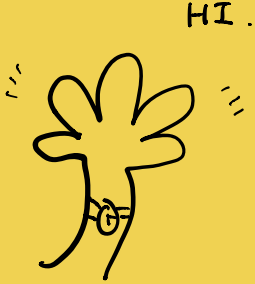
Lecture 5.
Convolutional Neural Network



Review

1. Limitation of "Linear" Classifier
   - XOR Gate
2. Perceptron
   - Perceptron = Linear Classifier
   - Analogy to Neurons
   - Building Block of the Neural Network
3. "MLP"
   - MLP and the Neural Network
   - The Universal Approximation Theorem
   - Where Backpropagation becomes Important
4. "Limitations of MLP"

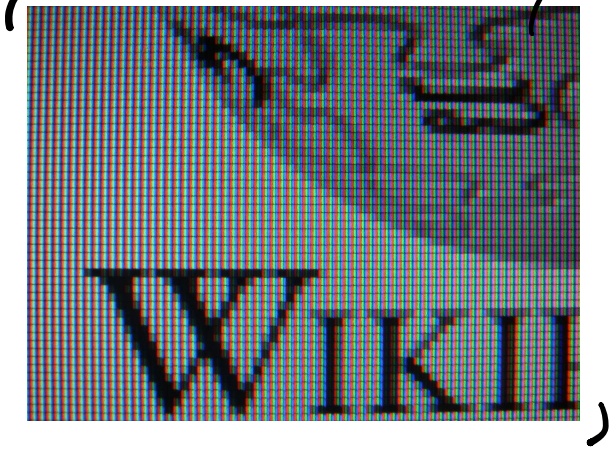
Today's Contents

CNN

1. Back to Image Classification
2. Convolutional Neural Network

1. Back to "Image Classification" : Pixels

Image = Matrix of Pixels



Pixel.
Picture Element.

Back to Image Classification : Pixels

0 ~ 255

What we see

What computer sees (grayscale)

$$\mathbf{x} = \begin{pmatrix} 251, 081, 068, 041, 032, 071, 197, \\ 196, 014, 132, 213, 187, 043, 041, \\ 174, 011, 200, 254, 244, 177, 164, \\ 202, 014, 012, 128, 242, 255, 255, \\ 253, 212, 089, 005, 064, 196, 253, \\ 255, 255, 251, 196, 030, 009, 165, \\ 127, 162, 251, 254, 197, 009, 105, \\ 062, 005, 100, 144, 097, 006, 170, \\ 207, 083, 032, 051, 053, 134, 250 \end{pmatrix}$$

Image from researchgate.net

Back to Image Classification : Pixels

RGB

Image from https://negliadesign.com/

CIFAR-10

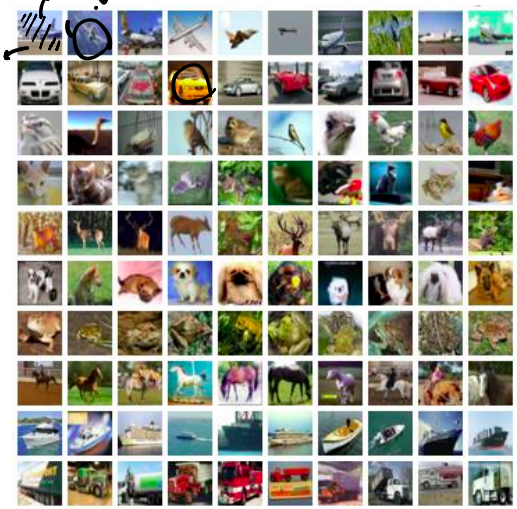
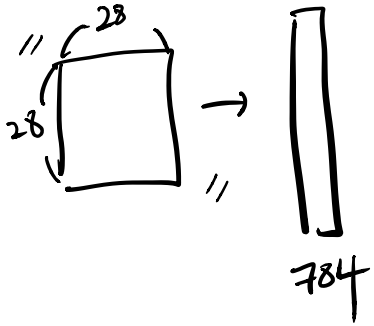
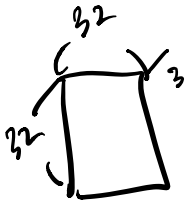


