ML·DL, AI 지식. ~ SVM, NN.

Lecture 12: Image Classification and Object Recognition

[AIX7021] Computer Vision

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

Tentative schedule

※ 시험범위: 전범위 ※

week	topic	date
1	Introduction and Basics	09.01
2	Image Process I	09.08
3	Image Process II	09.15
4	(휴강)	09.22
5	Feature Detection and Matching I	09.29
6	Feature Detection and Matching II	10.06
7	Clustering and Segmentation I & II	10.13 & 10.16
8	**Mid-Term Exam**	**10.20**
9	Mid-Term Solution	10.27
10	Dimensional Reduction I	11.03
11	Dimensional Reduction II & Robust Fitting and Matching I	11.10
12	Robust Fitting and Matching II	11.17
13	Image Classification & Object Recognition	11.24
14	Motion and Tracking	12.01
15	**Final exam**	12.08

대면

Overview

Object recognition: verification - is it a taxi?

→ classification

Object recognition: detection - are there cars?

True / False 결과.

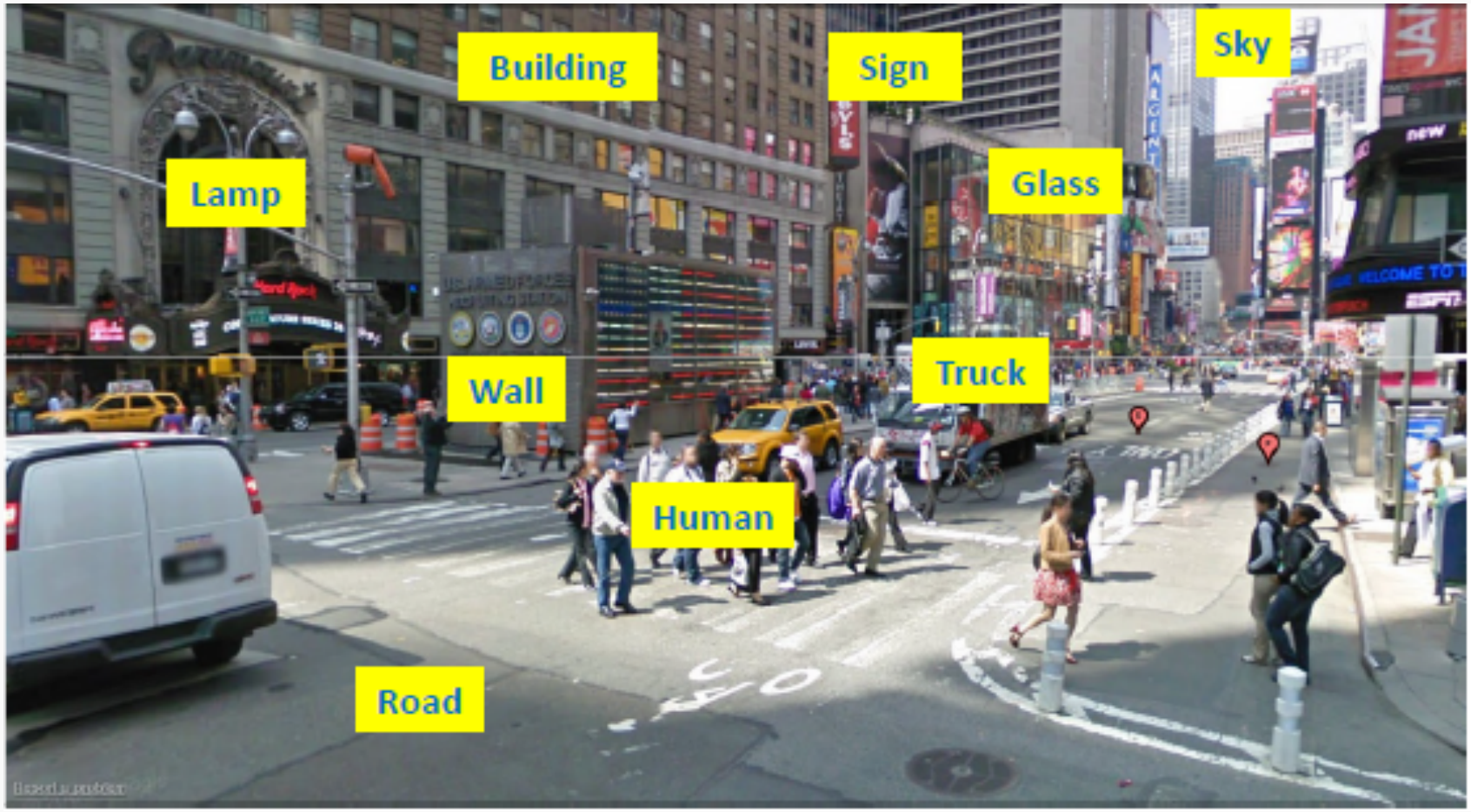
Object recognition: identification - is it a sign of ...?

verif, detec, iden, classify, reco → 맥락에 따른 통계.

↳ 한단계더 높은 인식 측정.

Object recognition: object categorization

이 전체 이미지 안에 들어있는 object들을 인식.



