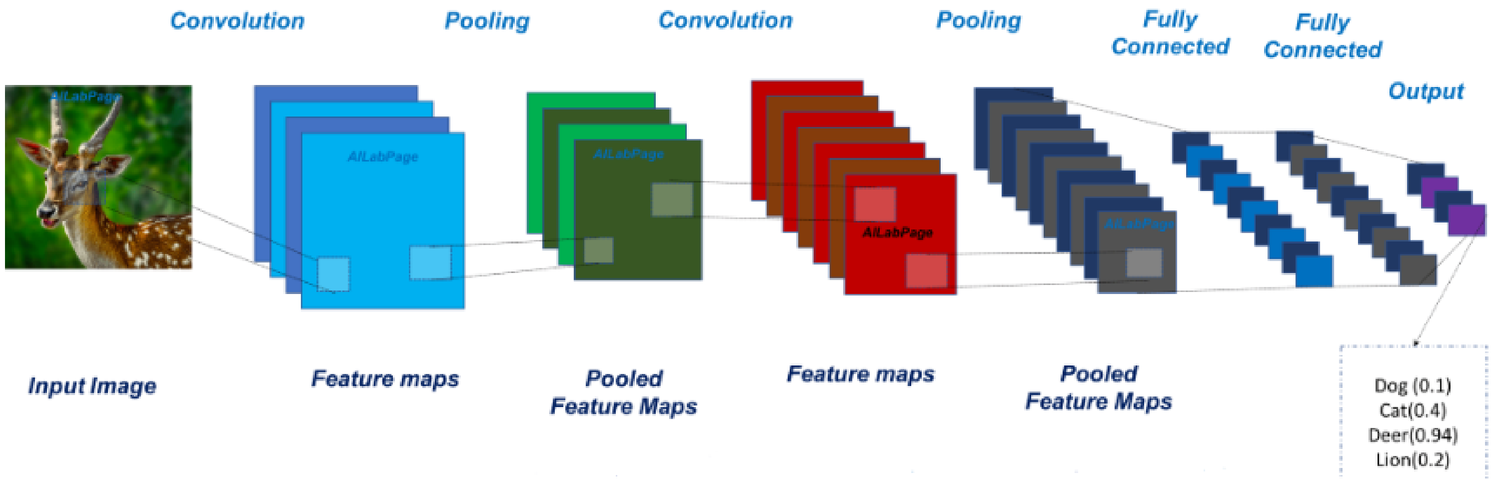
Convolutional Neural Network

Kim Gwangho, Seo Youjung, Lee Hyunbin

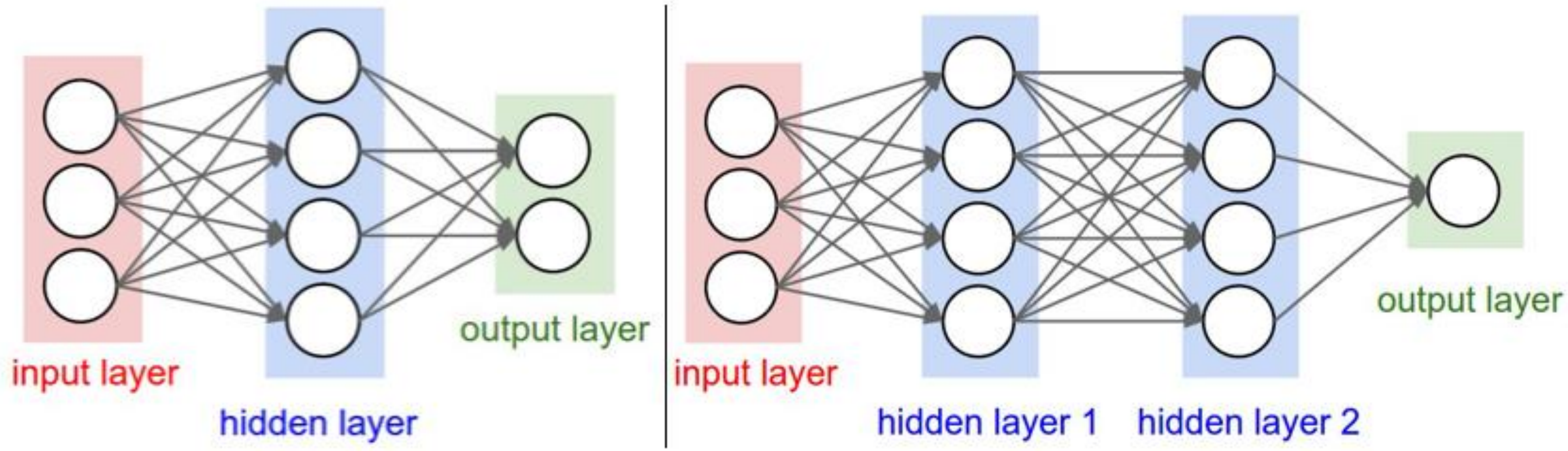
Table of Contents



- What is Convolutional Neural Network?(1-5)
- Feature Map(Filter, Stride, Padding, Output Size)
- 중간정리
- Pooling(Max, Average Pooling)
- ReLU, Training(경사하강법, Adam, SGD)
- Structure Diagram
- Code(Keras)

Source: mcai
https://mc.ai

Regular Neural Network(Fully Connected Layer)

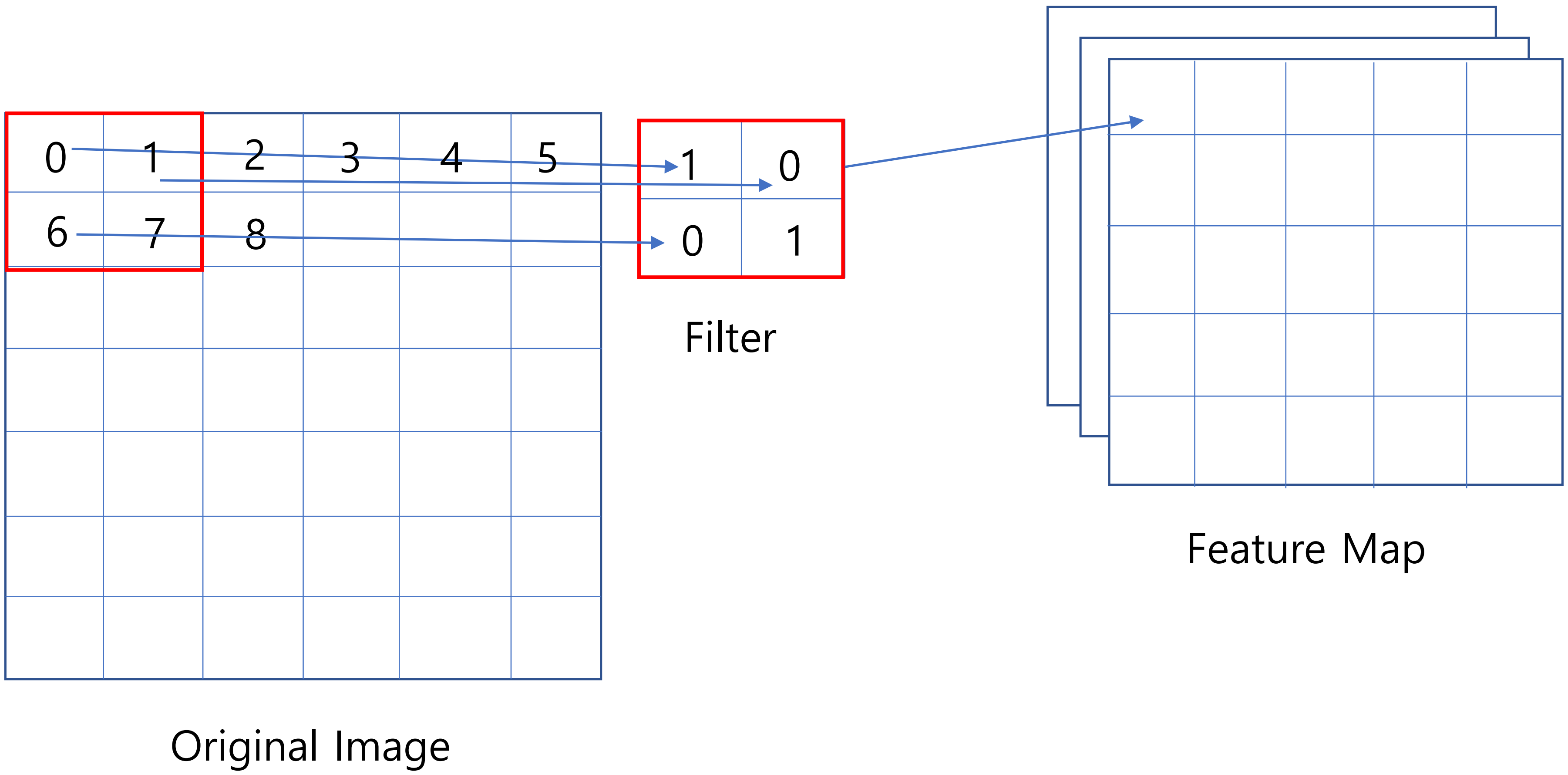


Left: A 2-layer Neural Network (one hidden layer of 4 neurons (or units) and one output layer with 2 neurons), and three inputs. **Right:** A 3-layer neural network with three inputs, two hidden layers of 4 neurons each and one output layer. Notice that in both cases there are connections (synapses) between neurons across layers, but not within a layer.