Image from https://www.kaggle.com/

Back to Image Classification : MLP



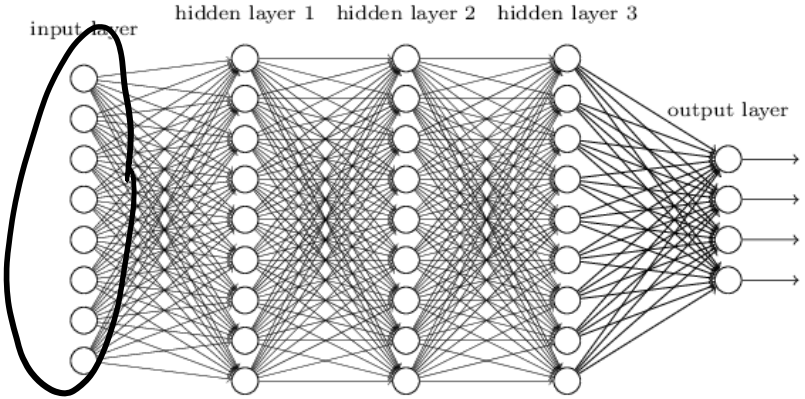


32x32x3

= 1024x3

= 3072

MLP



Back to Image Classification : MLP

Must Stretch Out Tensor into Vector

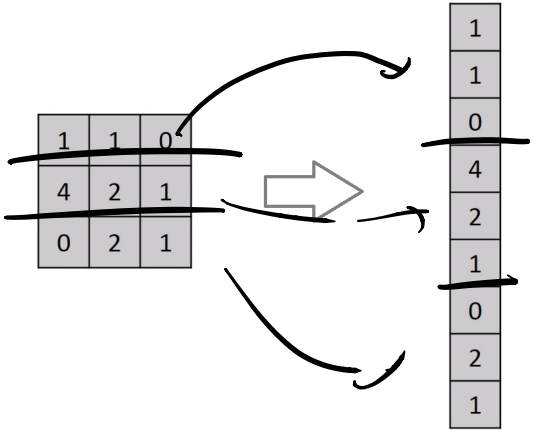


Image from https://towardsdatascience.com

Back to Image Classification : MLP

Worked well on MNIST

What about on CIFAR-10?

10%

10%

Back to Image Classification (MLP)

$3072 \times h_1 + h_1 \times h_2 + h_2 \times 10$

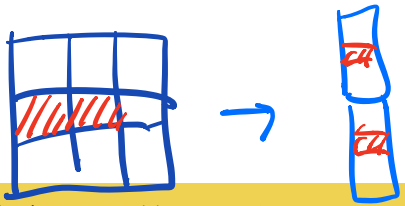
$3072 \rightarrow h_1 \rightarrow h_2 \rightarrow 10$

However, MLPs are not appropriate for Image Classification...

1. Too many parameters required

2. Spatial structure is lost

↳ Matrix/Tensor. ⟶ Vector.

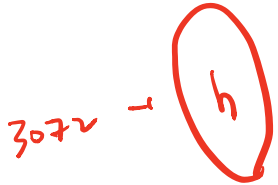


Back to Image Classification : MLP

We want a model that,

1. Uses fewer parameters than MLPs
2. Preserves the spatial structure

Back to Image Classification : MLP



Recall The Universal Approximation Theorem...

"1개의 Hidden Layer를 가진 MLP로 어떤 함수도 근사할 수 있다."

However,

"A feedforward network with a single layer is sufficient to represent any function,
But the layer may be infeasibly large and may fail to learn and generalize correctly."

CNN : Idea

We want a model that,

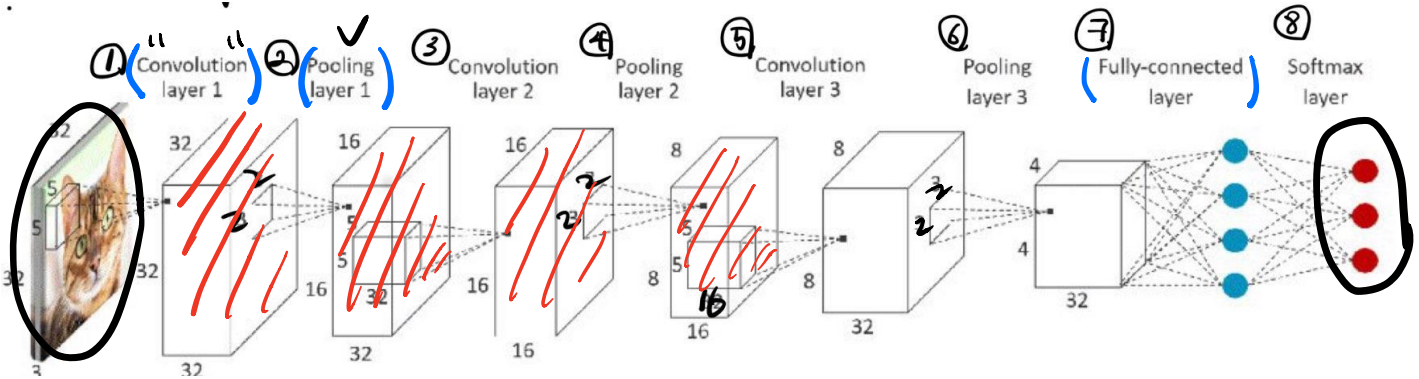
1. Uses fewer parameters than MLPs

2. Preserves the spatial structure

Convolutional Neural Network to the rescue!!!

CNN : Idea

Image from https://community.arm.com/



What happened?!?!?!