Object recognition: scene categorization

Image classification

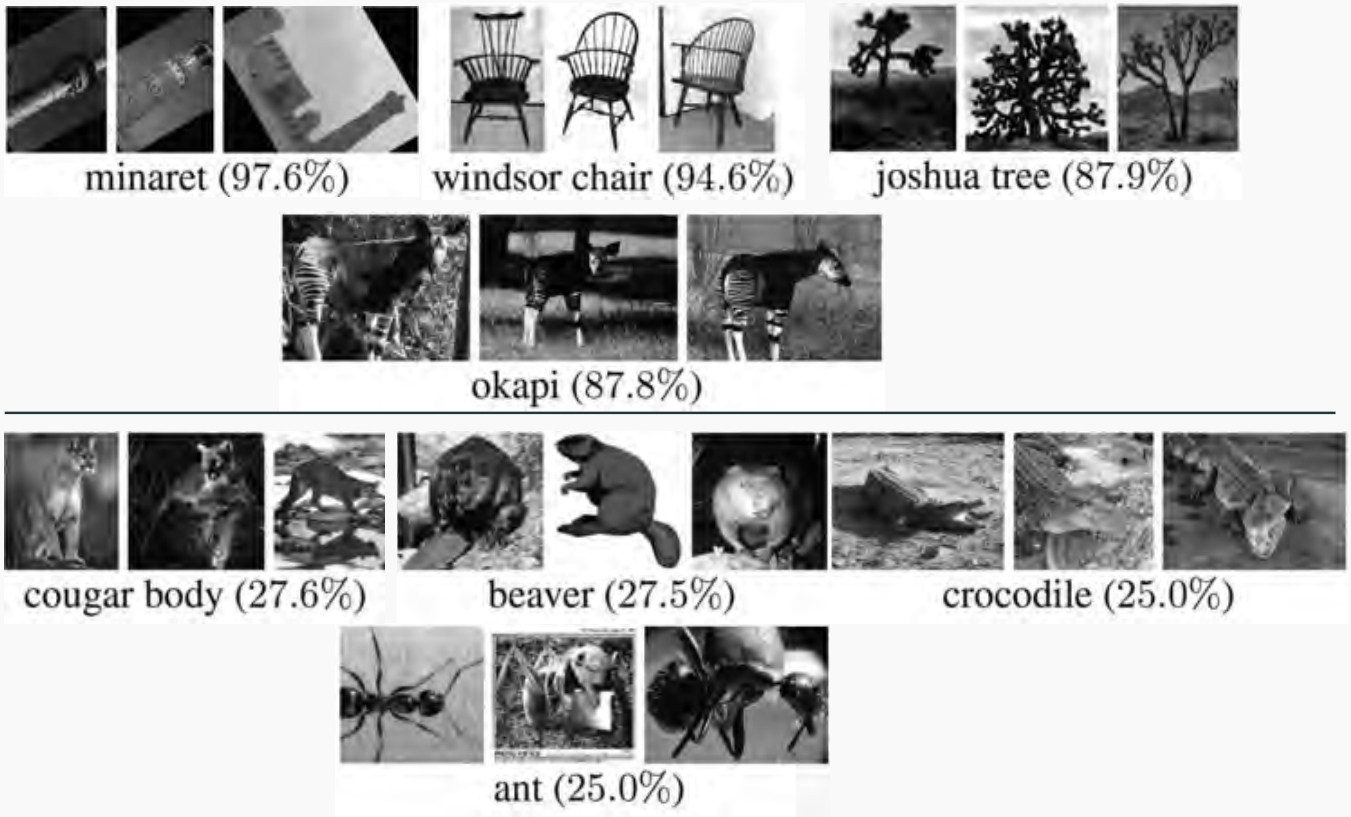


FIGURE 16.10: The spatial pyramid kernel is capable of complex image classification tasks. Here we show some examples of categories from the Caltech 101 collection on which the method does well (**top row**) and poorly (**bottom row**). The number is the percentage of images of that class classified correctly. Caltech 101 is a set of