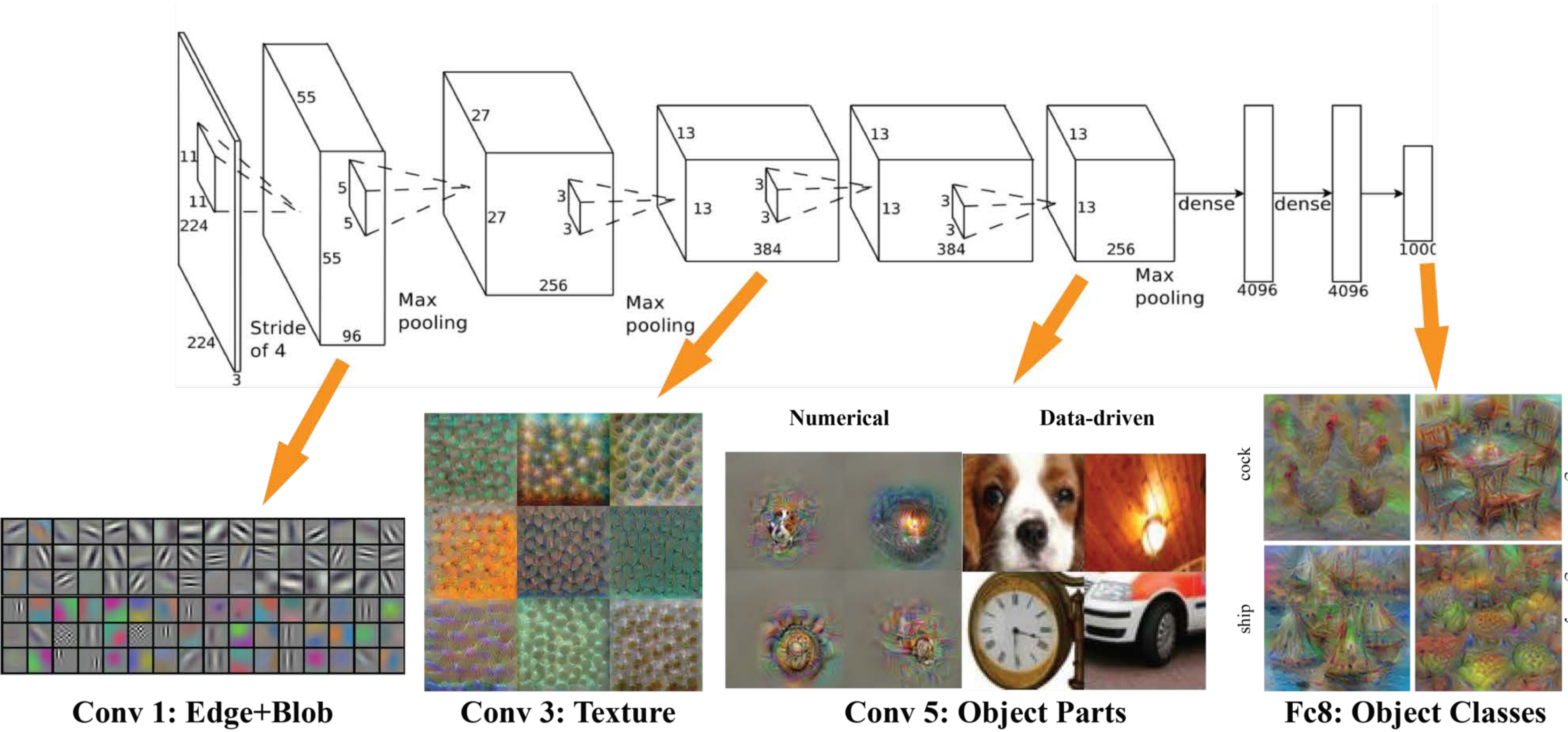
- By sharing the same weights (and bias) for the same filter, all the neurons in a convolution layer detect **exactly the same feature** in the preceding layer (input or intermediate pooled layer), just at different locations.
- This makes a filter "***shift invariant"*** – being able to find the feature anywhere in the entire image (or the preceding layer) wherever it occurred.
- For example, filters could detect edges, lines, corners and blobs of color.
- **Animation** of sliding window of receptive field, max pool etc.

