LeNet-5 Introduction

- a neural network architecture for handwritten and machine-printed character recognition -

2020

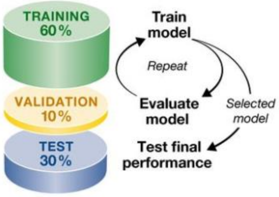
Ando Ki, Ph.D.
adki@future-ds.com

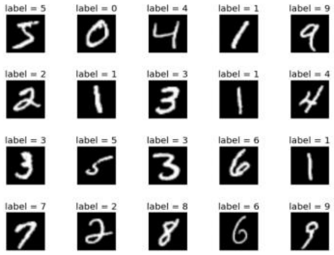
Table of contents

- MNIST
- LeNet-5 for MNIST
- LeNet-5 layer 1-7
- References

MNIST

- MNIST (Modified National Institute of Standards and Technology database)
  - Modified National Institute of Standards and Technology
  - Handwritten digits database
    - 10 classes: 0, 1, …, 9
    - training set: 60,000 training image
    - test set: 10,000 testing image

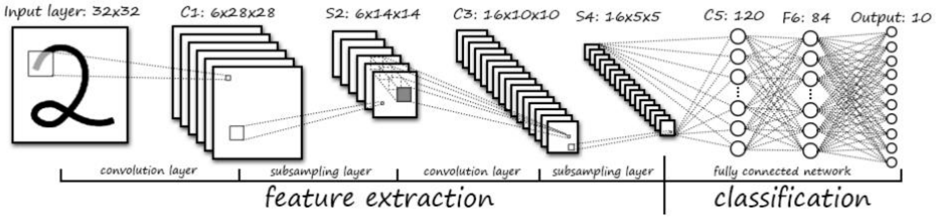




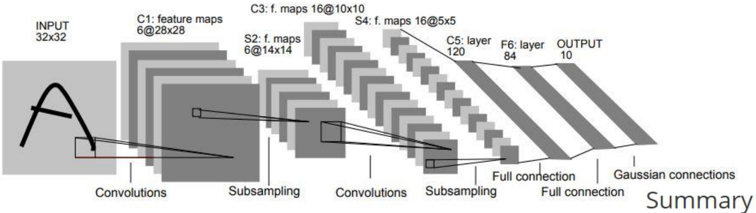
Why background color is black?

LeNet-5 for MNIST

- LeNet is one of the popular convolutional networks, and works well on digit classification tasks.
  - 1024 (32x32) inputs of black and white ➔ converted to floating number 0.0 ~ 1.0
  - 10 outputs representing digit 0 to 9



LeNet-5 for MNIST

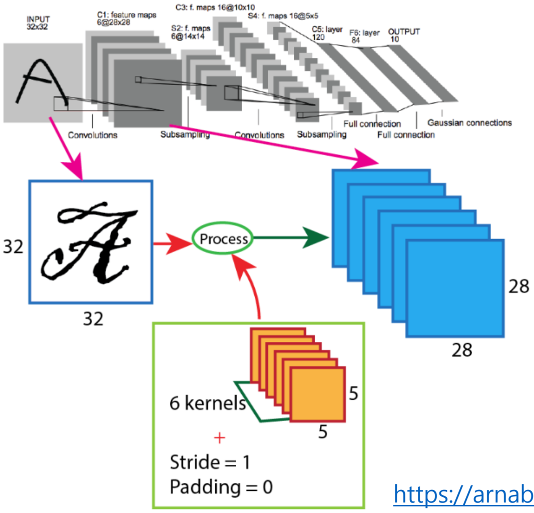


Original Image published in [LeCun et al., 1998]