images of 101 categories of objects; one must classify test images into this set of categories (Section 16.3.2). *This figure was originally published as Figure 5 of "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," by S. Lazebnik, C. Schmid, and J. Ponce, Proc. IEEE CVPR 2006, © IEEE 2006.*

Scene categorization

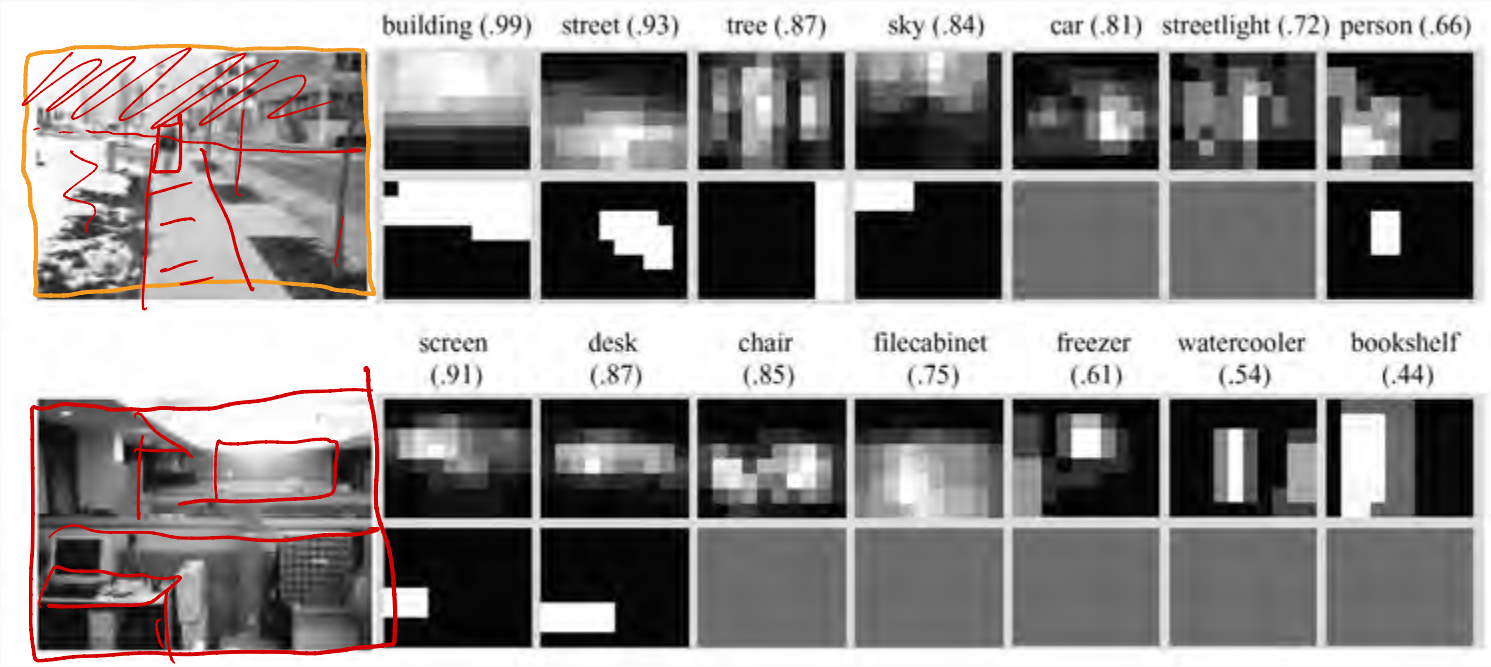
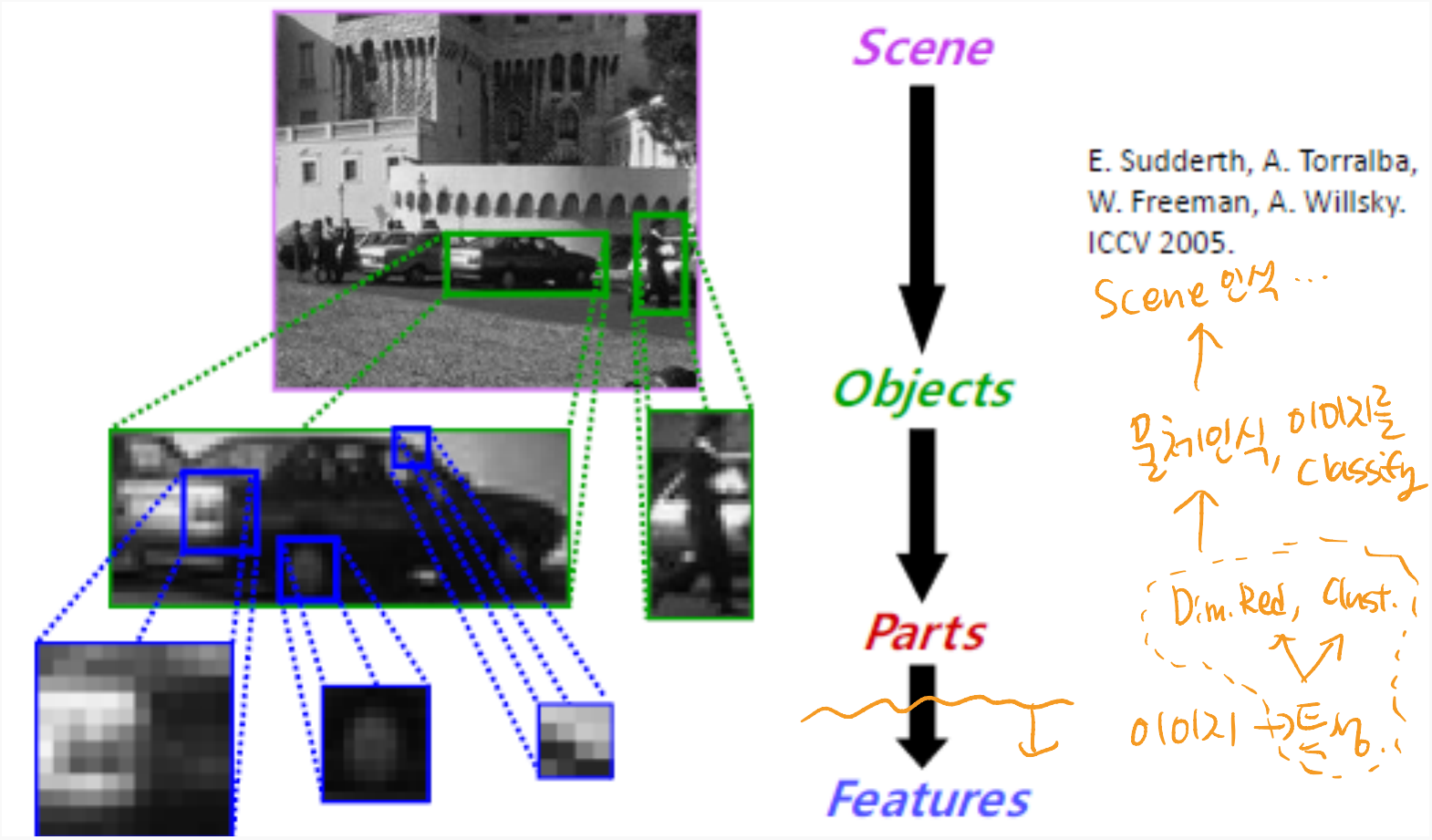
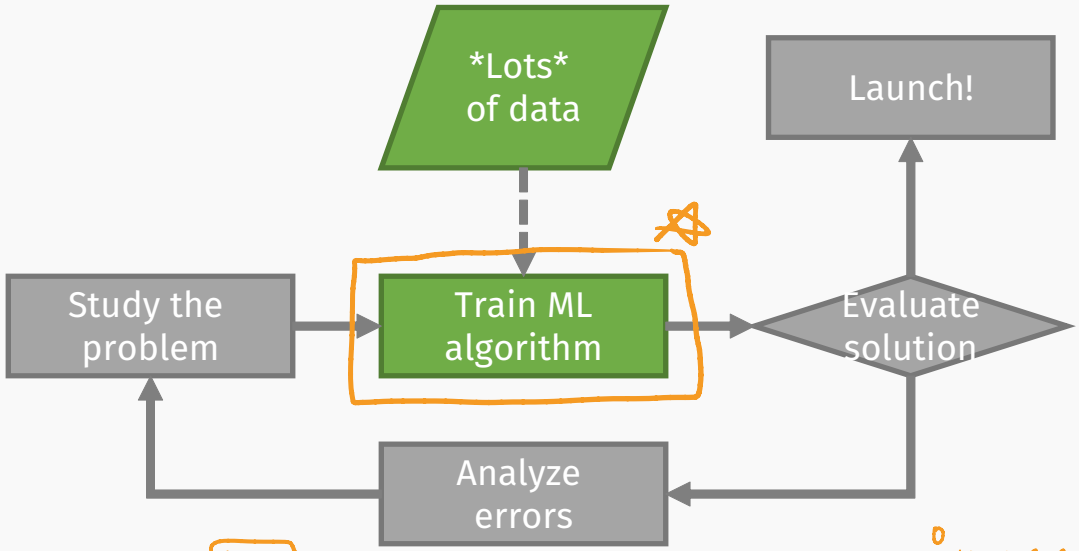