- MNIST example (from the NNDL book):



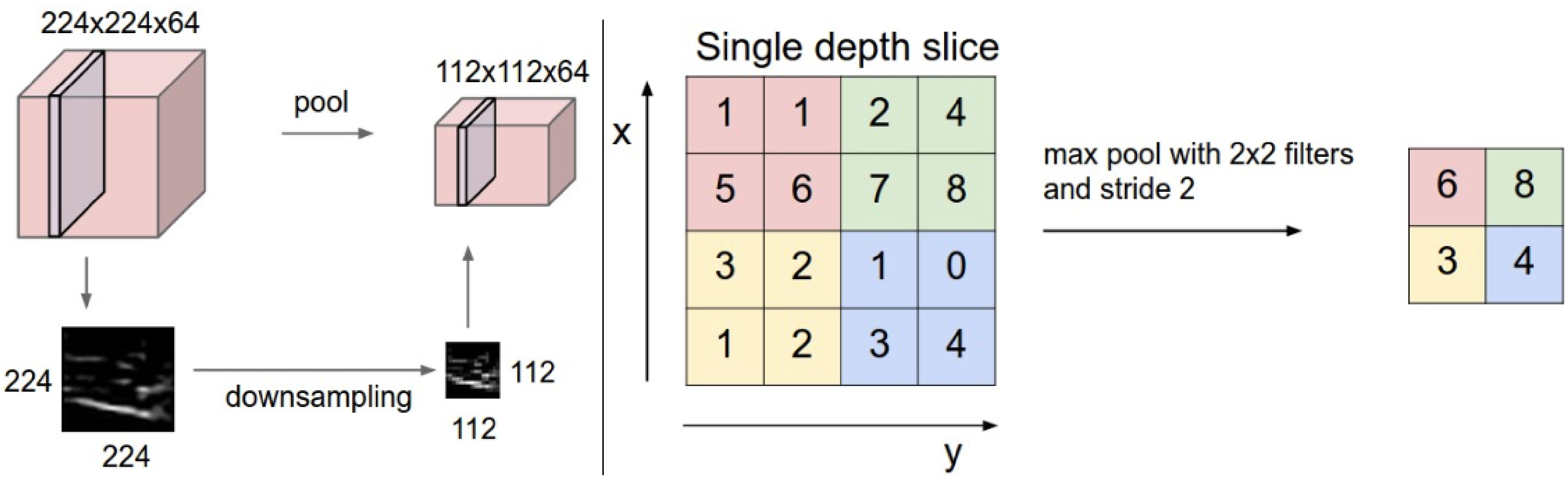
The 20 images correspond to 20 different feature maps (or filters, or kernels). Each map is represented as a 5×5 block image, corresponding to the 5×5 weights in the local receptive field.

- Color images typically have 3 channels (RGB), thus the input images have the depth of 3.
- Example: [AlexNet]



4 Pooling

- A pooling layer takes each feature map output from the convolutional layer and prepares a condensed feature map.



Pooling layer downsamples the volume spatially, independently in each depth slice of the input volume. **Left:** In this example, the input volume of size [224x224x64] is pooled with filter size 2, stride 2 into output volume of size [112x112x64]. Notice that the volume depth is preserved. **Right:** The most common downsampling operation is max, giving rise to **max pooling**, here shown with a stride of 2. That is, each max is taken over 4 numbers (little 2x2 square).

4 CNN Learning

- CNNs are a variation of feed-forward deep neural network. So all of the concepts we learned in the previous sections apply, in particular:
  1. Neuron Activation functions
  2. Cost/loss functions
  3. Cost/loss minimization
  4. Other hyper-parameters
- CNN learning is to learn the weights between layers.
- But the weights between a hidden/convolution layer and its preceding layer are a kernel (modulo activation function). So essentially this learning is to learn convolution kernels.

- Example of learned kernels