Summary of LeNet-5 Architecture

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

LeNet-5 for MNIST: layer 1

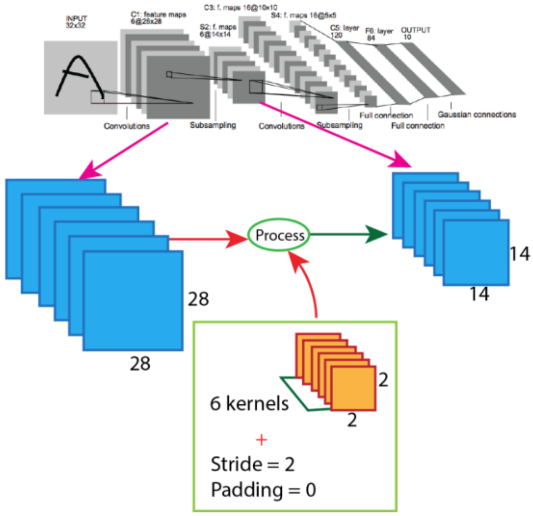


- 1^{st} convolution layer
  - Input: 32x32 pixels (W=32)
  - Convolution filter: 6 kernels with 5x5 (K=5)
  - Convolution padding: 0 (P=0)
  - Convolution: stride 1 (S=1)
  - Results in: 6 features of 28x28

$$Output = \frac{W-K+2(P)}{S}+1 = \frac{32-5+2(0)}{1}+1 = 28$$

https://arnabfly.github.io/arnab_blog/lenet5/

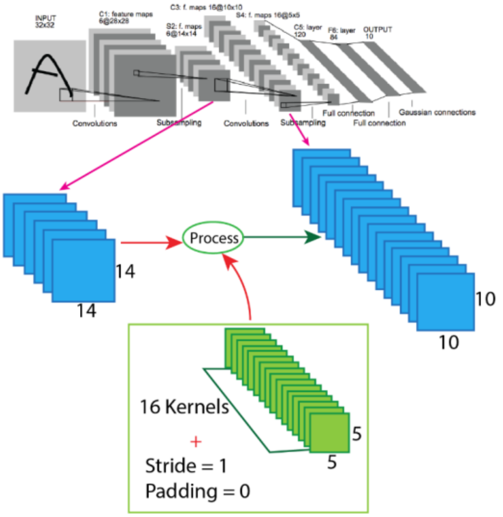
LeNet-5 for MNIST: layer 2



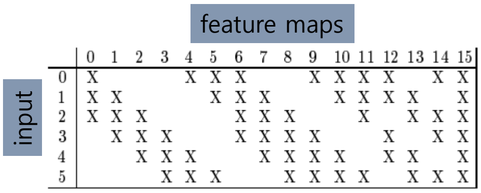
- 1st pooling layer (sub-sampling)
  - Input: 6 features with 28x28 (W=28)
  - Max pooling filter: 2x2 (K=2)
  - Pooling padding: 0 (P=0)
  - Pooling: stride 2 (S=2)
    - It generates ½ number of elements
  - Results in: 6 features of 14x14

$$Output = \frac{W - K + 2(P)}{S} + 1 = \frac{28 - 2 + 2(0)}{2} + 1 = 14$$

LeNet-5 for MNIST: layer 3



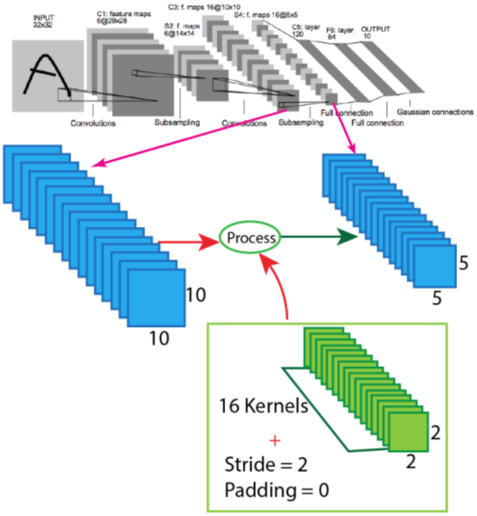
- 2nd convolution
  - Input: 6 features with 14x14 pixels (W=14)
    - 6 kernels are used at the previous stage
  - Convolution filter: 16 kernels with 5x5 (K=5)
  - Convolution padding: 0 (P=0)
  - Convolution: stride 1 (S=1)
    - It generates the same number of elements
  - Results in: 16 features of 10x10