FIGURE 16.4: Scenes are important, because knowing the type of scene shown in an image gives us some information about the objects that are present. For example, street's are typically at the bottom center of street scenes. These maps show probabilities of object locations (**top row**, for each image) extracted from scene information for the image to the left; brighter values are higher probabilities. Compare these with the true support of the object (**bottom row**, for each image); notice that, while knowing the scene doesn't guarantee that an object is present, it does suggest where it is likely to be. This could be used to cue object detection processes. *This figure was originally published as Figure 10 of "Context-based vision system for place and object recognition," by A. Torralba, K. Murphy, W.T. Freeman, and M.A. Rubin, Proc. IEEE ICCV 2003, © IEEE 2003.*

Hierarchy: scene, object, part and feature



Machine learning approach



$$\underbrace{y}_{\text{output}} = \underbrace{f}_{\text{prediction function}} \left(\underbrace{x}_{\text{image feature}} \right) \quad (1)$$

- **Training**: given a *training set* of labeled example $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing**: apply f to a *test example* x and output the predicted value $y = f(x)$

Machine learning approach

입력

↓

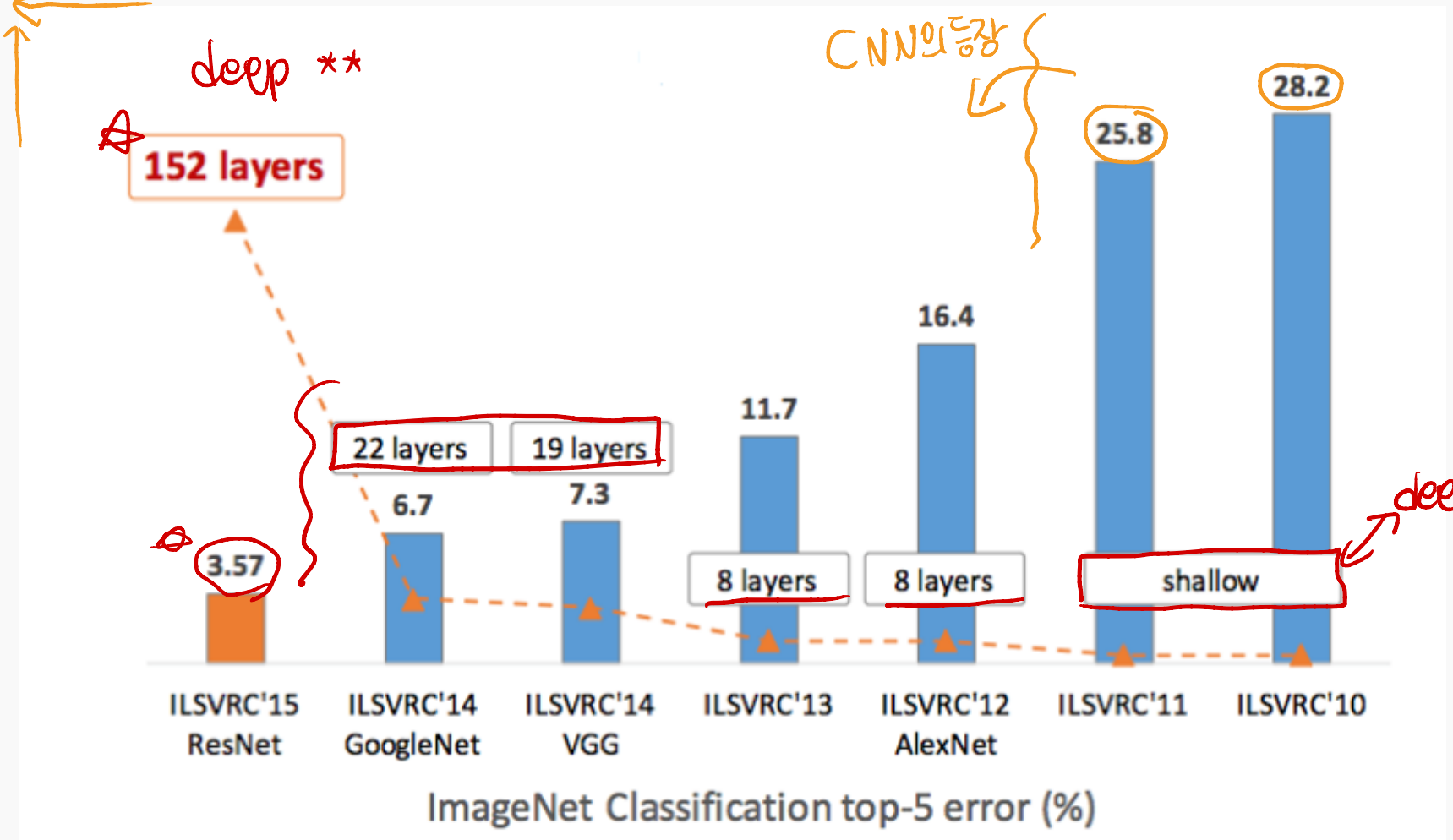
Classifiers → category, discrete value

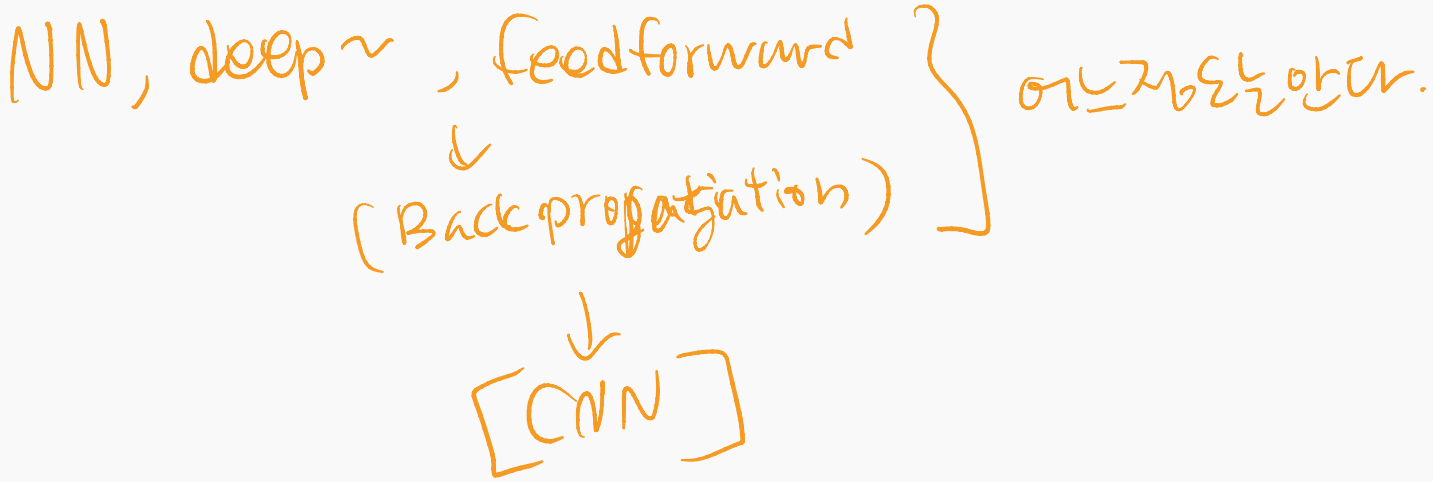
- Nearest neightbor
- Linear discriminant analysis (LDA)
- Neural network, Convolutional Neural Network : 신경망구조,
- Support vector machine (SVM) : ML쪽에서 제일 잘나갔던,
- Boosting
- And many others...

Image Classification

CNN

ImageNet challenge





Convolutional Neural Network (CNN)

The architecture of the visual cortex

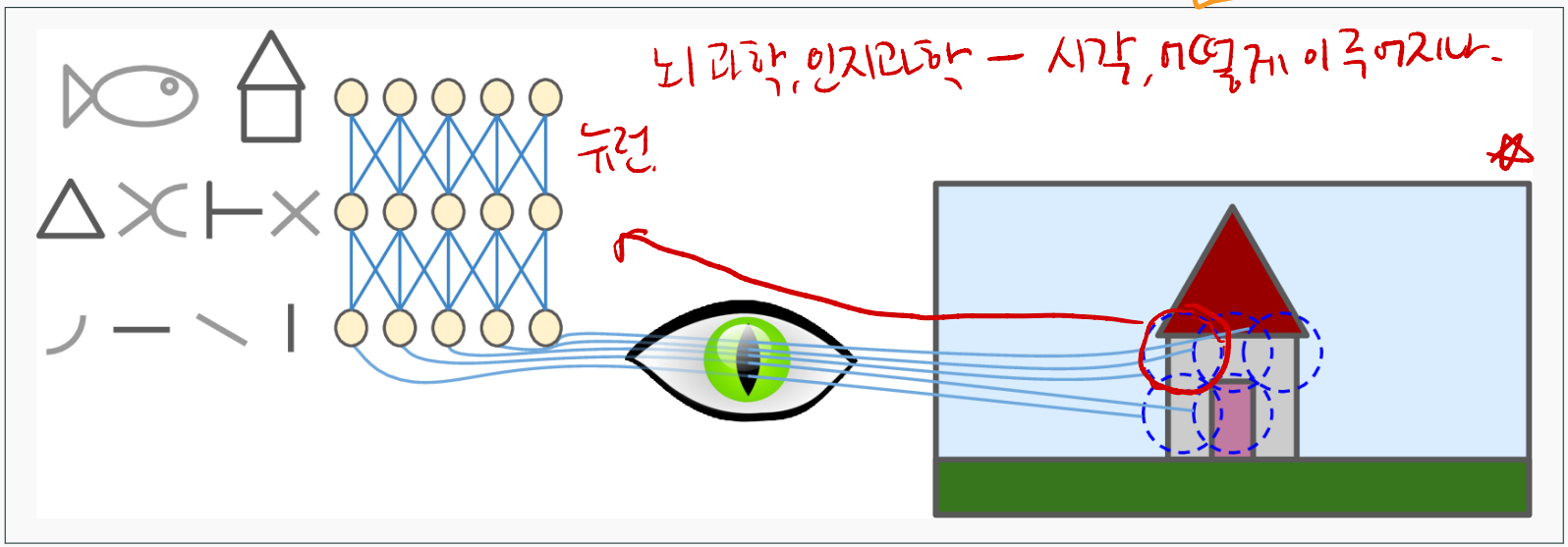


Figure 14-1. Local receptive fields in the visual cortex

In particular, they showed that many neurons in the visual cortex have a small **local receptive field**, meaning they react only to visual stimuli located in a limited region of the visual field (see Figure 14-1, in which the local receptive fields of five neurons are represented by dashed circles). The receptive fields of different neurons may overlap, and together they tile the whole visual field.

