



LeNet-5

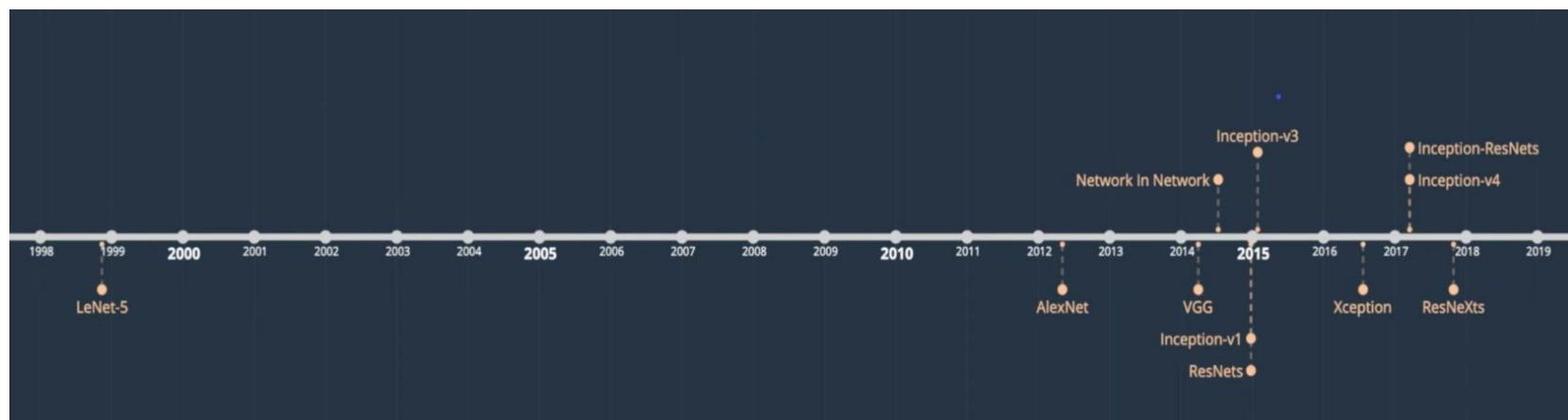
Computer Vision & Augmented Reality 연구실
학부연구생 강 준 구



Contents

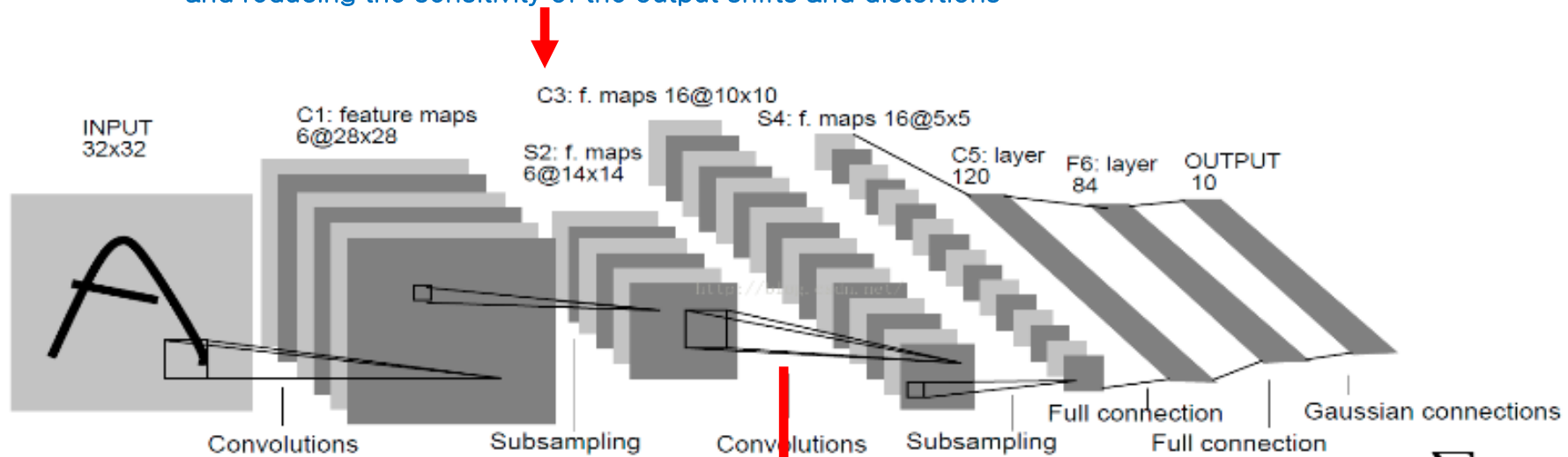
- ▶ Implementing LeNet-5 CNN Architecture
- ▶ Comparison
- ▶ Future work

LeNet-5(1998)



LeNet-5 Architecture

“a subsampling, reducing the resolution of the feature map, and reducing the sensitivity of the output shifts and distortions”



“It forces a break of symmetry in the network.”