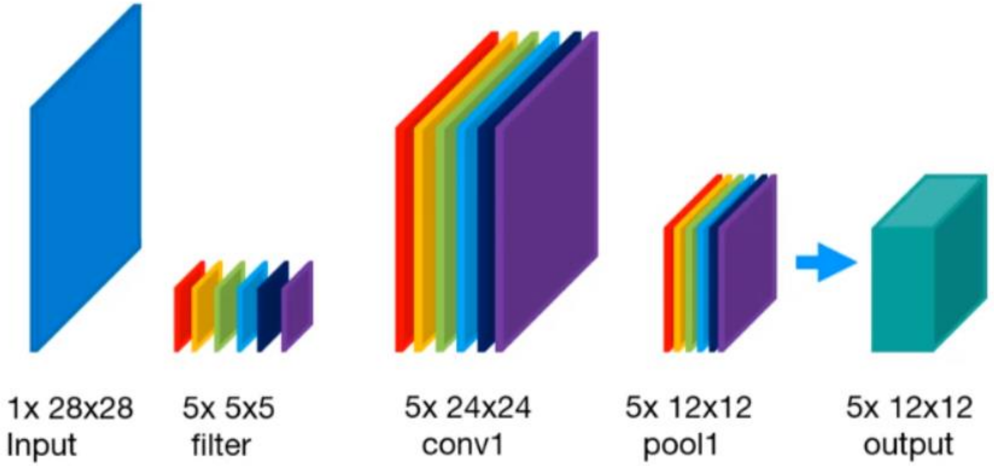
1. Connected with all of the adjacent neurons
2. Ignore spatial Image
3. Not applicable to colored image

Source: CS231n
http://cs231n.github.io/convolutional-networks/

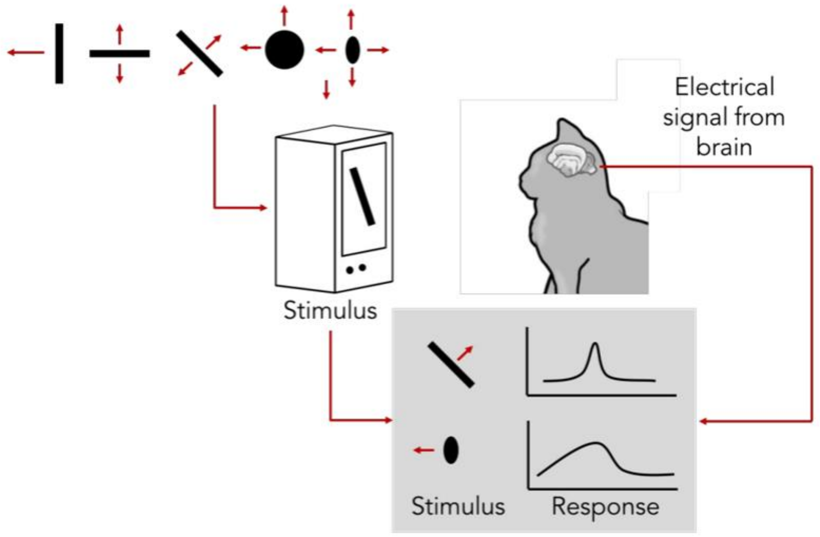
Convolutional Neural Network



What is CNN?



- ✓ Image is cut down to filters to be made as convolutional layers.
- ✓ Each output is added altogether to make up a output result.

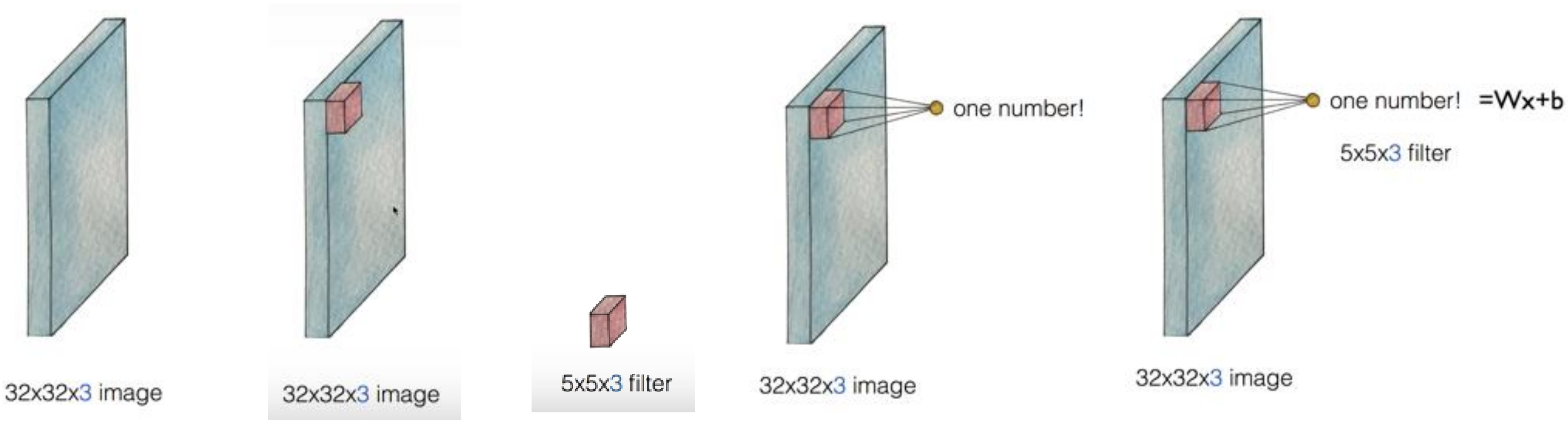


Hubel & Wiesel Experiment

- ✓ Different cells respond to different stimulus of an image

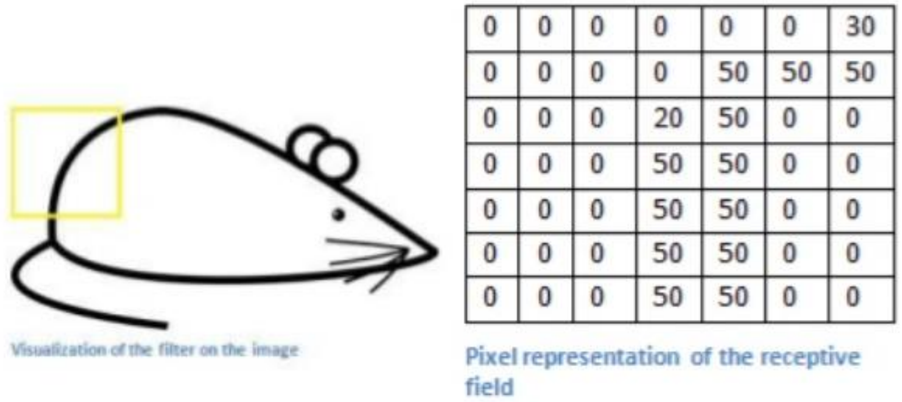
i.e: cells that respond to edges, oriented edges, shapes.

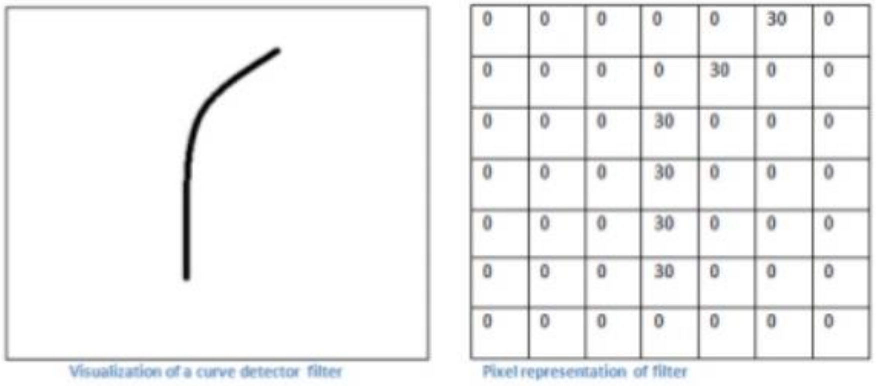
Filter



- ✓ One variable from one filter is made that represent a part of an image.
- ✓ In Wx+b, W indicate the filter 5x5x3 filter(using activation function)
- ✓ Filter value = Wx1 + Wx2 + Wx3 + ··· + WxN + b + (ReLU Wx + b)
- ✓ Filter value stays fixed throughout the entire process.

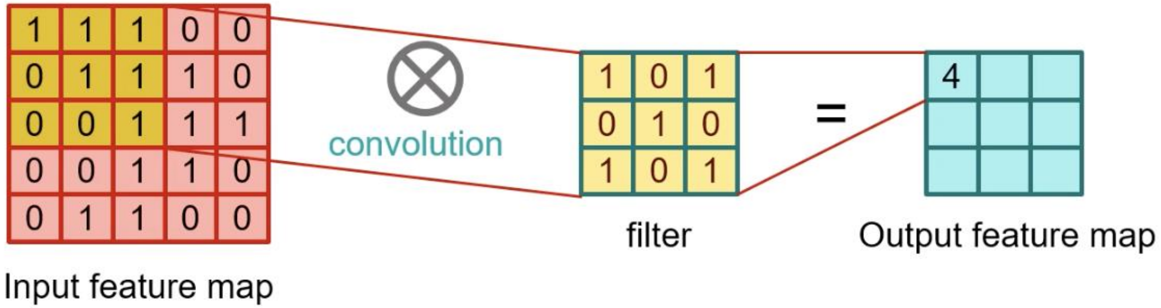
Filter





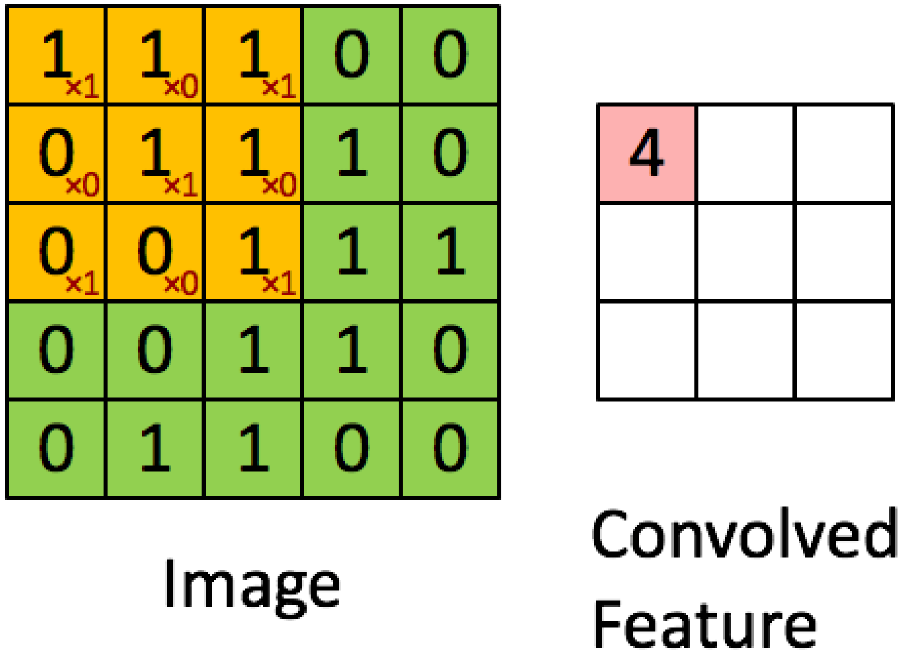
Feature Map

$$1\times1 + 1\times0 + 1\times1 + 0\times0 + 1\times1 + 1\times0 + 0\times1 + 0\times0 + 1\times1 = 4$$



- ✓ Filter is images which has the contents we'd like to detect.
- ✓ Size of filter is determined by how much pixel we'd like to detect at once.