CNN : Idea

1 What happens during Tensor → Tensor Mapping?

2 What are Convolution, Pooling, Fully-Connected Layers?

3 How does this Model extract features from spatial structure?

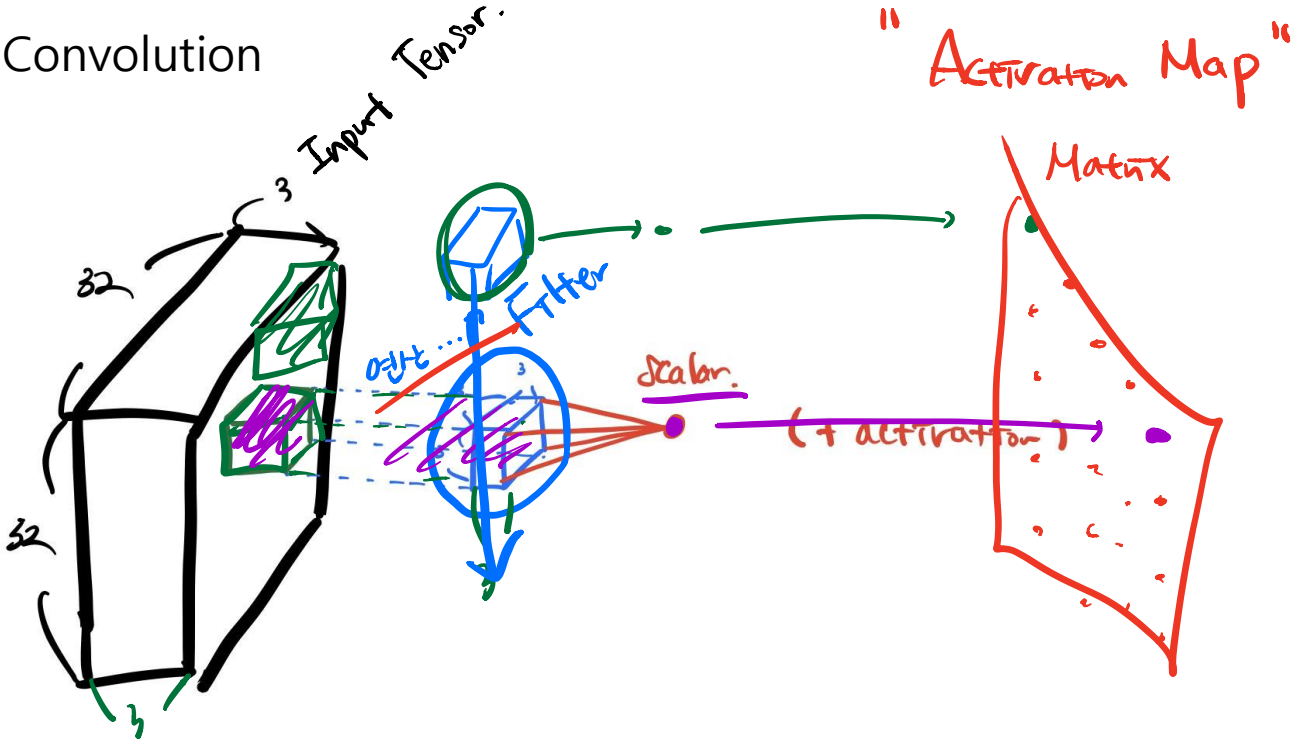
4 Does it use less parameters than the MLP?

CNN : Convolution

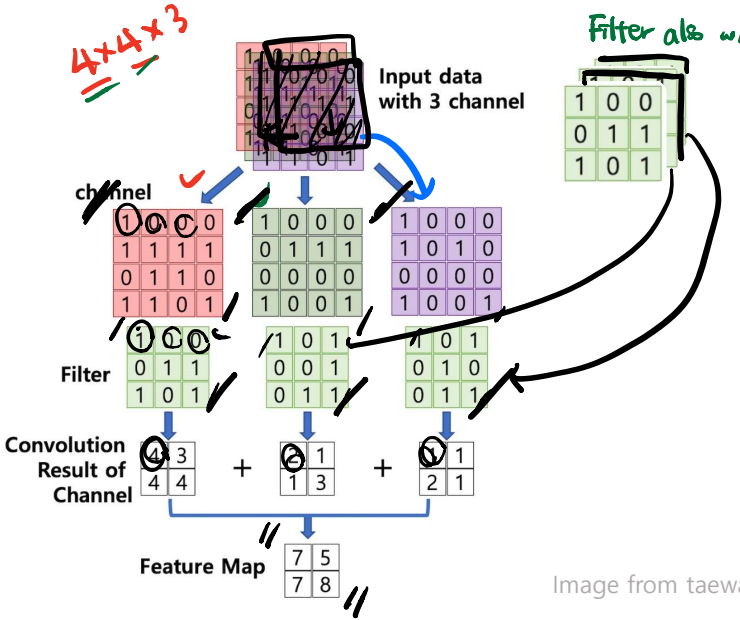
How do we map Tensor to Tensor, while preserving the spatial structure?

Apply **Filter** to small regions in the Tensor (Image)

CNN : Convolution



CNN : Convolution



While sliding over the tensor,

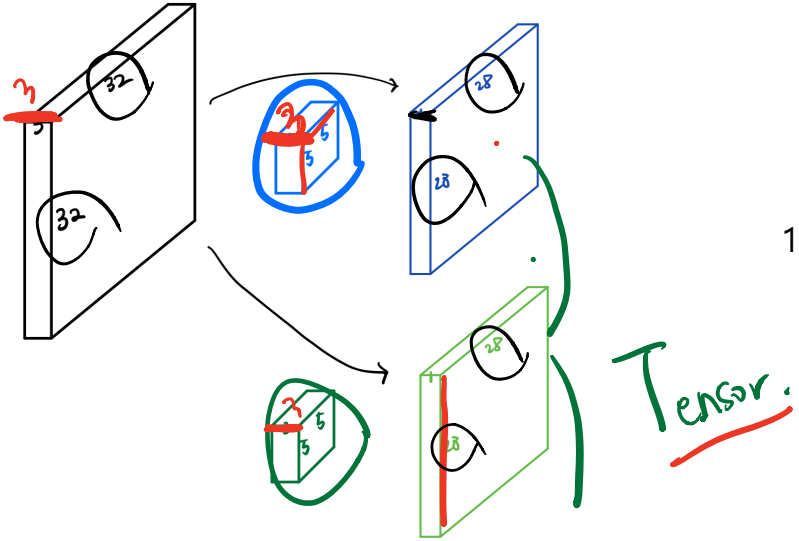
calculate the

sum of elementwise multiplication

This is called, "**Convolution**"