	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

$$y_i = \sum_j (x_j - w_{ij})^2$$

Euclidean
Radial Basis Function units
x=input vector,
w=parameter vector
i= 10, j=84

Implementing LeNet-5

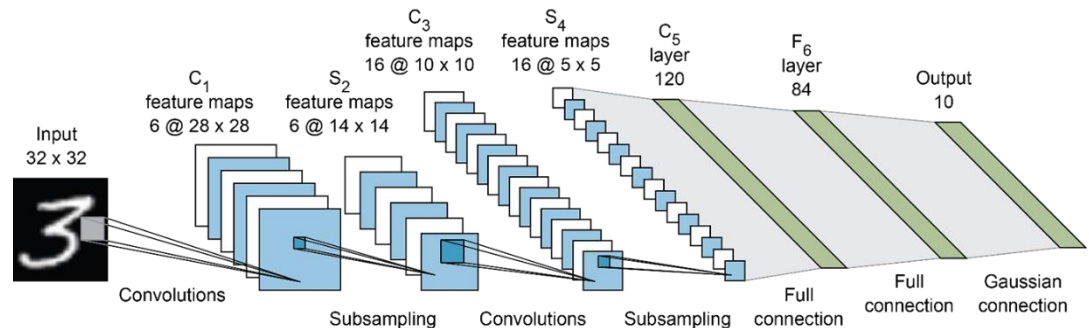
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, AveragePooling2D
from tensorflow.keras.layers import Flatten, Dense
from tensorflow.keras.layers import ZeroPadding2D

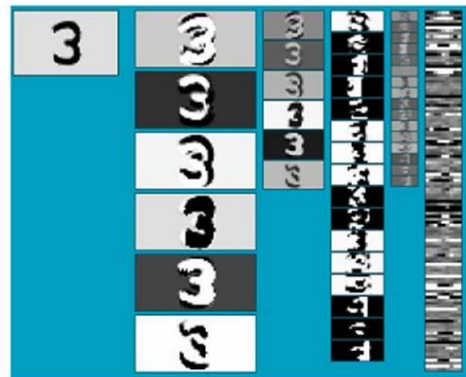
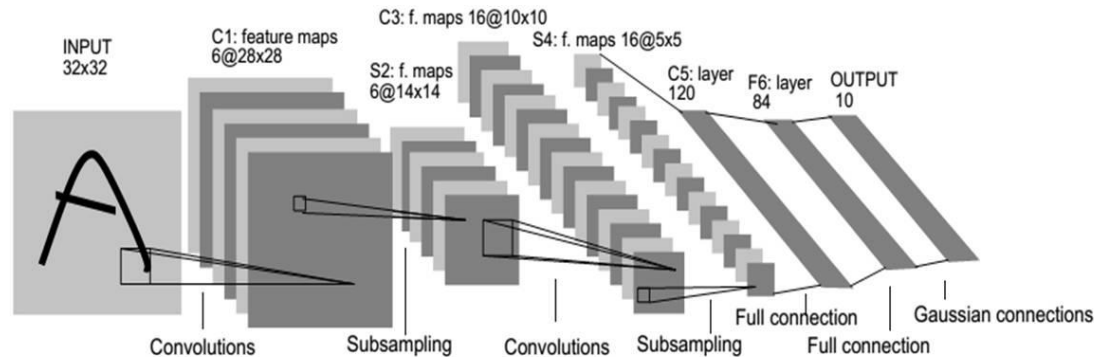
LeNet5=Sequential()
LeNet5.add(ZeroPadding2D(padding=2))
LeNet5.add(Conv2D(filters=6, kernel_size=5, padding='valid', strides=1, activation='tanh'))
LeNet5.add(AveragePooling2D(pool_size=2, strides=2))
LeNet5.add(Conv2D(filters=16, kernel_size=5, padding='valid', strides=1, activation='tanh'))
LeNet5.add(AveragePooling2D(pool_size=2, strides=2))
LeNet5.add(Flatten())
LeNet5.add(Dense(units=120, activation='tanh'))
LeNet5.add(Dense(units=84, activation='tanh'))
LeNet5.add(Dense(units=10, activation='softmax'))