What happens when filter slides through the image?



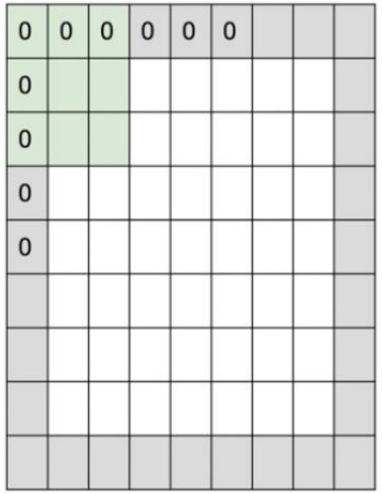
- Key idea is to observe parts of the image rather than the whole.
- Feature map indicates distinctive feature of image (ex.sharpness, blue, edges)

✓ Stride = a value of how much we would like the filter to move(이동크기)

Filter = 3 x 3, Stride = 1

Filter = 3 x 3, Stride = 2

Padding

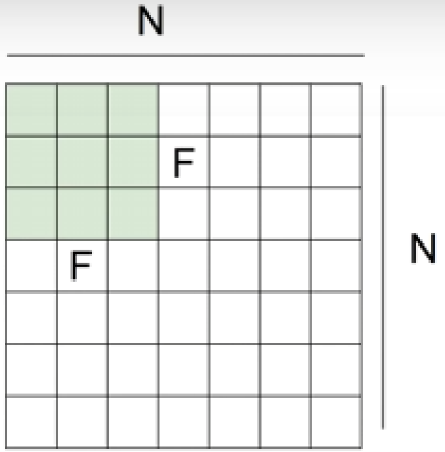


e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

7x7 output!
in general, common to see CONV layers with stride 1, filters of size FxF, and zero-padding with (F-1)/2. (will preserve size spatially)
e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

- ✓ To try to preserve as much output as possible.
- ✓ To identify the edges of the image

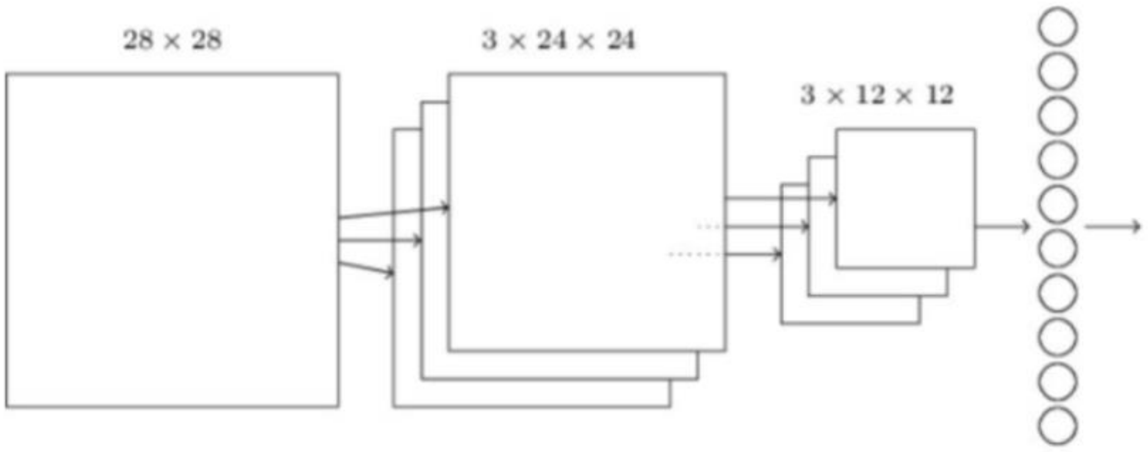
Output Size

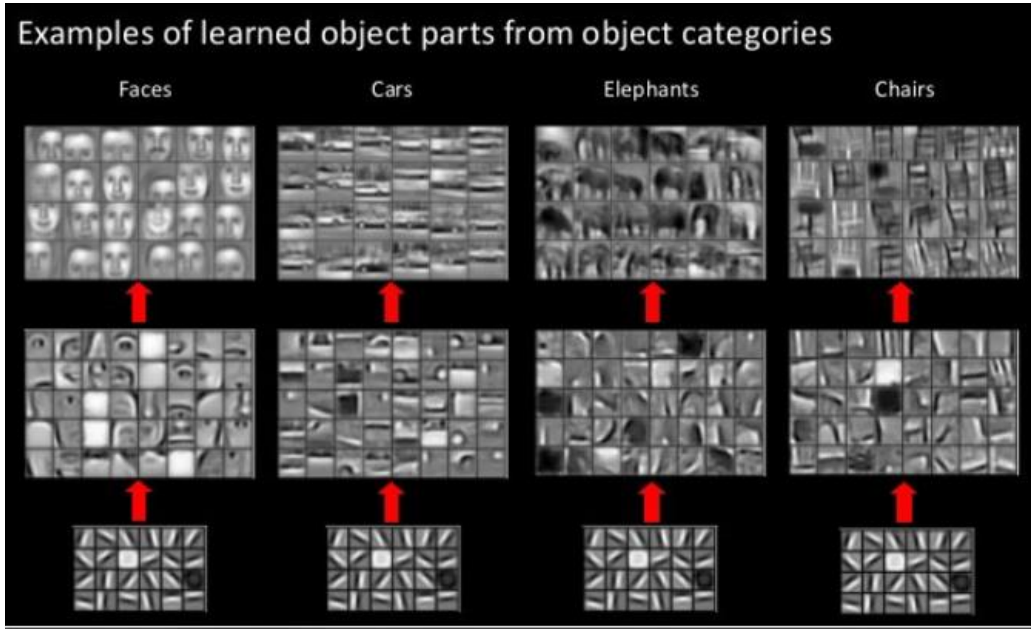


Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33 :\

7 x 7 input spatially
Assume 3 x 3 filter
= 5 x 5

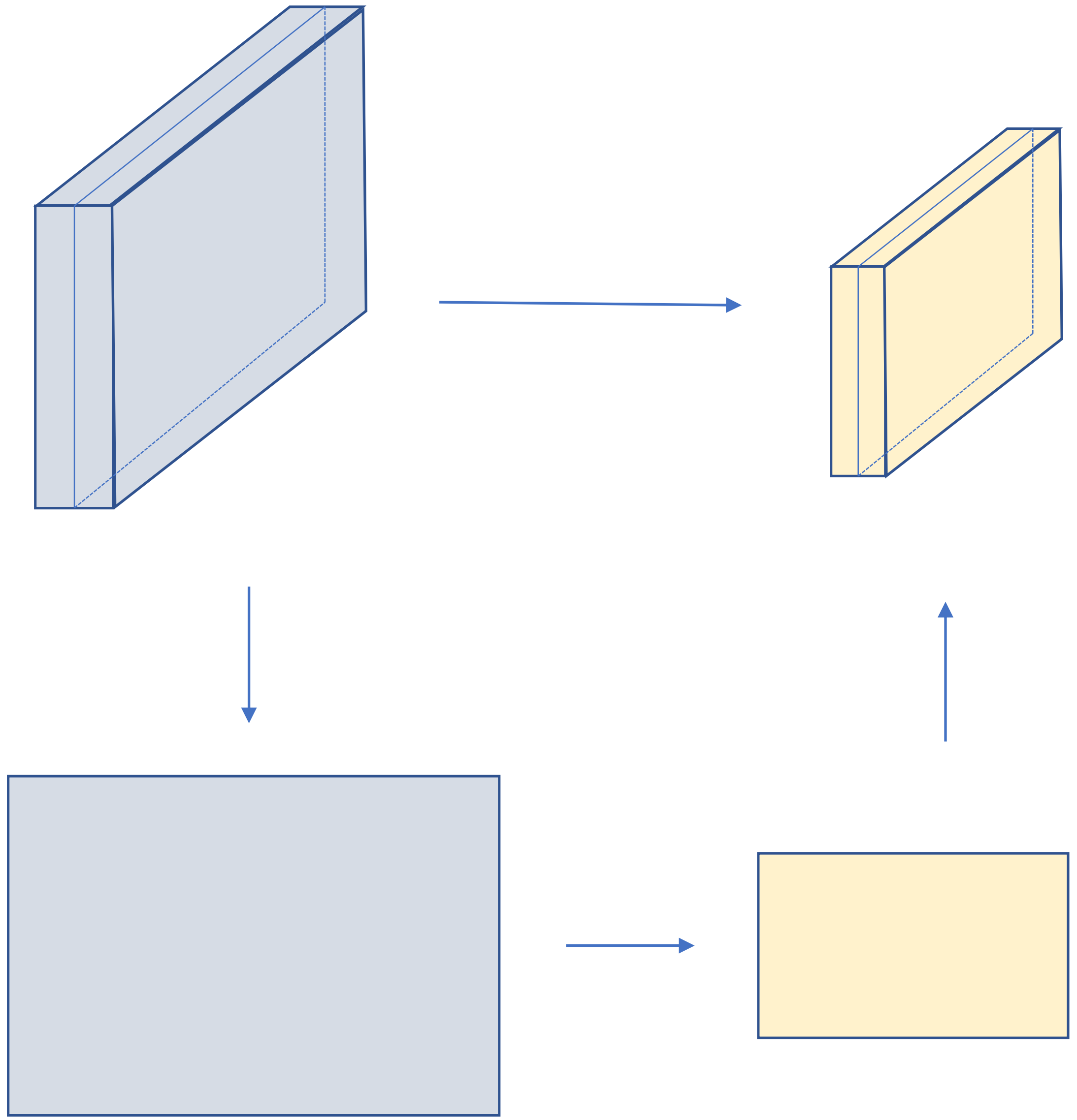




- ✓ A set of convolutional layer and pooling layer are obtained of which the layer are able to identify the images.
- ✓ https://www.youtube.com/watch?v=f0t-OCG79-U