Image from taewan.kim/ppst/cnn

CNN : Convolution



T → M
T

1 Filter = 1 Activation Map

CNN : Convolution

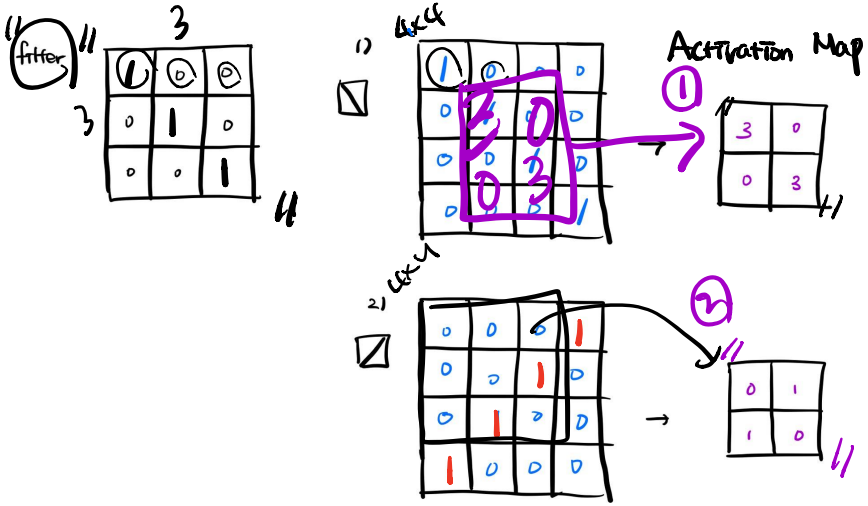
MLP X.

Then, how do filters extract features from the image,

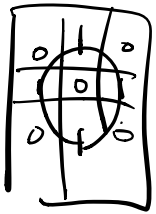
while preserving the spatial structure?

CNN : Convolution

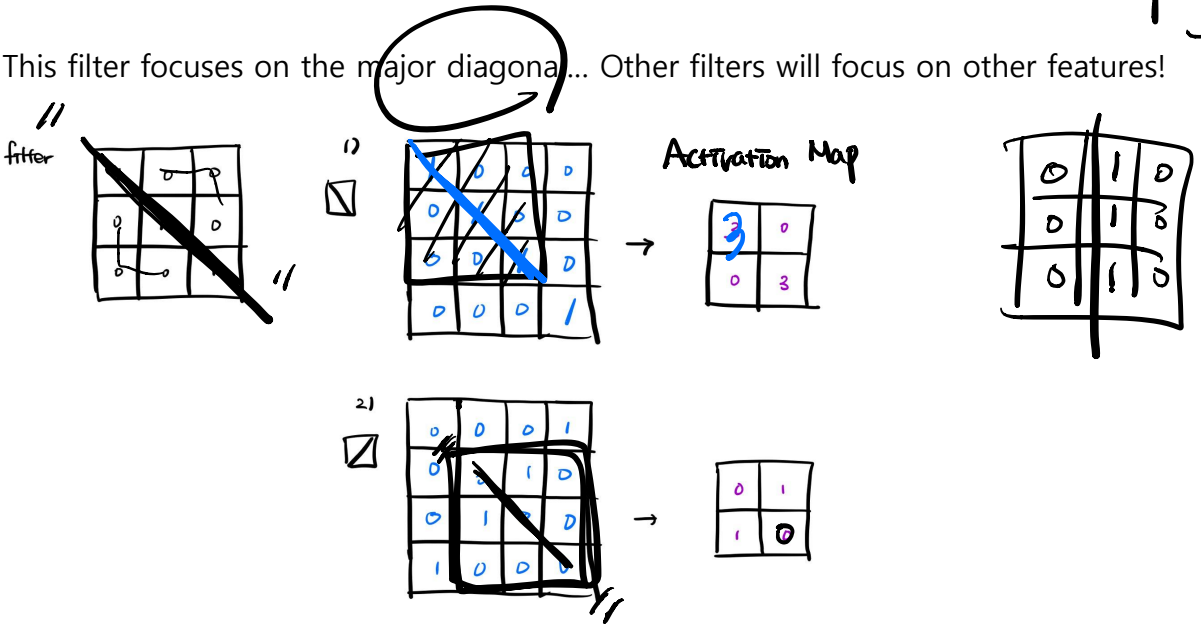
For simplicity, assume White = 0 , Black = 1, and activation = RELU



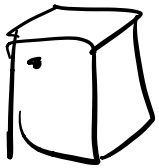
CNN : Convolution



This filter focuses on the major diagonal... Other filters will focus on other features!



CNN : Convolution



Filters extract features from the spatial structure of the tensor

"They are Parameters!!!"

** So, they must be Updated via Backpropagation.

For more detail, read "https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-and-code-714ef1c38199"