LeNet5.build(input_shape=(None,28,28,1))
LeNet5.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
zero_padding2d (ZeroPadding2D)	(None, 32, 32, 1)	0
conv2d (Conv2D)	(None, 28, 28, 6)	156
average_pooling2d (AveragePooling2D)	(None, 14, 14, 6)	0
conv2d_1 (Conv2D)	(None, 10, 10, 16)	2416
average_pooling2d_1 (AveragePooling2D)	(None, 5, 5, 16)	0
flatten (Flatten)	(None, 400)	0
dense (Dense)	(None, 120)	48120
dense_1 (Dense)	(None, 84)	10164
dense_2 (Dense)	(None, 10)	850
Total params: 61,706		
Trainable params: 61,706		
Non-trainable params: 0		

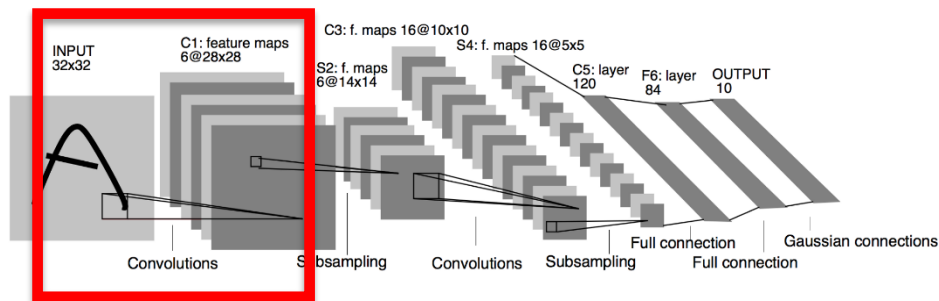




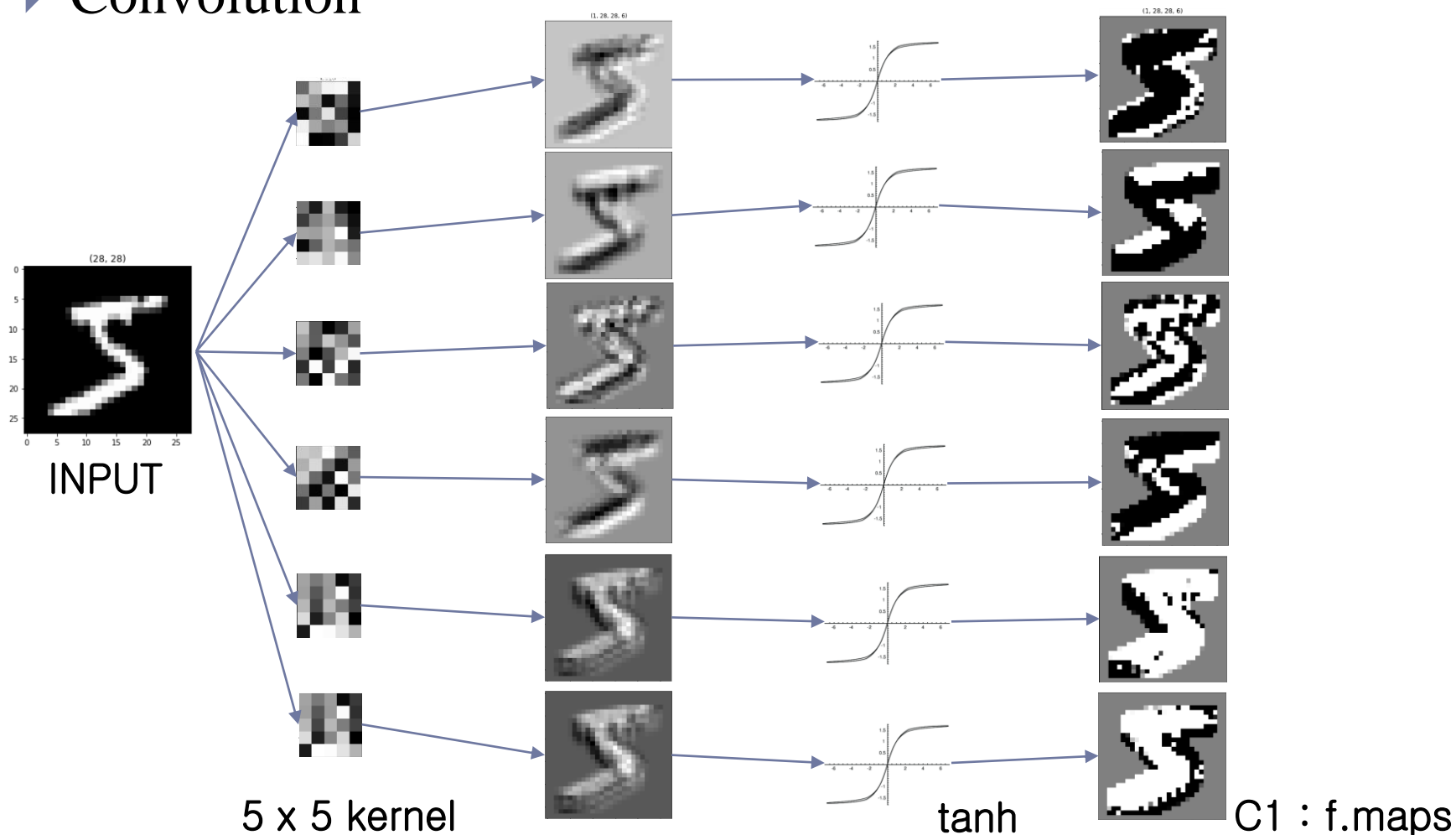
[3D Visualization of a Convolutional Neural Network \(cmu.edu\)](https://github.com/engineerjkk/DeepLearning_from_Scratch/blob/main/Final_LeNet.ipynb)