Pooling

- Pooling layers reduce the dimensions of the data by replacing data of specific area to representative value.
- Reduction size of feature
- Reduction amount of calculation



60	88	10	94	74	42	53	60	48
75	48	9	14	61	17	26	93	43
13	70	26	53	31	47	8	52	93
64	23	70	2	28	36	9	25	68
63	10	66	74	30	67	80	8	10
38	38	77	47	52	42	21	88	40
92	31	25	91	87	34	75	66	40
17	44	53	29	69	62	29	29	68
66	59	76	19	26	25	25	21	28

88	94	93
77	74	88
92	91	75

Max Pooling

- Reduce dimension of the data by replacing <u>maximum</u> value of specific area.

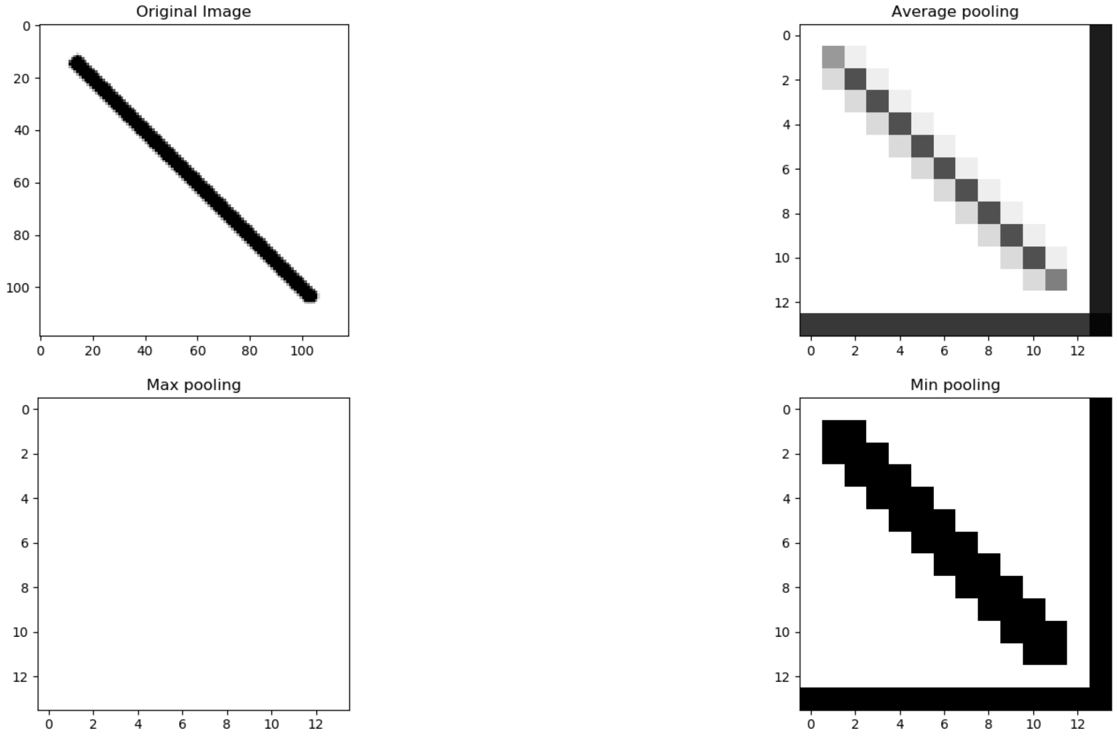
60	88	10	94	74	42	53	60	48
75	48	9	14	61	17	26	93	43
13	70	26	53	31	47	8	52	93
64	23	70	2	28	36	9	25	68
63	10	66	74	30	67	80	8	10
38	38	77	47	52	42	21	88	40
92	31	25	91	87	34	75	66	40
17	44	53	29	69	62	29	29	68
66	59	76	19	26	25	25	21	28

44.33	48.11	52.89
49.89	42	38.78
51.44	49.11	42.33

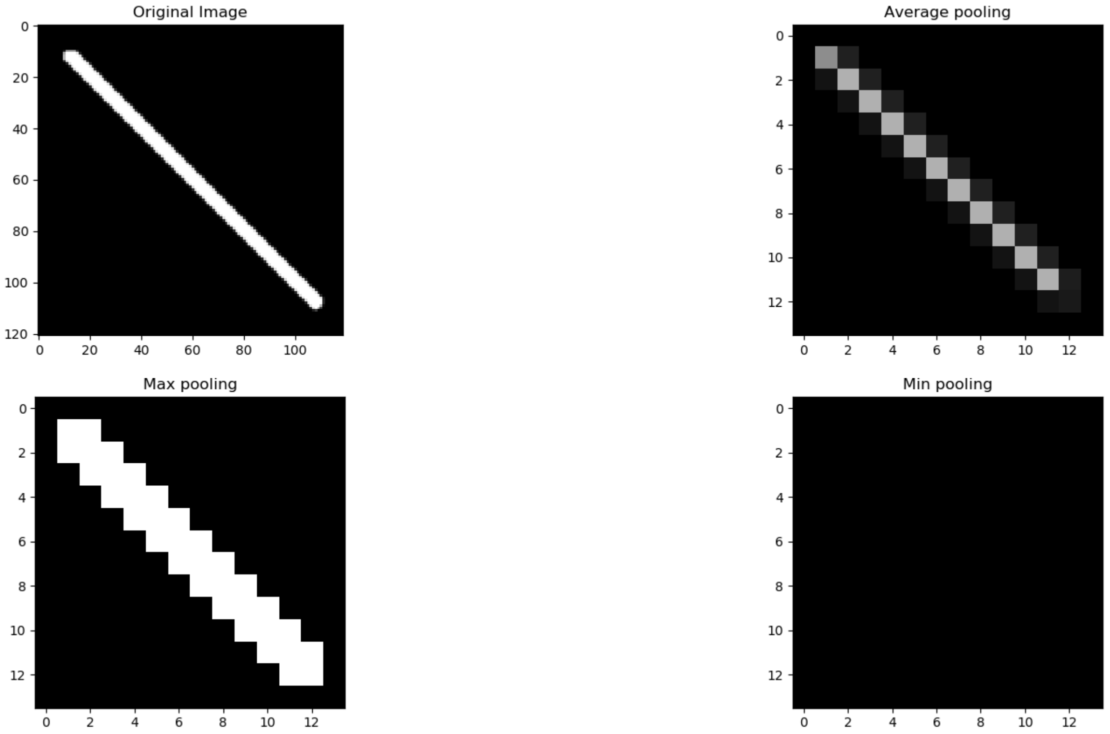
Average Pooling

- Reduce dimension of the data by replacing average value of specific area.

Average, Max and Min pooling of size 9x9 applied on an image



Min pooling gives better result for images with white background and black object

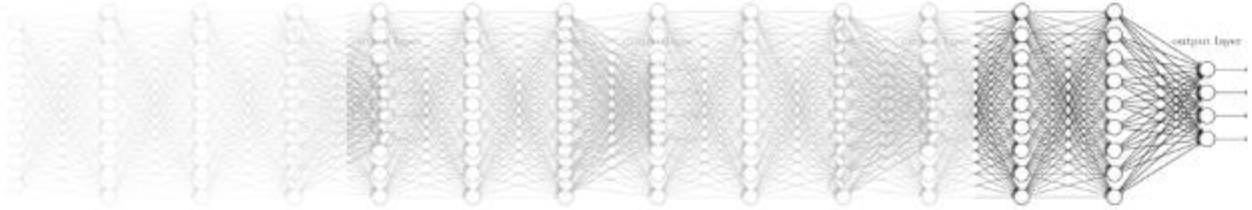


Max pooling gives better result for the images with black background and white object