I O F S

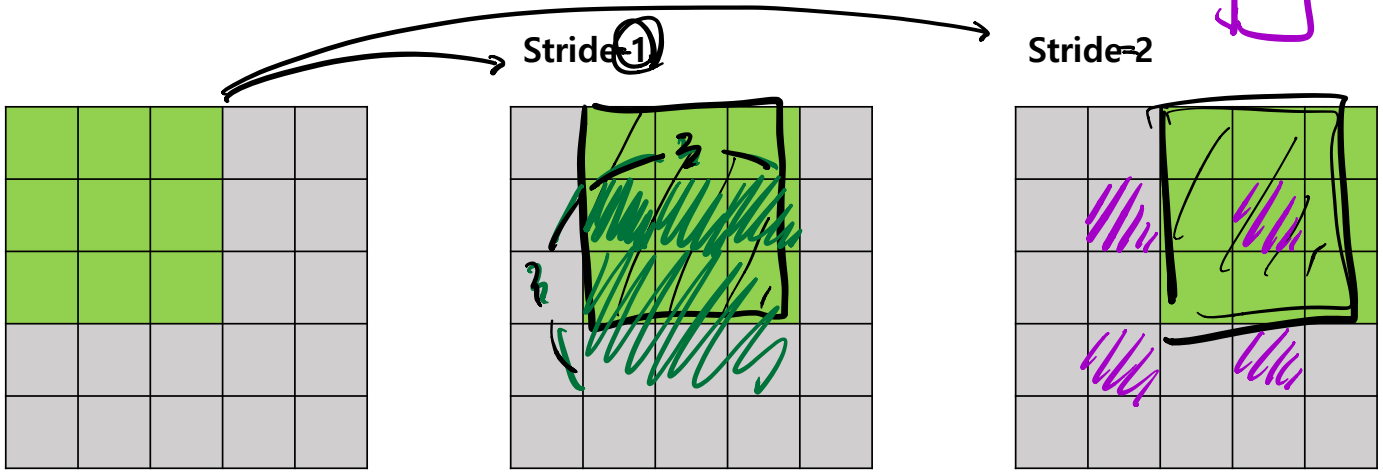
$$O = \frac{I - F}{S} + 1$$

S = 1.
I = 5. F = 3

CNN : Convolution

How much does a filter move per slide? **Stride**

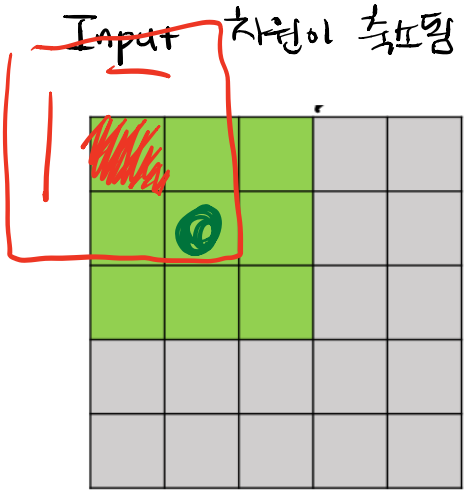
$\frac{5-3}{1} + 1 = 3$



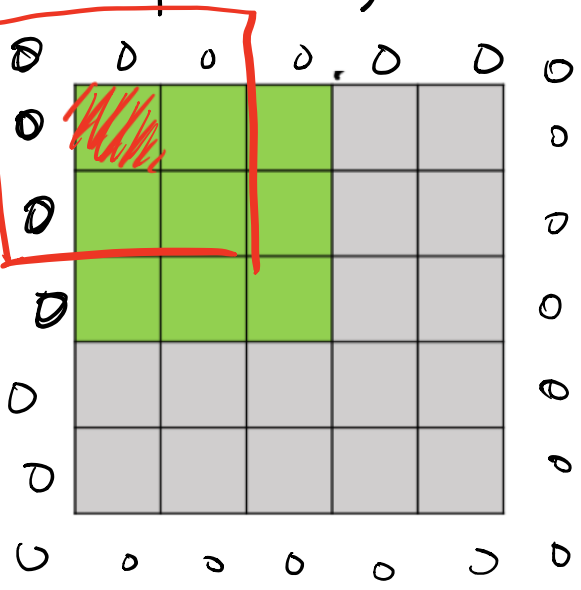
CNN : Padding.

$$O = \frac{I - F}{S} + 1 \leq I - F + 1 < I$$

Input 차원이 축소됨... why ?



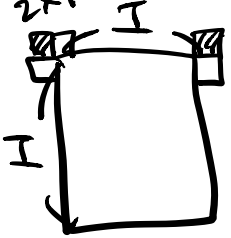
Remedy this via, "Padding"



However,

Data Loss ⟷ "Noise."

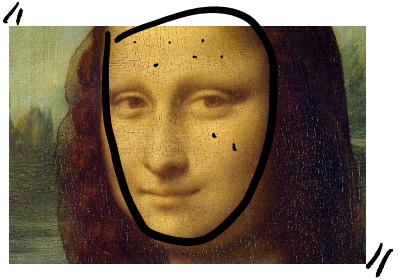
CNN : Convolution



$$O_{W} = \frac{I_{W} + 2Pa_{W} - F_{W}}{S_{W}} + 1$$

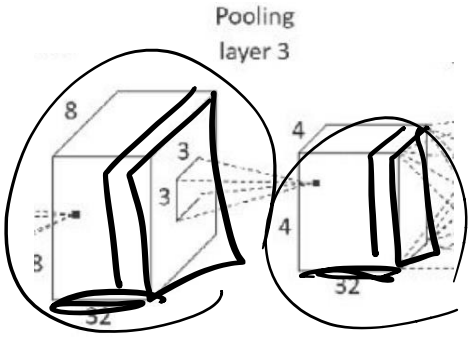
O = output size N = input size F = filter size S = stride Pa = padding size