https://github.com/engineerjkk/DeepLearning_from_Scratch/blob/main/Final_LeNet.ipynb

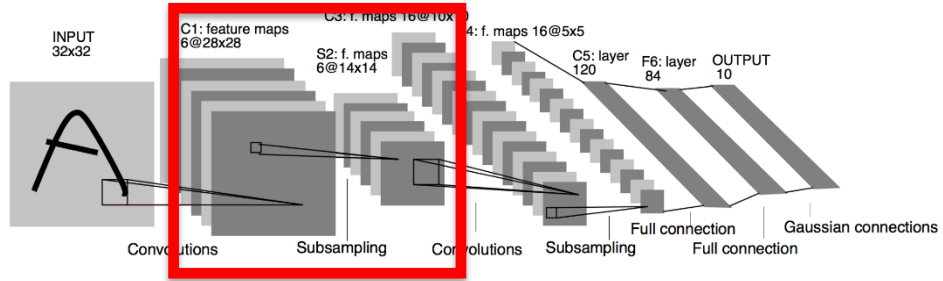
InPut -> C1 : feature maps



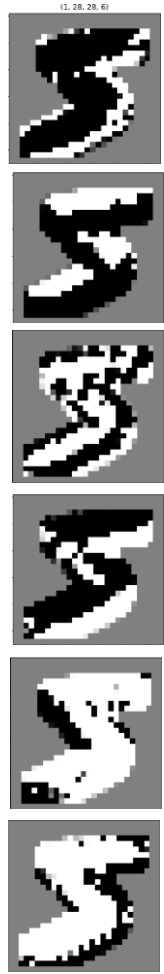
► Convolution



C1 -> S2 feature maps



Subsampling



C1 : f.maps