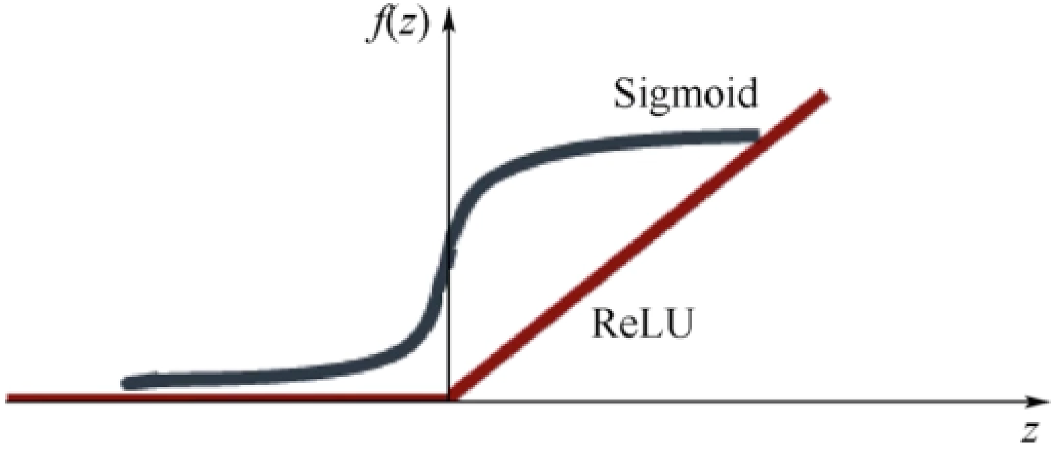
ReLU

- To avoid Vanishing gradient

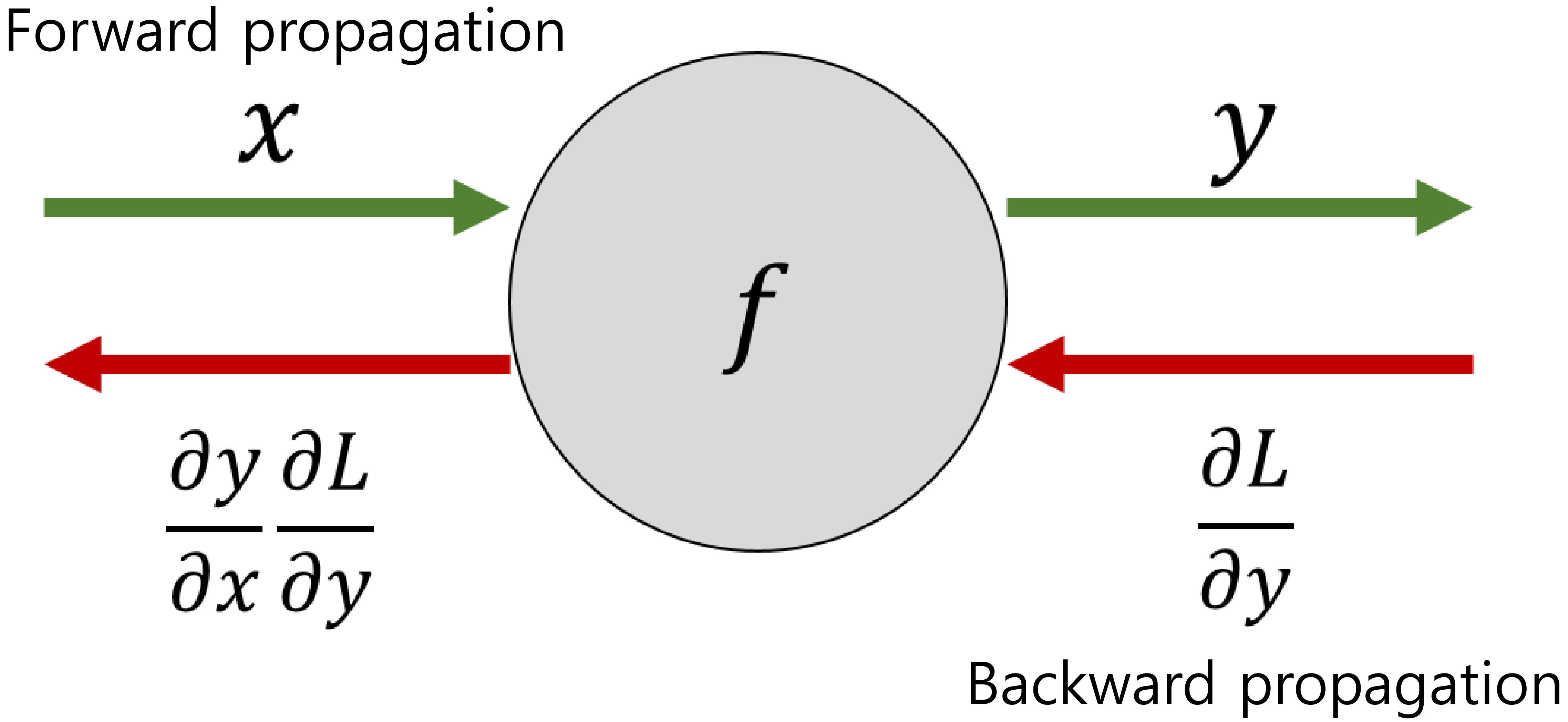
Vanishing gradient (NN winter2: 1986-2006)



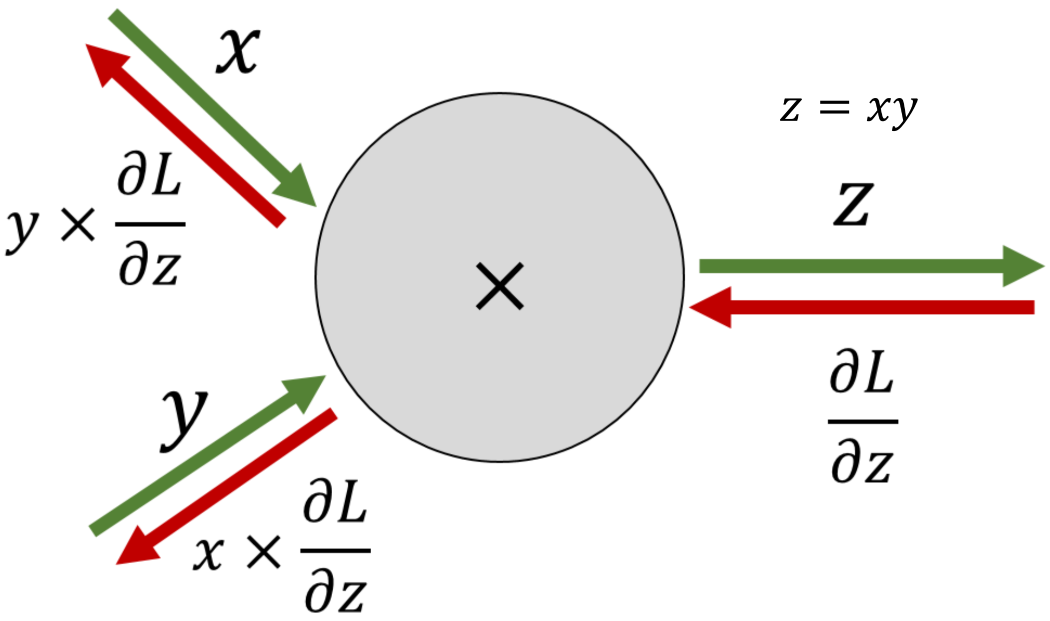
Sigmoid!