Convolutional layer

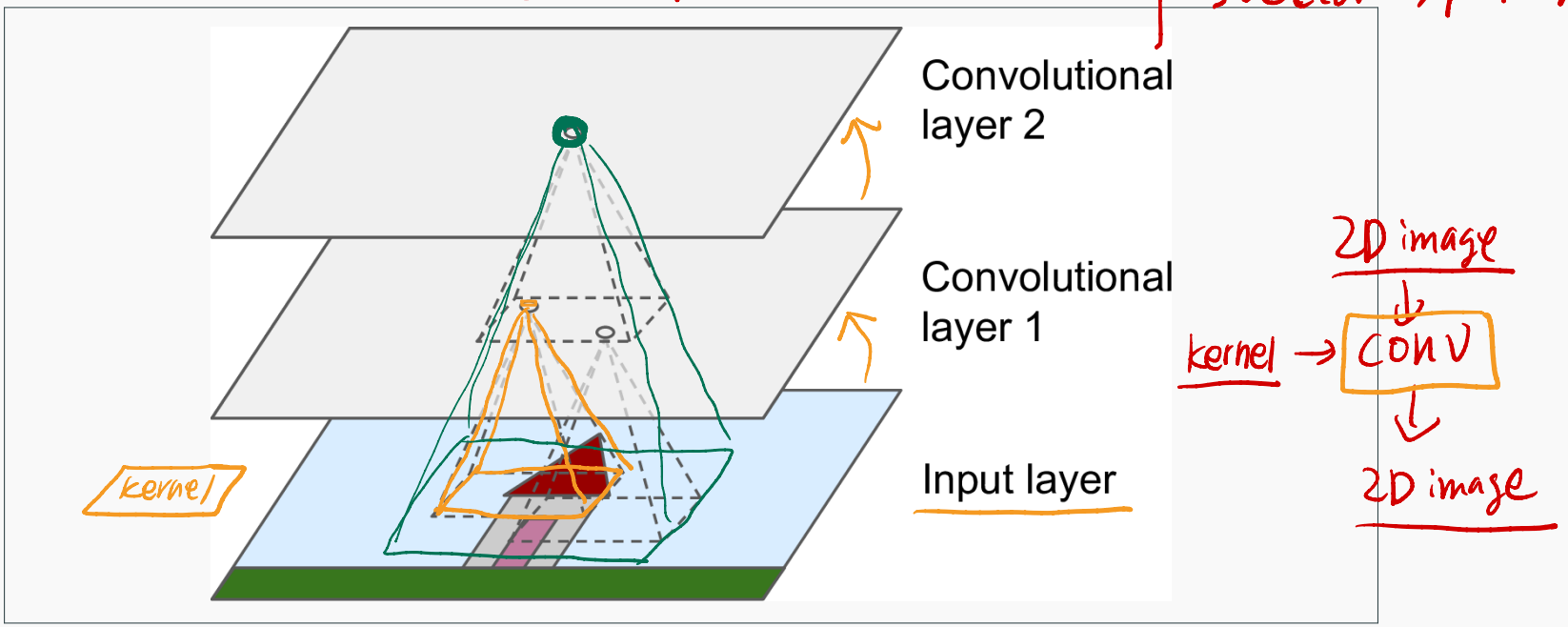


Figure 14-2. CNN layers with rectangular local receptive fields

The most important building block of a CNN is the **convolutional layer**:6 neurons in the first convolutional layer are not connected to every single pixel in the input image (like they were in previous chapters), but only to pixels in their receptive fields (see Figure 14-2). In turn, each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer. This architecture allows the network to concentrate on small low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer, and so on.

Convolutional neural network

- 컨볼루션 신경망(CNN)은 다층 신경망(multi-layer neural networks)의 특별한 형태
- CNN은 시각 피질의 국소 수용역(Local Receptive Field, LRF)과 시신경 세포의 방향 선택적 성격의 발견에 의해서 신경 생물학적인 영감으로 만들어짐
- 내재적이면서 자동적으로 유의미한 특징(feature)들을 추출하는 신경망 구조
- CNN 은 이미지로부터 위상학적 성질들을 추출할 수 있는 피드포워드 (feed-forward) 신경망
- 다른 모든 신경망들처럼 CNN 은 변형된 역전파 알고리즘을 사용하여 학습
- CNN 은 최초한의 전처리 만으로도 픽셀 이미지들로부터 시각적 패턴을 직접 인식할 수 있도록 설계
- 극도로 변화가 많은 패턴들도 인식 가능함 (예: 손글씨)

Convolution

Given a filter kernel $\mathcal{H}$, the convolution of the kernel with image $\mathcal{F}$ is an image $\mathcal{R}$. The (i, j)-th component of $\mathcal{R}$ is given by

$$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{uv}. \tag{20}$$

- **kernel** of the filter: the pattern of weights used for a linear filter
- **convolution**: the process of applying the filter

This operation is called **convolution**

$$R(f) = (h * f) \tag{21}$$

- *commutative*: $(g * h)(x) = (h * g)(x)$
- *associative*: $(f * (g * h)) = ((f * g) * h)$
- *distributive*: $f * (g + h) = f * g + f * h$

Convolution: connections and padding

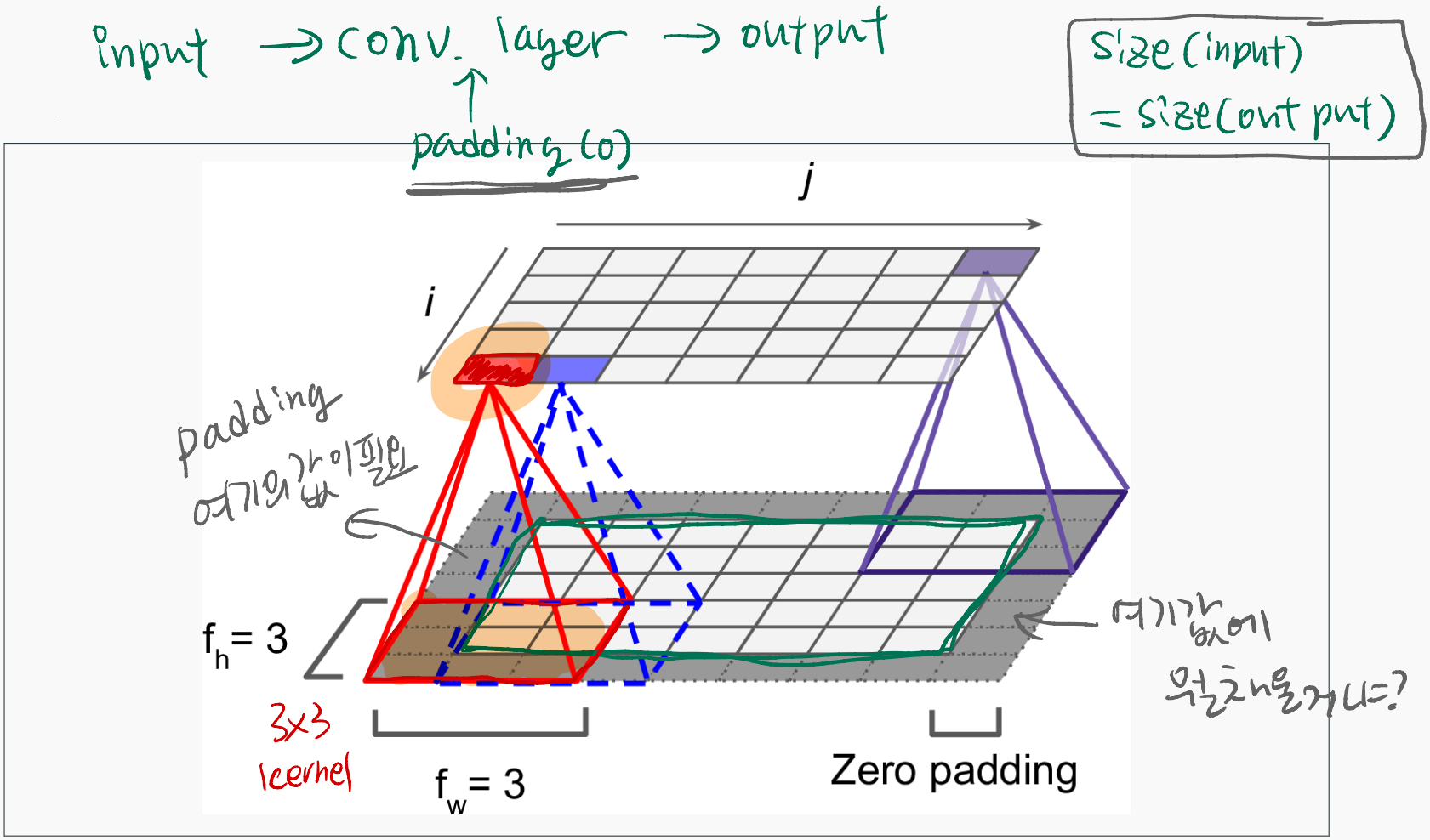


Figure 14-3. Connections between layers and zero padding

Convolution: stride and dimensional reduction

Conv = 바로 옆 / 위 / 아래 픽셀로 움직임.