CNN : Pooling



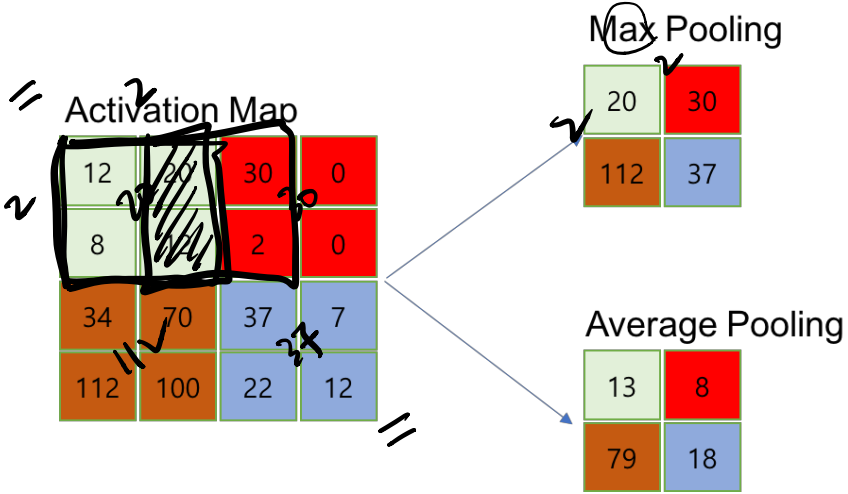
We want to reduce the dimension of the feature,

via **Pooling**



CNN : Pooling

Extract the representative value over a region... No parameter involved!!!



4x4 → 2x2

Image from taewan.kim/ppst/cnn

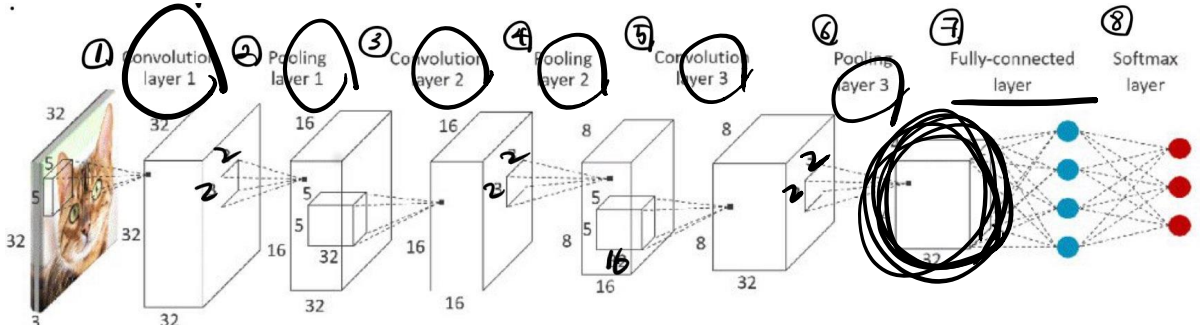
CNN : Pooling

$$O = \frac{I}{Po}$$

O = output size I = input size Po = pool size

** Generally, Pool Size = Pooling Stride

CNN : Fully-Connected Layer

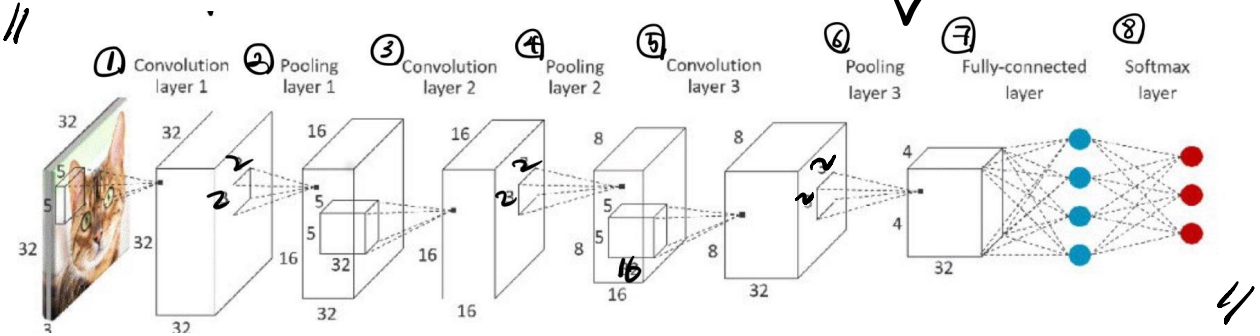


Extract features with **CONV Layers**

Then, reduce dimension with **POOL Layers**

Finally, flatten data and make classification via **FC(Fully-Connected) Layer**

CNN : Architecture



Extract features with **CONV Layers**

Then, reduce dimension with **POOL Layers**

Finally, flatten data and make classification via **FC(Fully-Connected) Layer**

CNN : Architecture

Does it use less parameters than the MLP?