Backpropagation

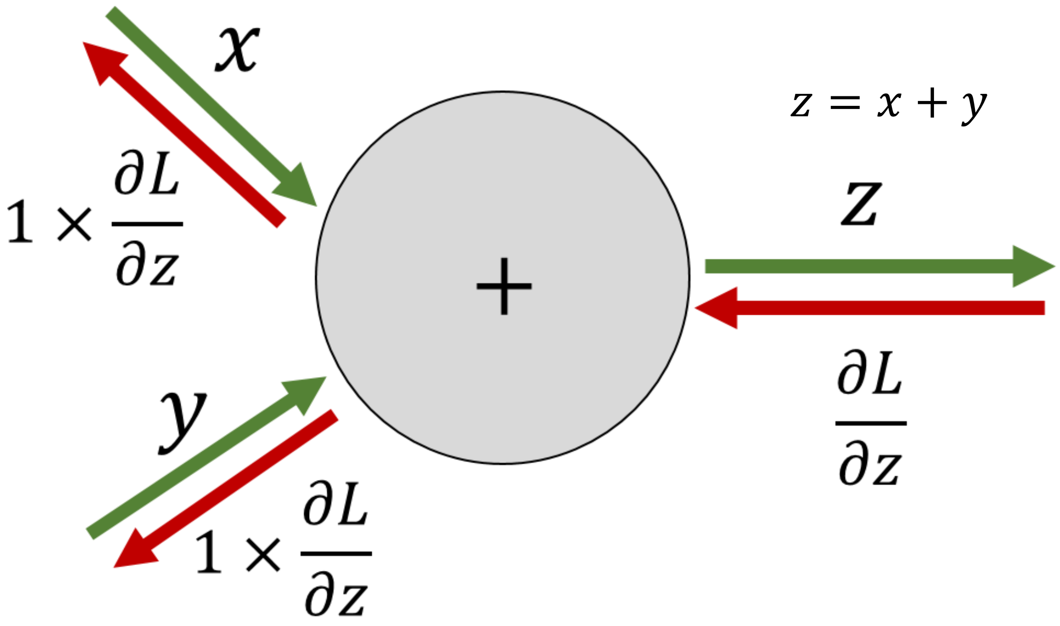


$$\frac{\partial z}{\partial x} = \frac{\partial (xy)}{\partial x} = y$$

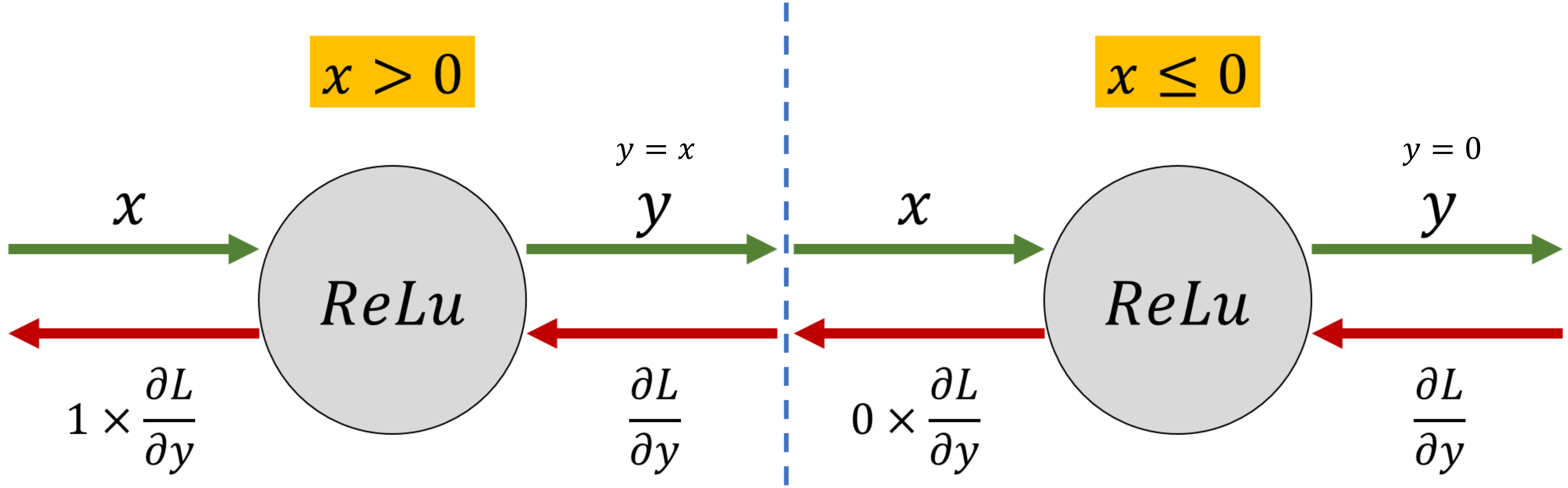


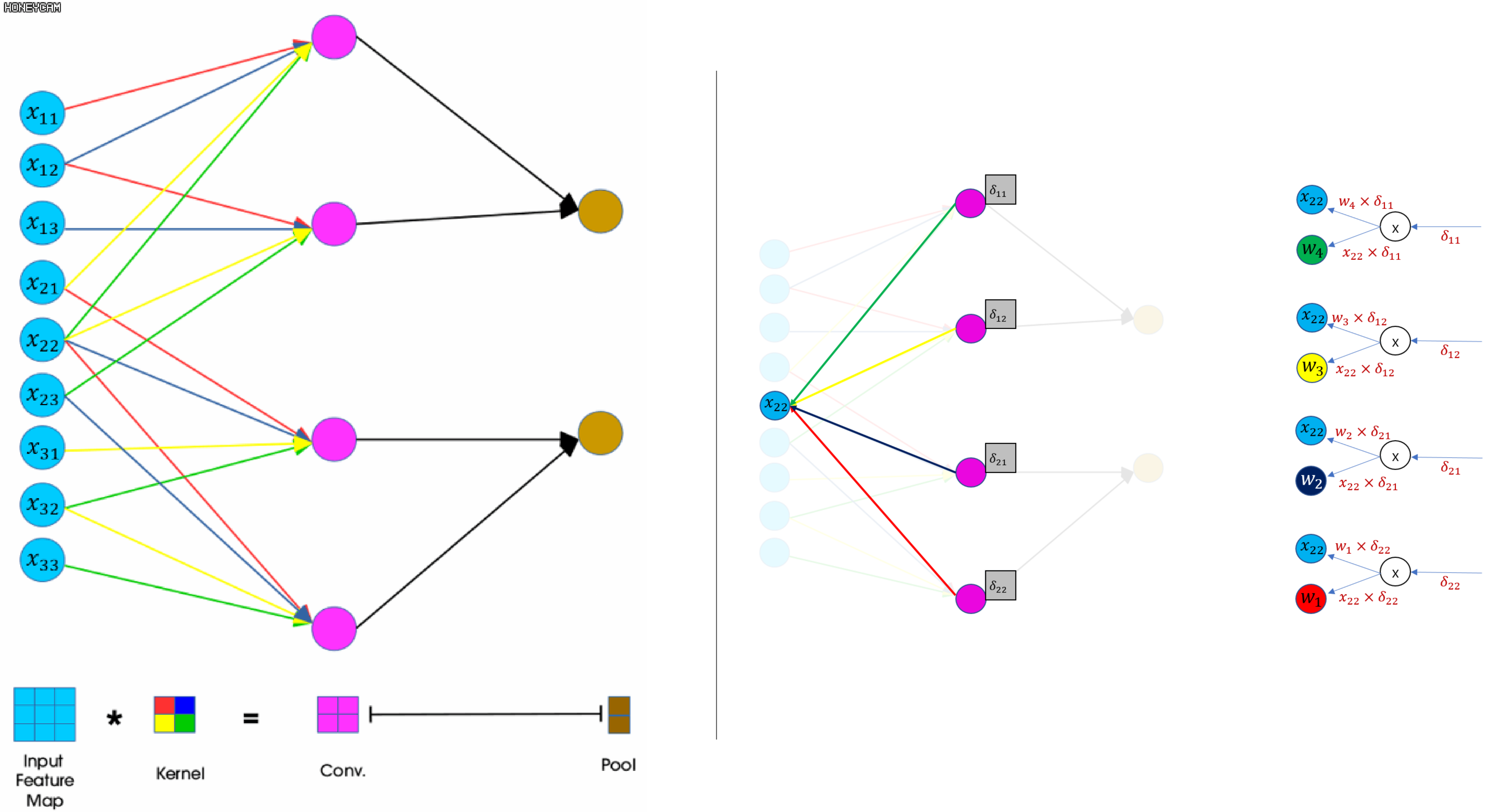
$$\frac{\partial z}{\partial y} = \frac{\partial (xy)}{\partial y} = x$$

$$\frac{\partial z}{\partial x} = \frac{\partial (x + y)}{\partial x} = 1$$

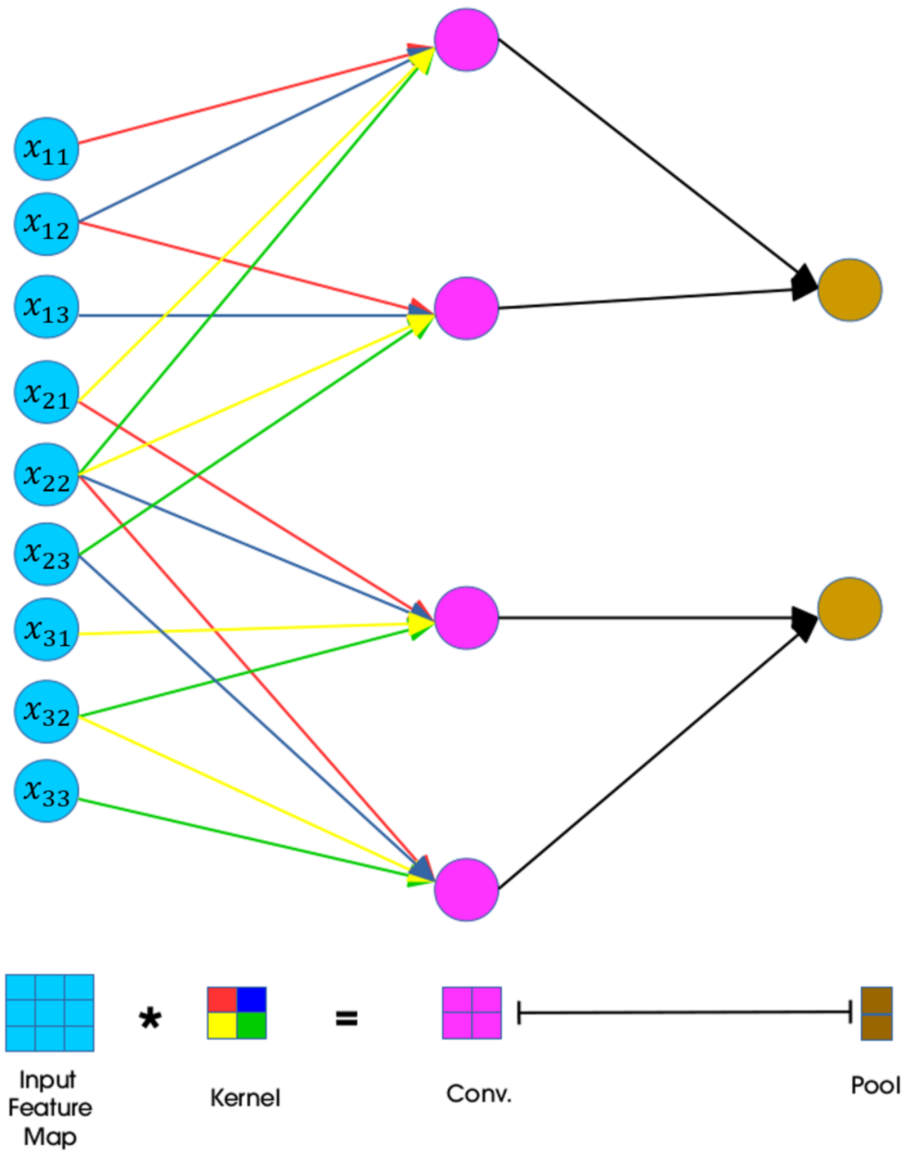


$$\frac{\partial z}{\partial y} = \frac{\partial (x + y)}{\partial y} = 1$$

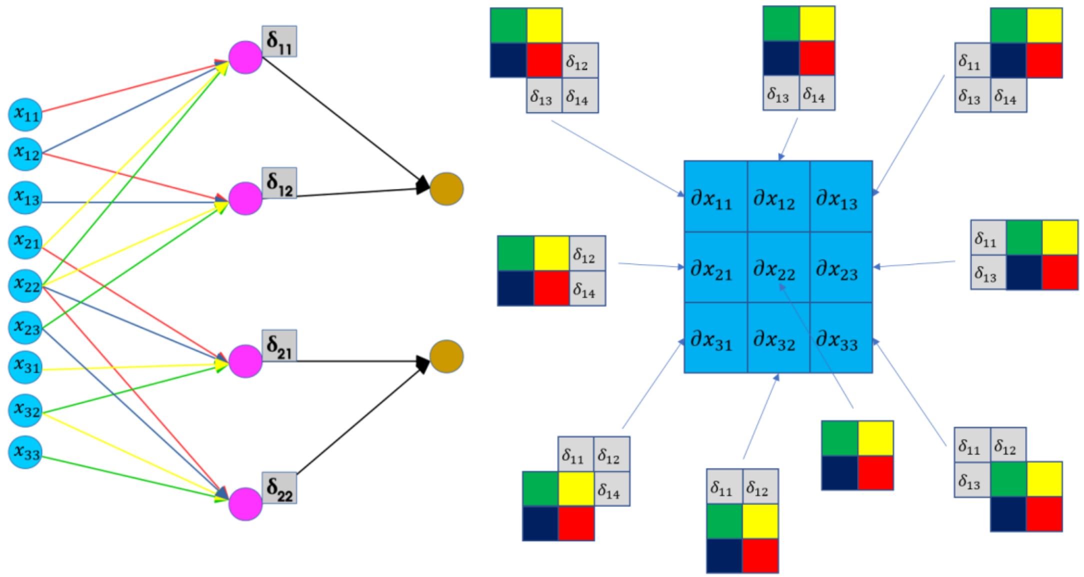

$x > 0$
$y = x$
x
y
$ReLu$
$1 \times \frac{\partial L}{\partial y}$
$\frac{\partial L}{\partial y}$
$x \leq 0$
$y = 0$
x
y
$ReLu$
$0 \times \frac{\partial L}{\partial y}$
$\frac{\partial L}{\partial y}$