Stride만큼 건너뛰는거.

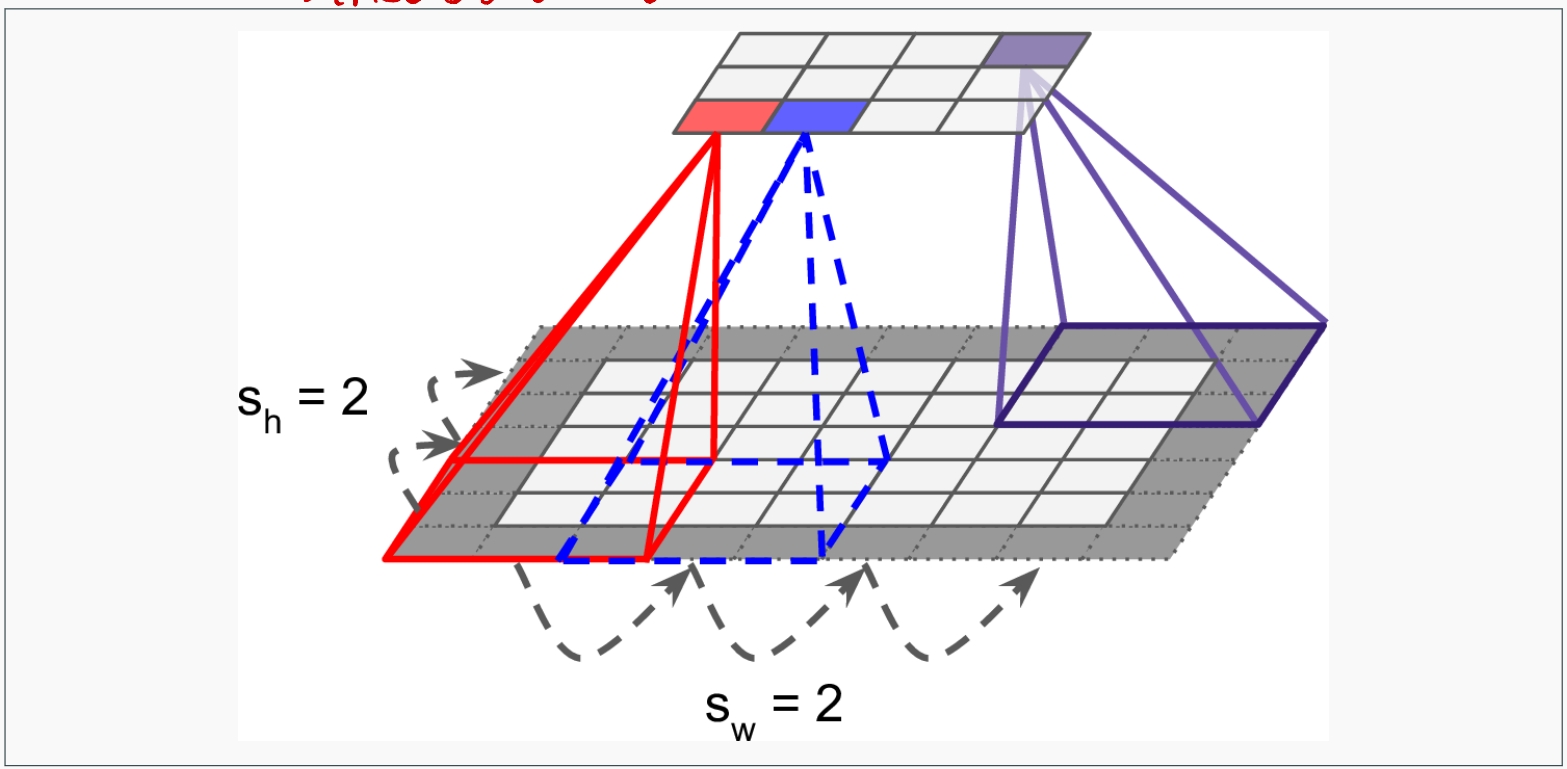


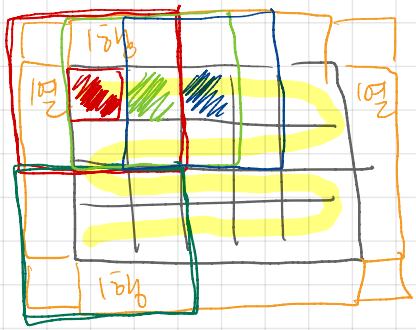
Figure 14-4. Reducing dimensionality using a stride of 2

padding, stride

input → Conv. → output ?

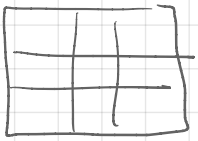
↑

kernel



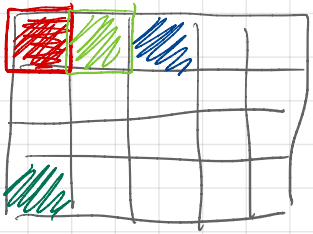
4x5
input

*

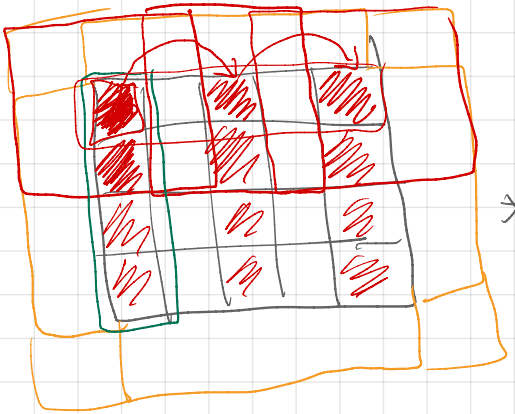


3x3
kernel

=

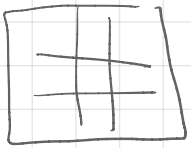


4x5
output (padding)



4x5
input

*

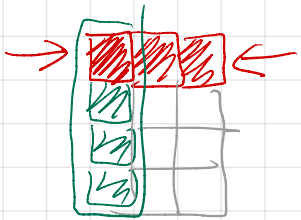


(3x3
kernel)

=

(1x2
stride)

위아래 1칸씩
양옆은 2칸씩.