feature maps

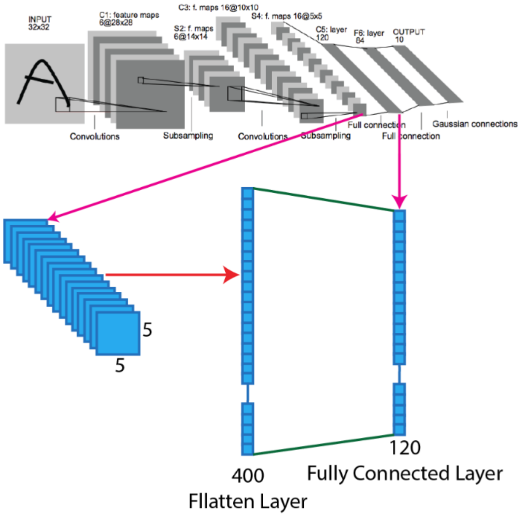
input	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

LeNet-5 for MNIST: layer 4



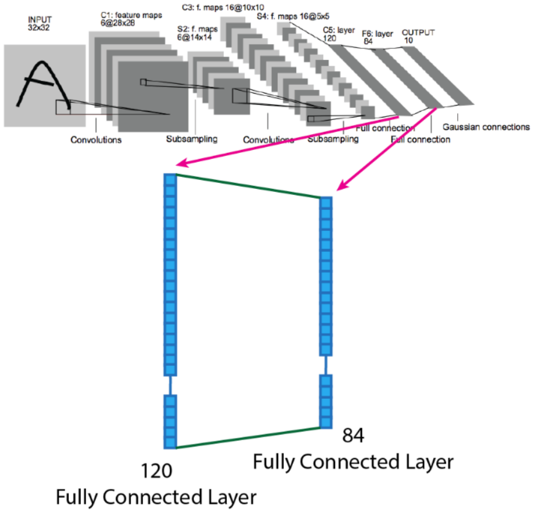
- 2nd pooling
  - ▶ Input: 16 features with 10x10
  - ▶ Max pooling filter: 2x2
  - ▶ Pooling padding: 0
  - ▶ Pooling: stride 2
    - ➲ It generates ½ number of elements
  - ▶ Results in: 16 features of 5x5

LeNet-5 for MNIST: layer 5



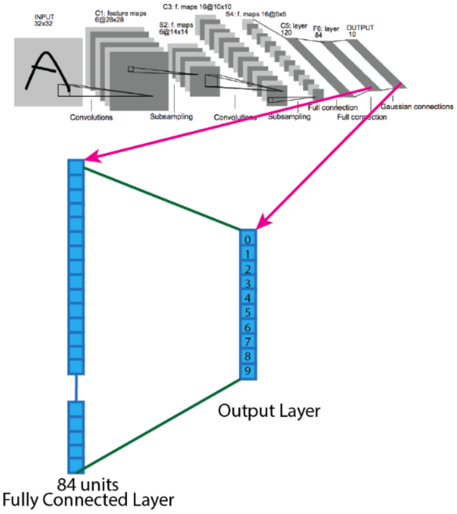
- fully connected layer for flatten
  - ▶ Input: 16 features with 5x5
  - ▶ Reshaping: 3-D array to 1-D vector
    - ➲ 16x5x5 ➔ 400
  - ▶ Output: 120
    - ➲ Neurons: 120

LeNet-5 for MNIST: layer 6



- fully connected layer for flatten
  - Input: 120 feature map
  - Output: 84
    - Neurons: 84