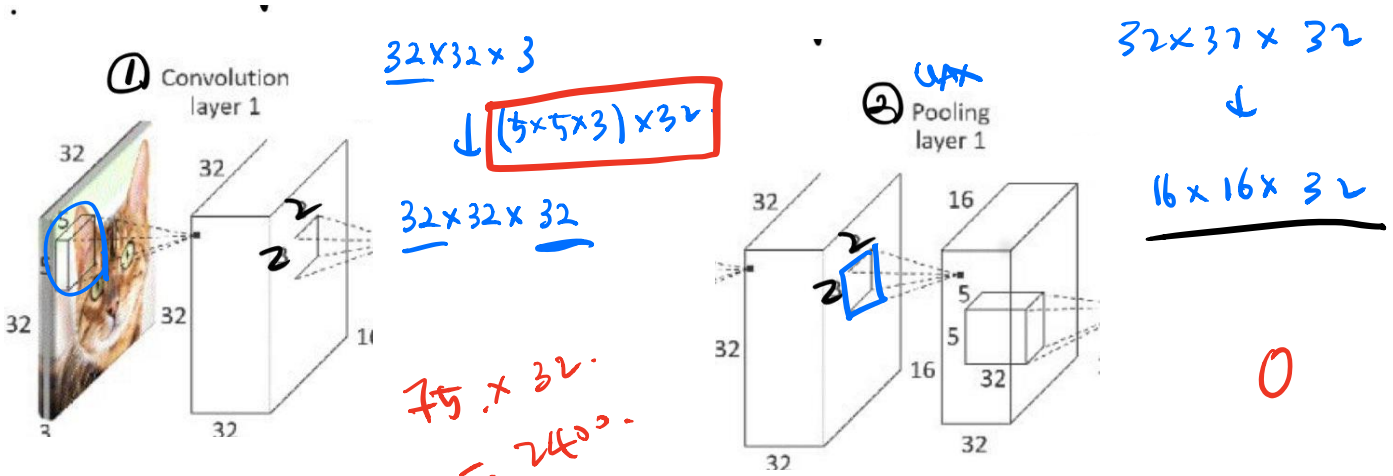
Let’s follow the CNN Architecture and figure it out!!!

CNN : Architecture

CONV1

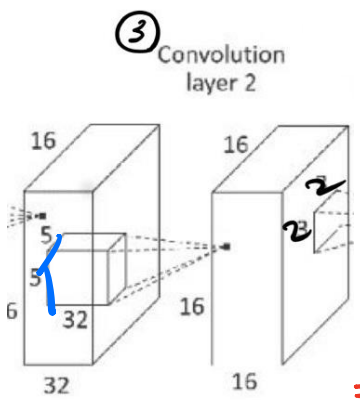
$\frac{32 + 2\times2 - 5}{1} + 1 = 32$

POOL1



CNN : Architecture

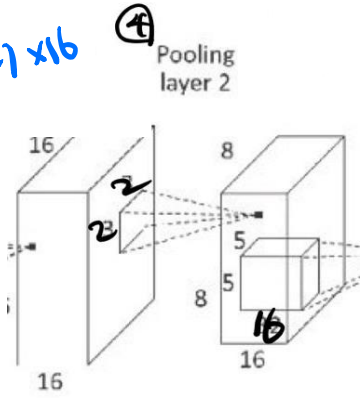
CONV2



16×16×32
↓ (5×5×32)×16
16×16×16

Param
= 5×5×32×16
= 12800

MAX
POOL2

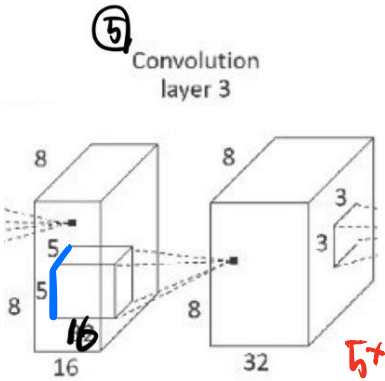


16×16×16
↓
8×8×16

0

CNN : Architecture

CONV3



8x8x16

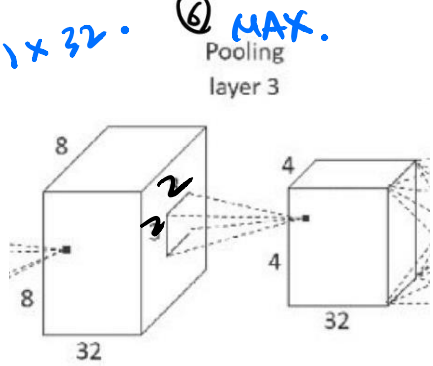
↳ 5x5x16x32.

8x8x3 ✓

5x5x16x32

=12800.

POOL3



8x8x32

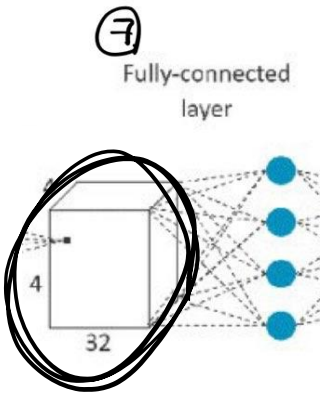
↳

4x4x32.

0

CNN : Architecture

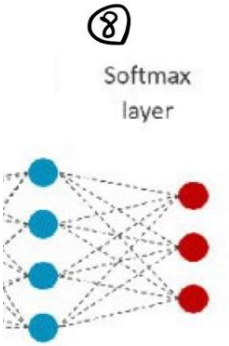
FC



4x4x 32
↓ 512

0

Softmax



512 → 10

5120

CNN : Architecture

$2400 + 0 + 12800 + 0 + 12800 + 0 + 0 + 5120 = 33120.$

Total (33120.) Parameters were Used!

CNN : Architecture

However, if we used MLP instead, assuming CONV + POOL as one Hidden Layer,

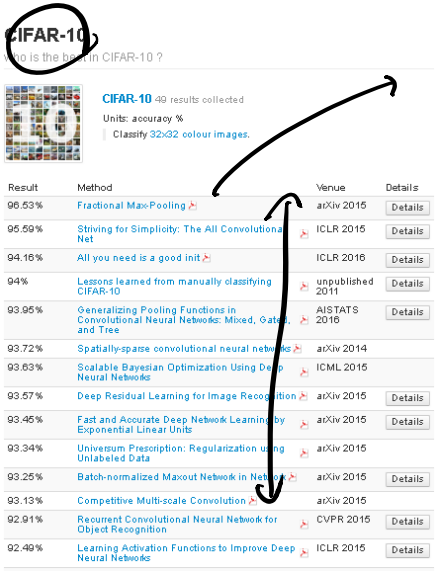
32 * 32 * 3 → 16 * 16 * 32 → 8 * 8 * 16 → 4 * 4 * 32 → 10

Total (35107840) Parameters were Used!