4x3
output.
(padding 0)

Convolution: example

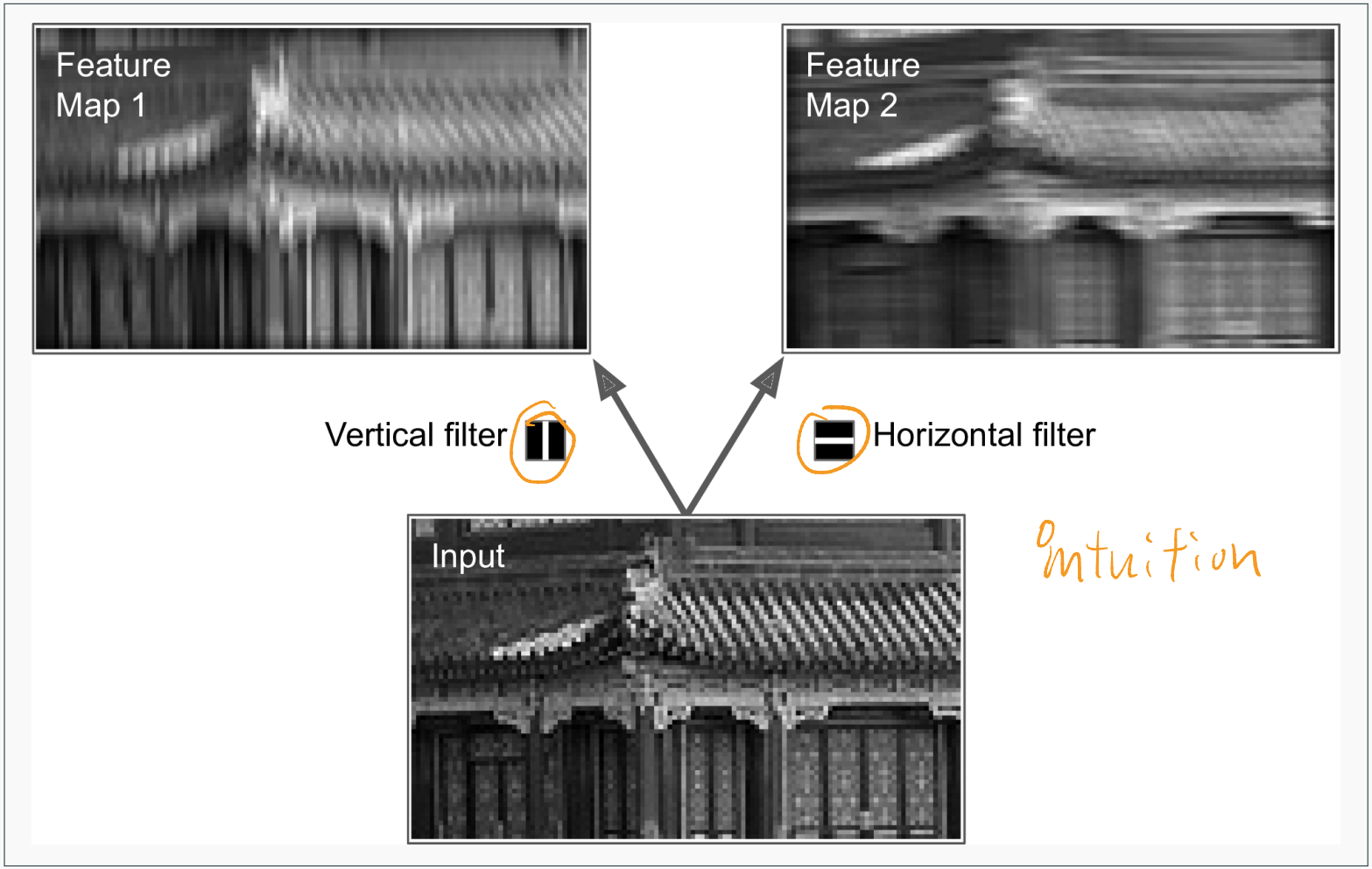
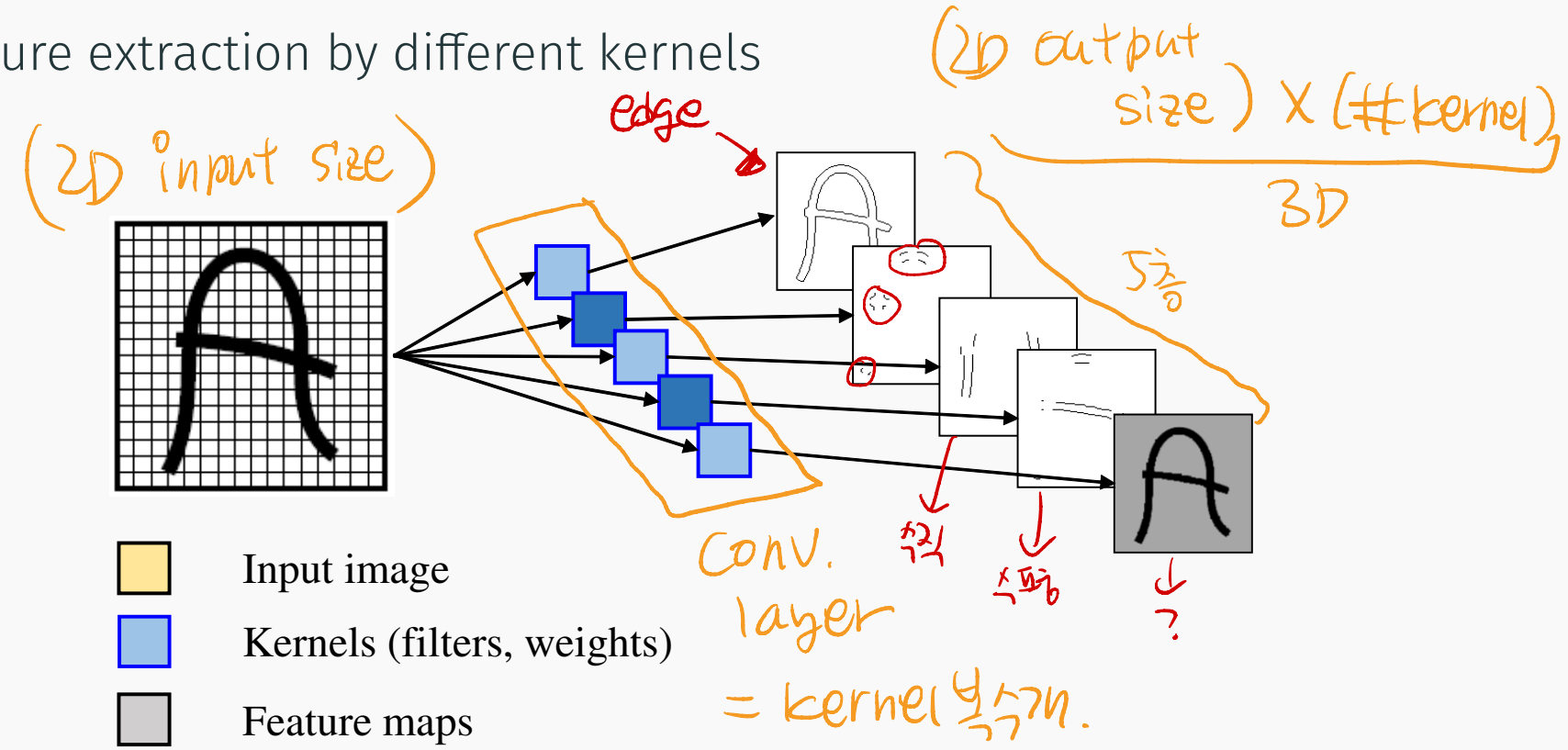


Figure 14-5. Applying two different filters to get two feature maps

Convolution: filter effect

Feature extraction by different kernels



- 컨볼루션은 입력 이미지의 다른 부분들에 존재하는 동일한 특징들을 찾아냄
- Feature map에 있는 뉴런은 template or kernel과 일치되는 경우에 활성화