7	3	5	2
8	7	1	6
4	9	3	9
0	8	4	5

$$\frac{7 + 3 + 8 + 7}{4}$$

Average pooling

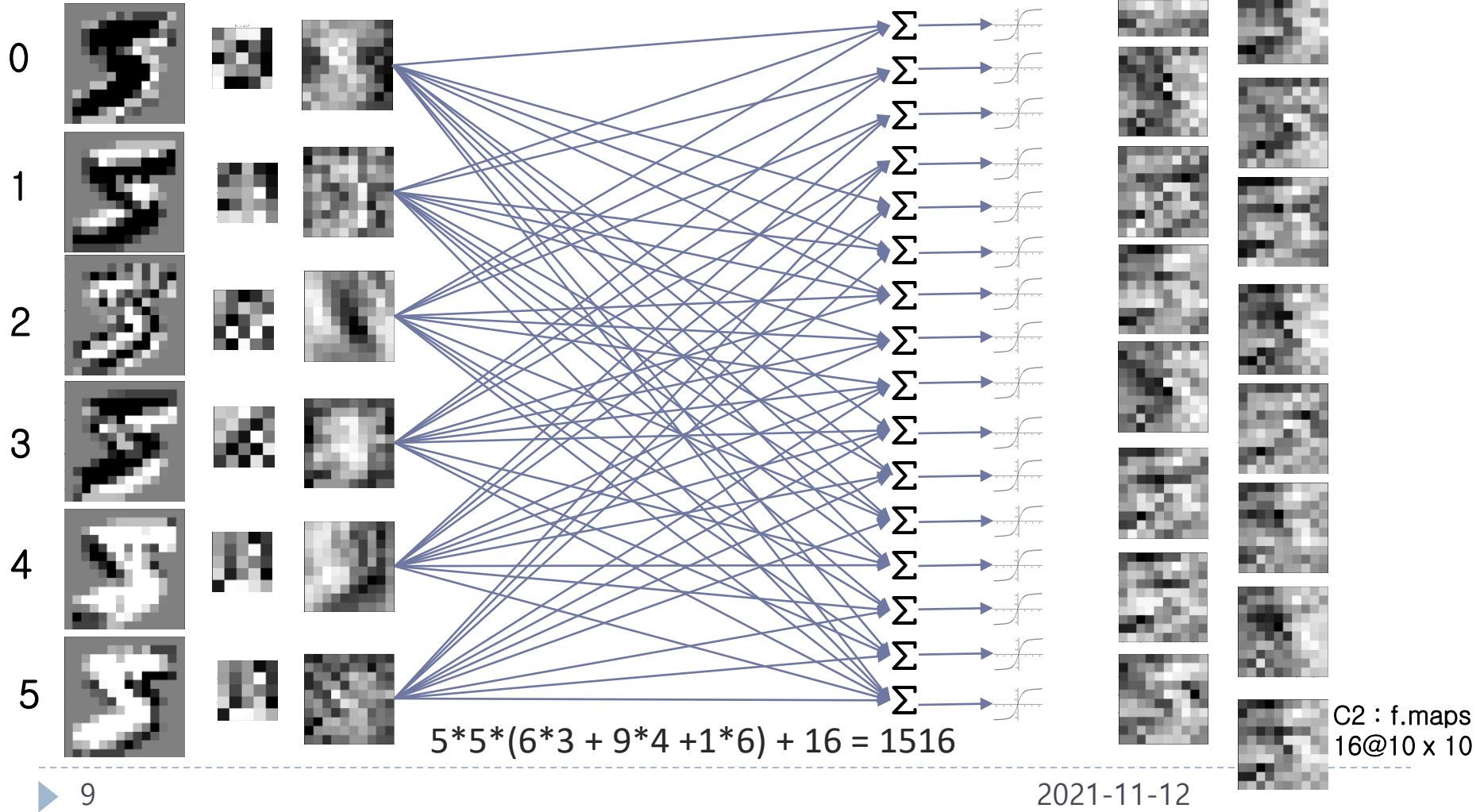
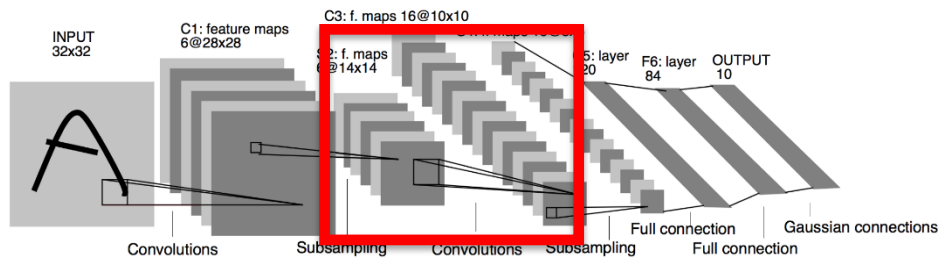
6.25	



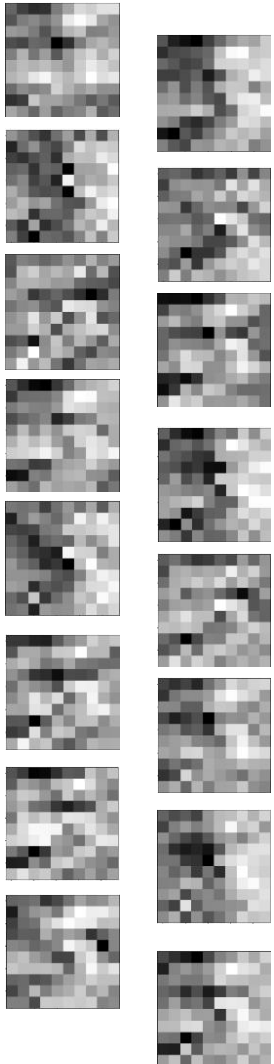
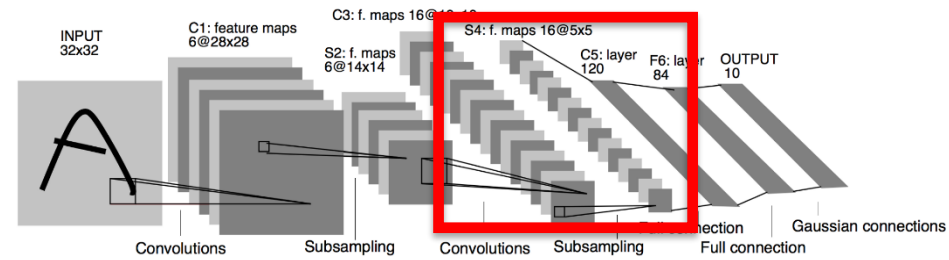
S2 : f.maps
6@14 x 14