CNN : Architecture

CNN uses much less Parameters than the MLP!!!

CNN : Performance



CIFAR-10

Who is the best in CIFAR-10 ?

CIFAR-10 49 results collected

Units: accuracy %

Classify 32x32 colour images.

Result	Method	Venue	Details
96.53%	Fractional Max-Pooling	arXiv 2015	Details
95.59%	Striving for Simplicity: The All Convolutional Net	ICLR 2015	Details
94.16%	All you need is a good init	ICLR 2016	Details
94%	Lessons learned from manually classifying CIFAR-10	unpublished 2011	Details
93.95%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
93.72%	Spatially-sparse convolutional neural networks	arXiv 2014	
93.63%	Scalable Bayesian Optimization Using Deep Neural Networks	ICML 2015	
93.57%	Deep Residual Learning for Image Recognition	arXiv 2015	Details
93.45%	Fast and Accurate Deep Network Learning by Exponential Linear Units	arXiv 2015	Details
93.34%	Universum Prescription: Regularization using Unlabeled Data	arXiv 2015	
93.25%	Batch-normalized Maxout Network in Network	arXiv 2015	Details
93.13%	Competitive Multi-scale Convolution	arXiv 2015	
92.91%	Recurrent Convolutional Neural Network for Object Recognition	CVPR 2015	Details
92.49%	Learning Activation Functions to Improve Deep Neural Networks	ICLR 2015	Details

96.53%

80%..

Classification Rankings :
https://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html

CNN : Performance

50%.

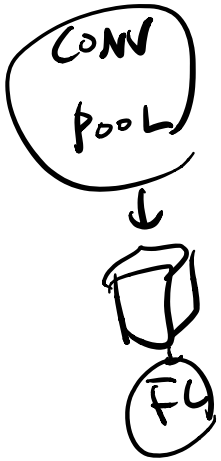
Review

MLP

We want a model that,

1. Uses fewer parameters than MLPs

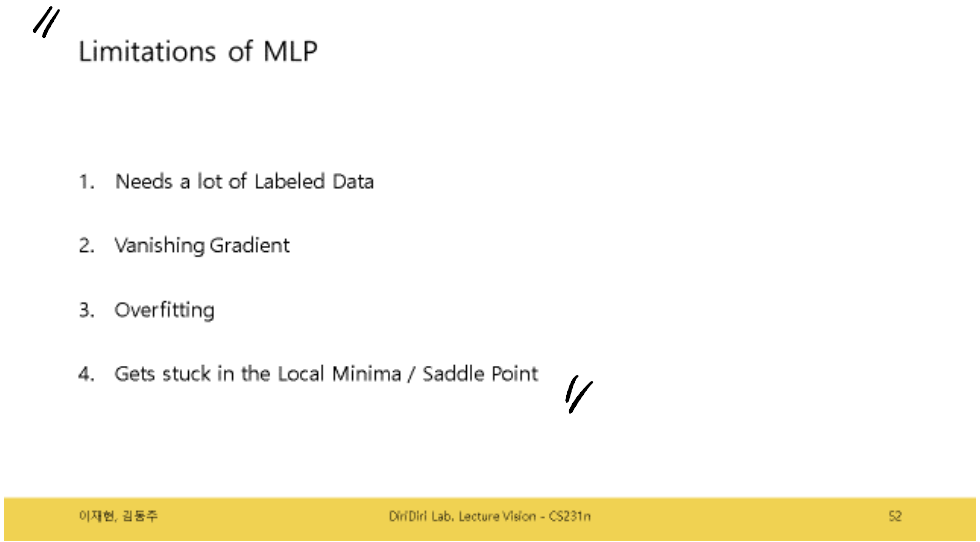
2. Preserves the spatial information



CNN

Convolutional Neural Network to the rescue!!!

Preview on Next Lecture(s)



Preview on Next Lecture(s)

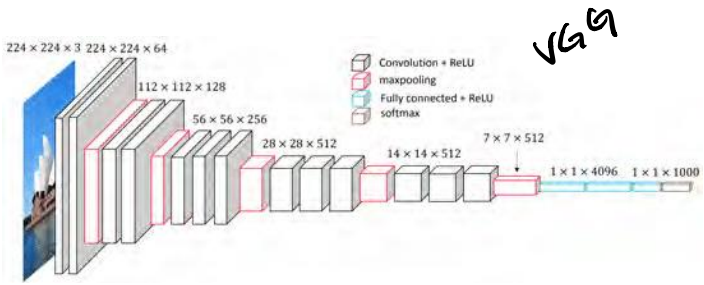


Image from researchgate.net

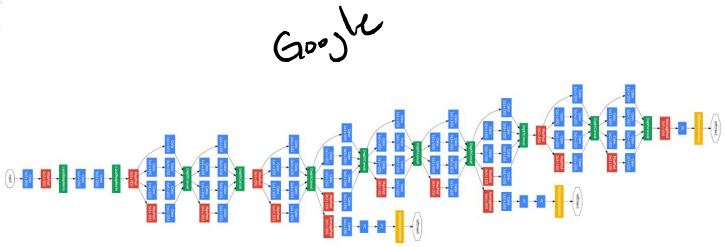


Image from medium.com