LeNet-5 for MNIST: layer 7

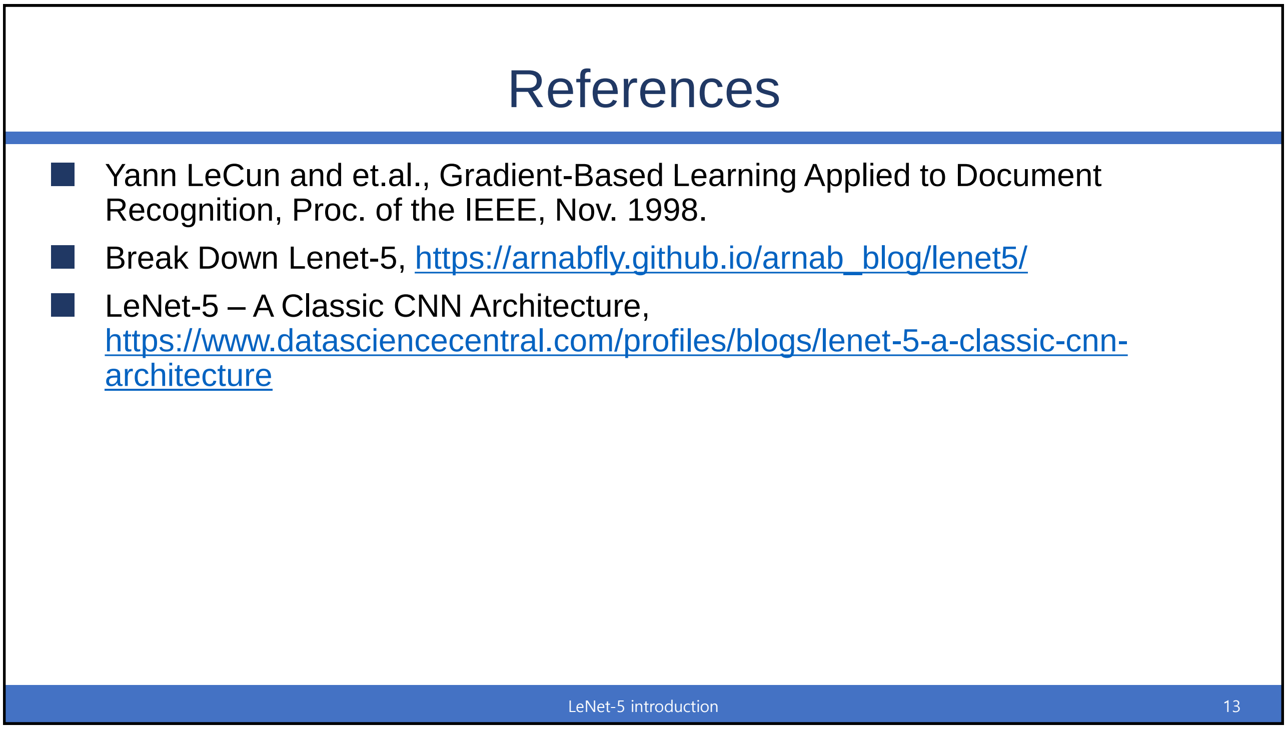


- fully connected layer for flatten
  - Input: 84 feature map
  - Output: 10
    - Neurons: 10

The output layer is composed of Euclidean Radial Basis Function unit (RBF), one for each class, with 84 inputs each. The outputs of each RBF unit i-th y is computed as follows.

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

Nowadays, softmax is used instead.



References

- Yann LeCun and et.al., Gradient-Based Learning Applied to Document Recognition, Proc. of the IEEE, Nov. 1998.
- Break Down Lenet-5, https://arnabfly.github.io/arnab_blog/lenet5/
- LeNet-5 – A Classic CNN Architecture, https://www.datasciencecentral.com/profiles/blogs/lenet-5-a-classic-cnn-architecture