	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



Subsampling C3->S4

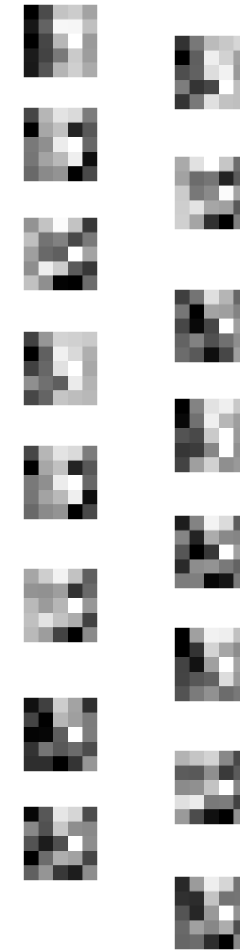


7	3	5	2
8	7	1	6
4	9	3	9
0	8	4	5

$$\frac{7 + 3 + 8 + 7}{4}$$

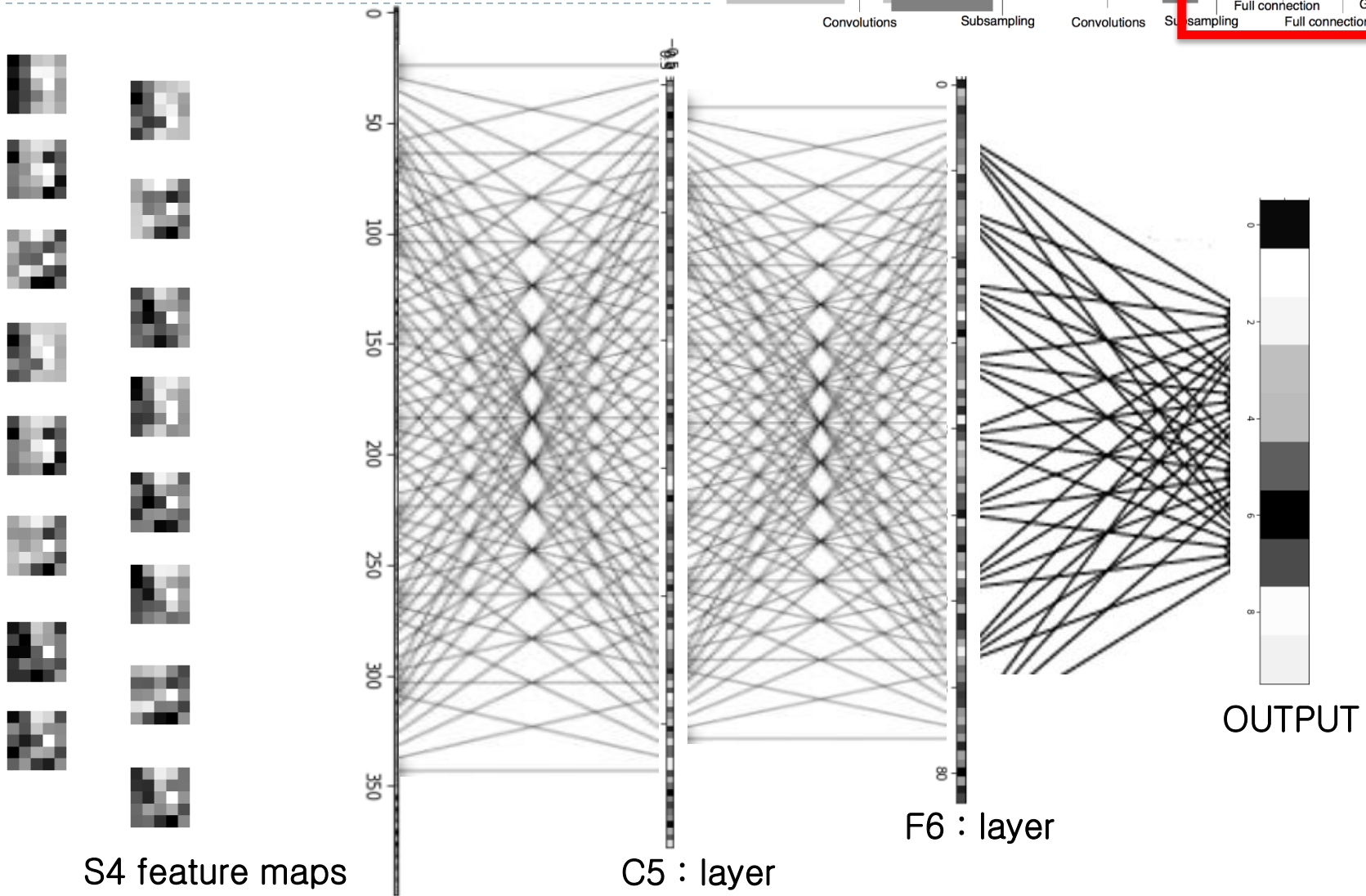
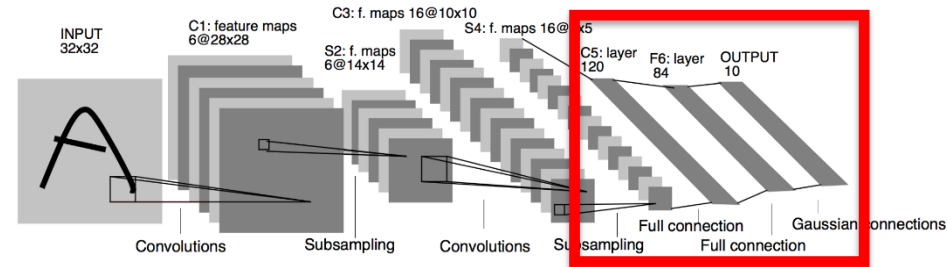
Average pooling