HONEYCAM
x_{11} x_{12} x_{13} x_{21} x_{22} x_{23} x_{31} x_{32} x_{33}
*
=
Input Feature Map
Kernel
Conv.
Pool
x_{22}
δ_{11} δ_{12} δ_{21} δ_{22}
x_{22} $w_4 \times \delta_{11}$ W_4 $x_{22} \times \delta_{11}$ x δ_{11}
x_{22} $w_3 \times \delta_{12}$ W_3 $x_{22} \times \delta_{12}$ x δ_{12}
x_{22} $w_2 \times \delta_{21}$ W_2 $x_{22} \times \delta_{21}$ x δ_{21}
x_{22} $w_1 \times \delta_{22}$ W_1 $x_{22} \times \delta_{22}$ x δ_{22}

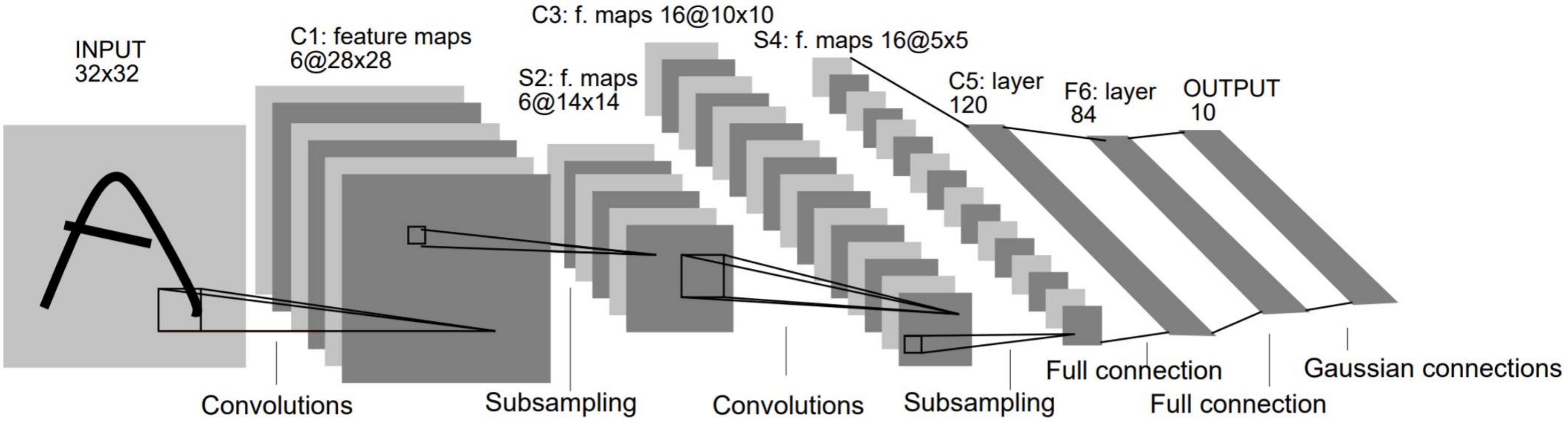


x_{11}
x_{12}
x_{13}
x_{21}
x_{22}
x_{23}
x_{31}
x_{32}
x_{33}
*
=
Input Feature Map
Kernel
Conv.
Pool



x_{11}
x_{12}
x_{13}
x_{21}
x_{22}
x_{23}
x_{31}
x_{32}
x_{33}
δ_{11}
δ_{12}
δ_{21}
δ_{22}
∂x_{11} ∂x_{12} ∂x_{13}
∂x_{21} ∂x_{22} ∂x_{23}
∂x_{31} ∂x_{32} ∂x_{33}
δ_{11} δ_{12} δ_{13} δ_{14}

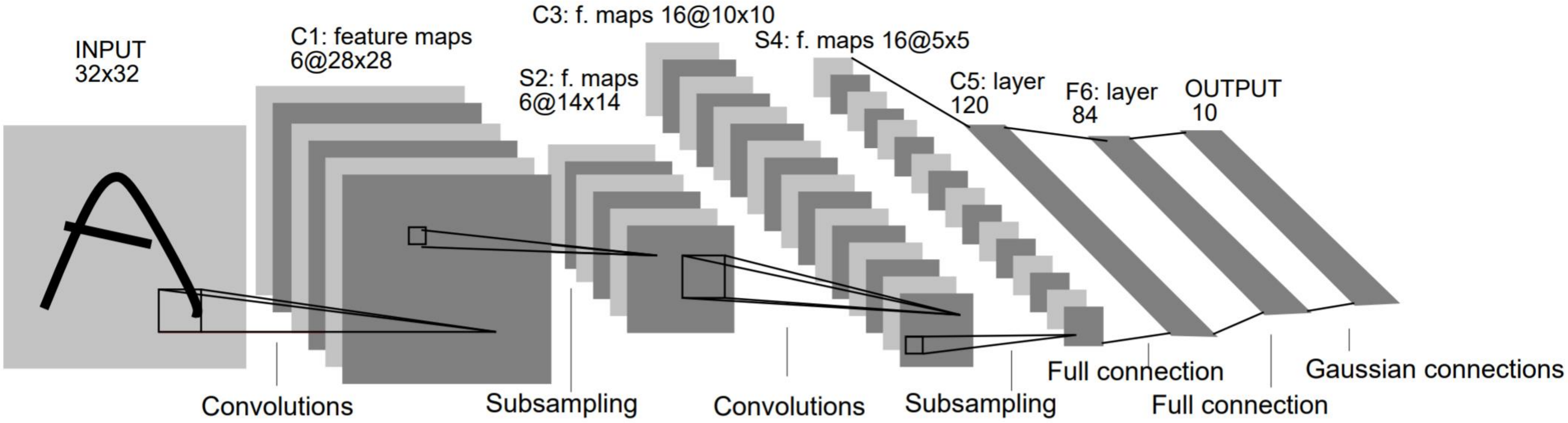
LeNet 5 - Overview



- Three key ideas: Local Receptive Fields, Shared Weights, Sub-sampling(Pooling)

- Input: 32x32 pixel image
- Largest character is 20x20
  (All important info should be in the center of the receptive field of the feature detectors)
- pixel values are normalized (Mean of pixels=0, Std of pixels=1)

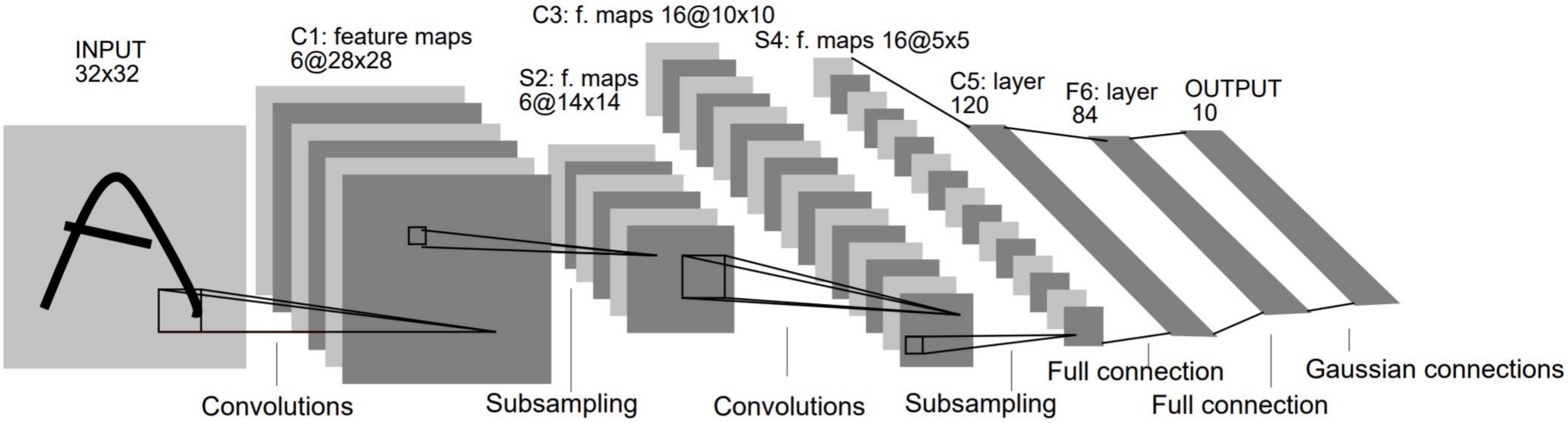
LeNet 5 - Layer C1



- Convolutional layer with 6 feature maps of output size 28x28.
- Each unit of C1 has a 5x5 receptive field in the input layer