6.25	



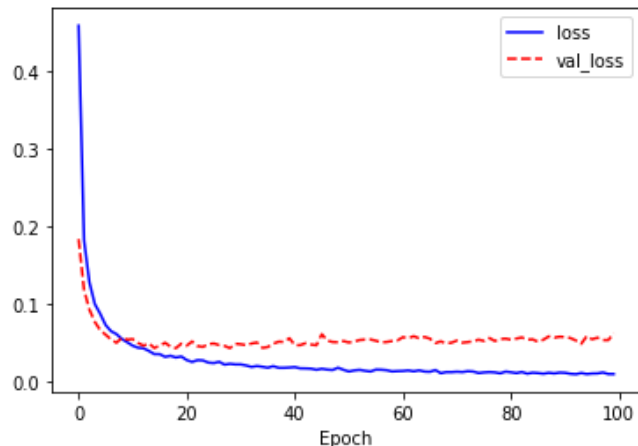
S4 : f.maps
16@5 x 5

Classifier

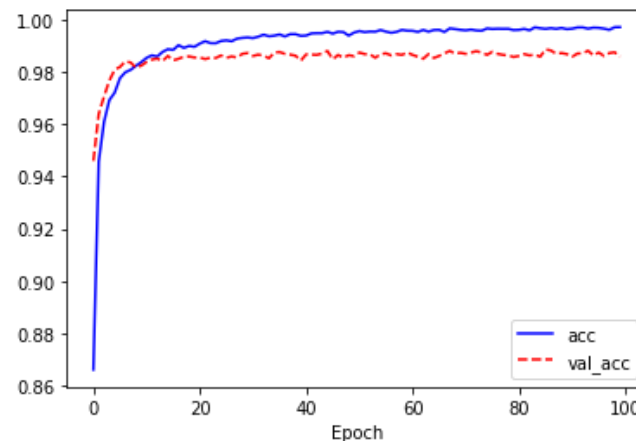


Comparison

LeNet5

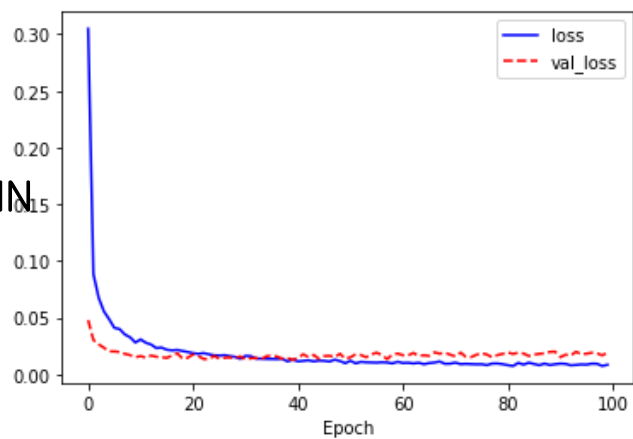


loss: 0.0617

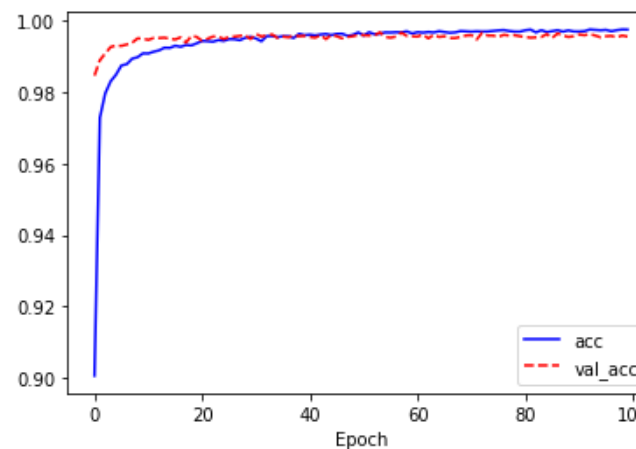


accuracy: 0.9857

Latest CNN



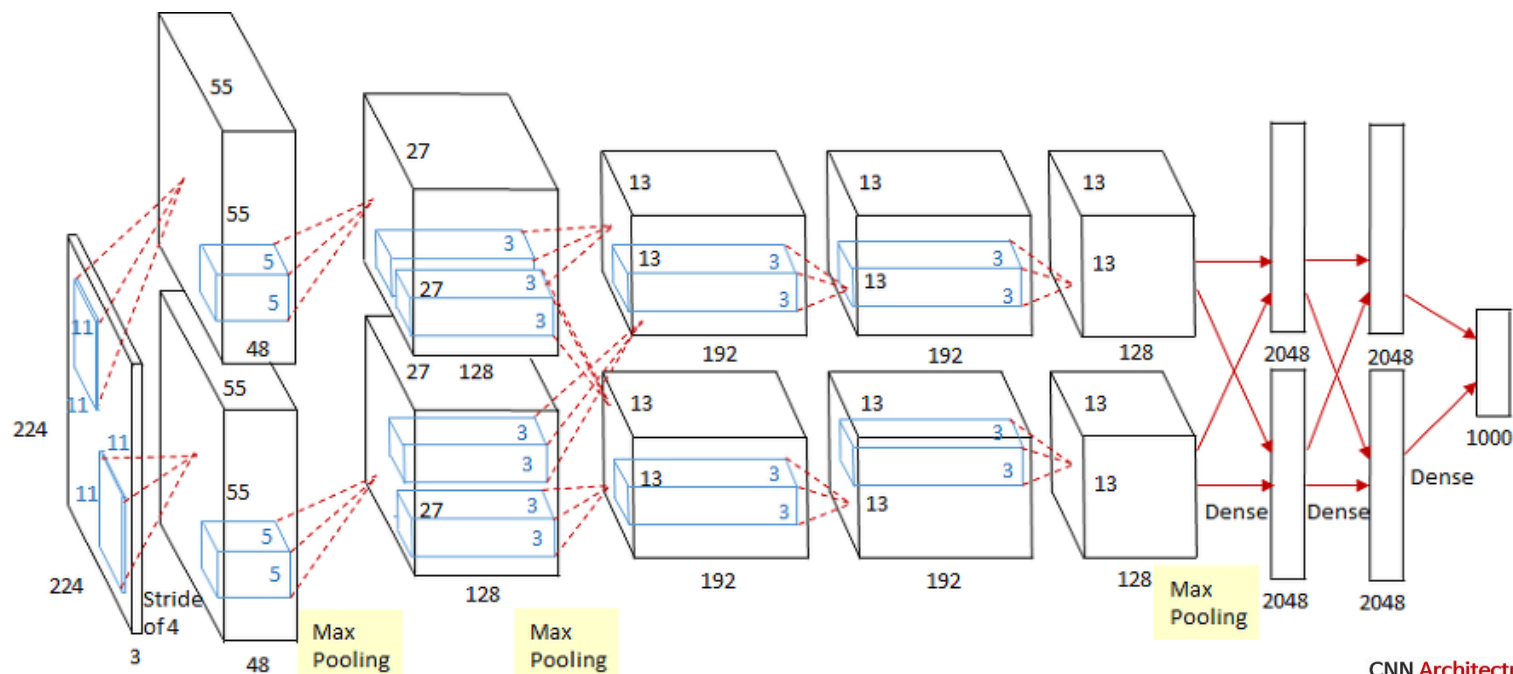
loss: 0.0191



accuracy: 0.9955

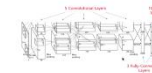
Future work

► AlexNet



CNN Architectures

AlexNet



GoogLeNet



VGG



ResNet