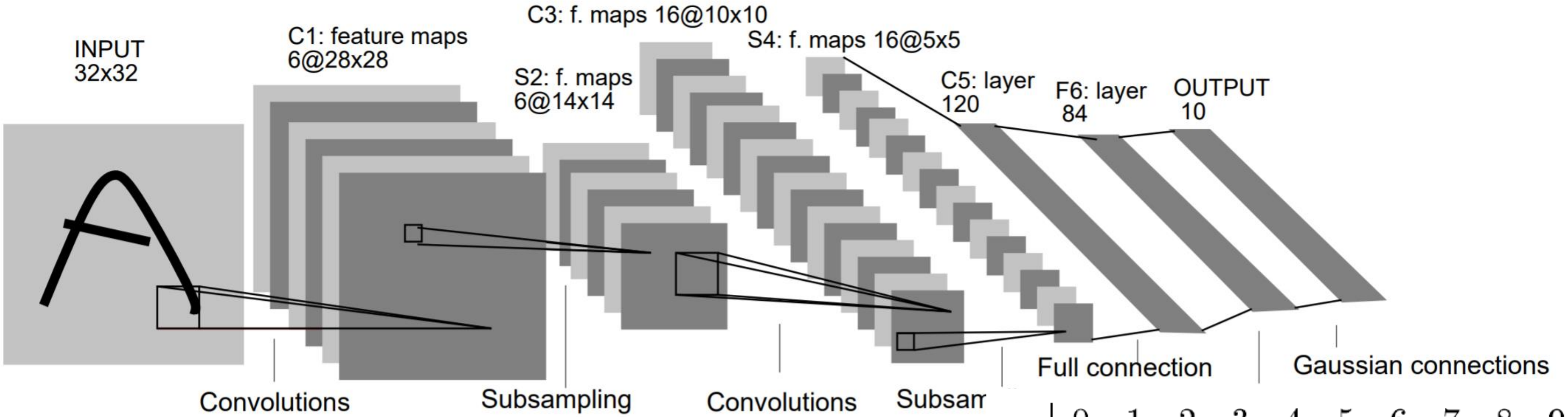
- Parameters: (5x5+1)x6 = 156
- Connections: {28x28}x{(5x5+1)x6} = 122304

LeNet 5 - Layer S2



- Subsampling layer with 6 feature maps of output size 14x14
  (2x2 non overlapping receptive fields in C1 Layer)

LeNet 5 - Layer C3



- Convolutional layer with 16 feature maps of size 10x10

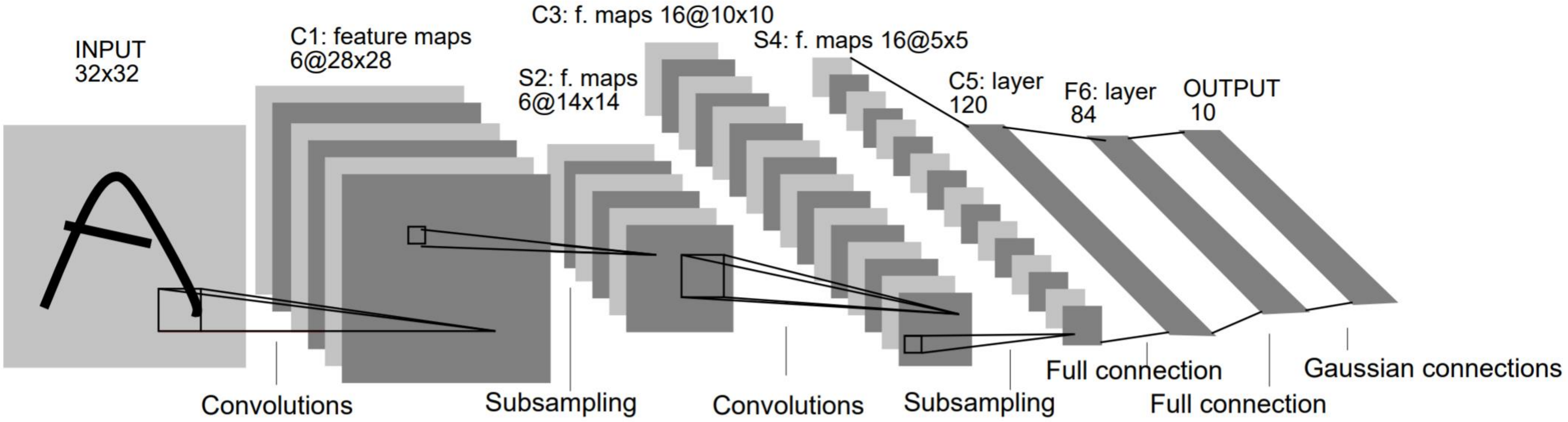
- Parameters: (5x5+1)x16 = 1516
- Connections: 10x10x(5x5+1)x6 = 151600

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

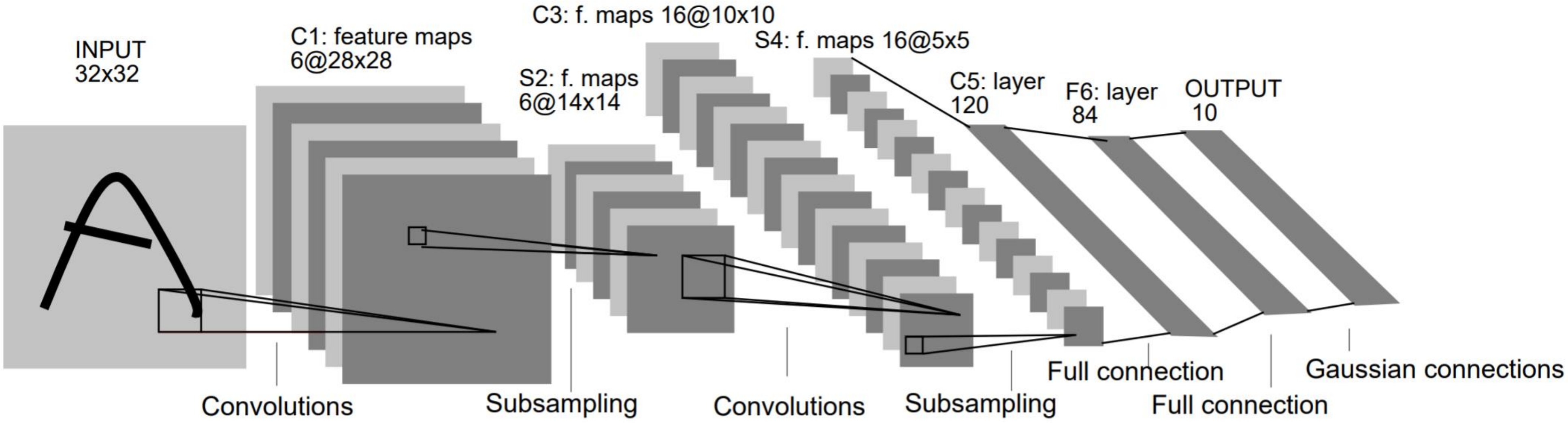
Each column indicates which feature map in S2 are combined by the units in a particular feature map of C3.

LeNet 5 - Layer S4



- Subsampling layer with 16 feature maps of size 5x5
- 2x2 non overlapping receptive fields in C3 Layer

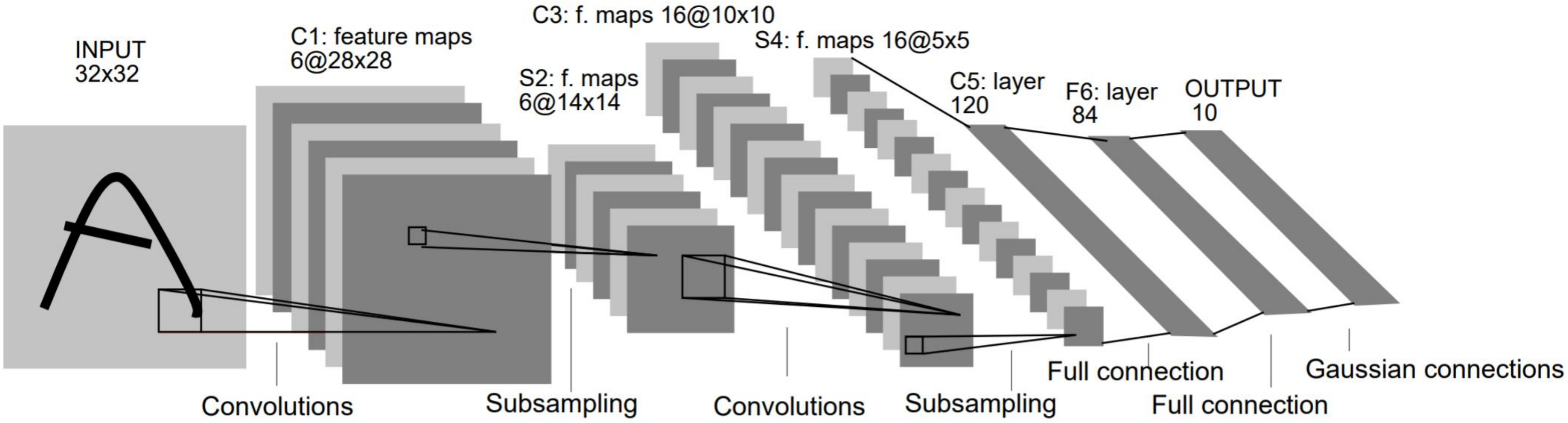
LeNet 5 - Layer C5



- Convolutional layer with 120 feature maps of size 1x1

- Parameters: 120x(16x25+1) = 48120
- Connections: 120x(5x5+1)x16 = 48120

LeNet 5 - Layer F6



- 84 Fully connected units. 84x(120+1) = 10164 is parameters and connections count
- Output layer: 10 classes
- Weight update: Backpropagation

Code Using Colab

- https://colab.research.google.com/drive/1a5I75n01V7vpyZOVa9NGaqsRSMKOM9fH#scrollTo=5nAIPe4IF_oT&forceEdit=true&sandboxMode=true

- https://colab.research.google.com/drive/1mRH80iQBMqQc0aft8Pf8Bk4BZwPVA2Mh#forceEdit=true&sandboxMode=true

참고 문헌

- Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient- Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, November 1998
- Harsh Agrawal (Sept 8th, 2015) ECE: 6504, Deep Learning For Perception
- **Convolutional Neural Networks of deeplearning.ai hosted in Coursera**
- https://arclab.tistory.com/150
- http://blog.naver.com/PostView.nhn?blogId=laonple&logNo=220648539191
- https://reniew.github.io/07/
- https://bskyvision.com/418