CNN: convolution layers with multiple feature maps

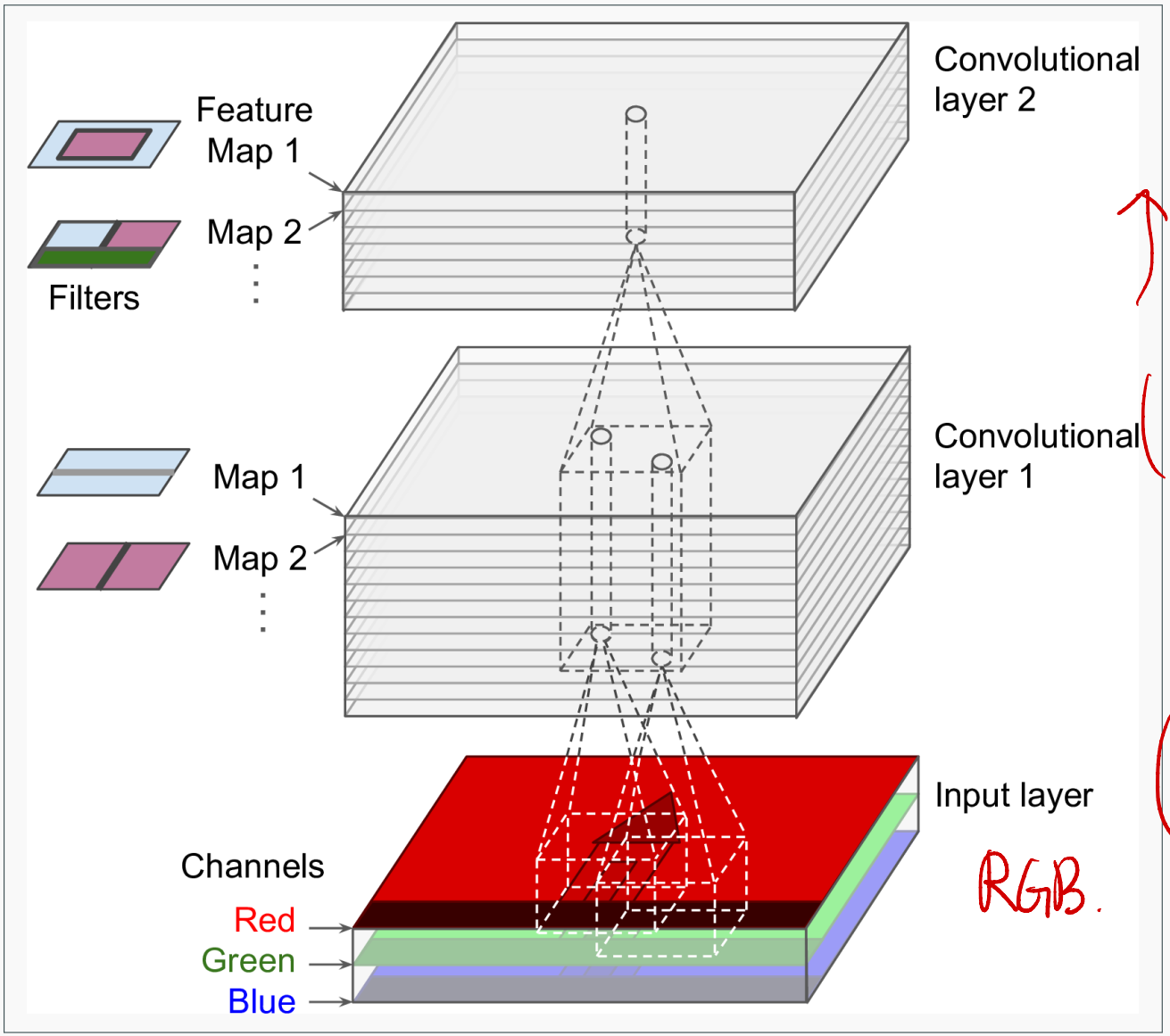
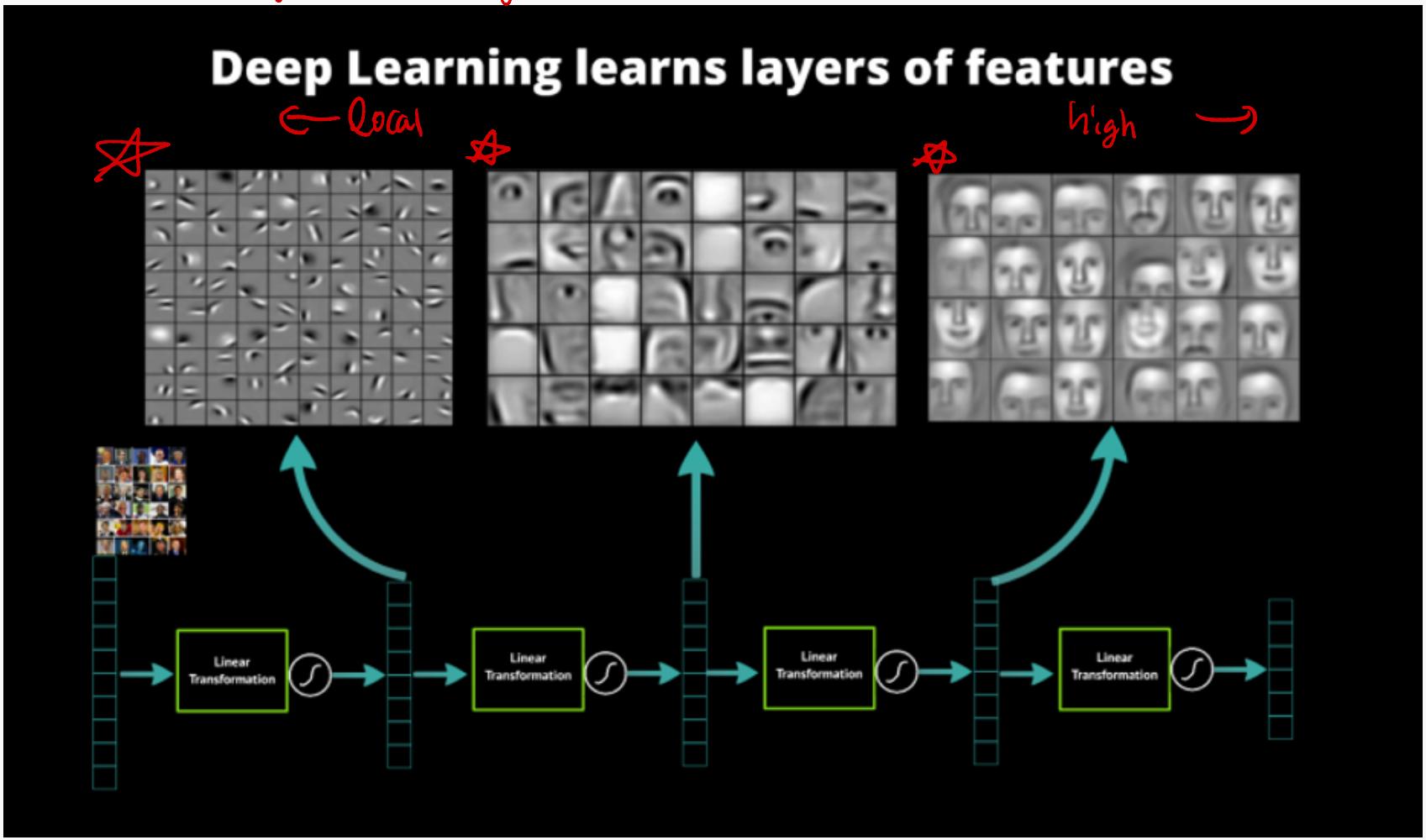


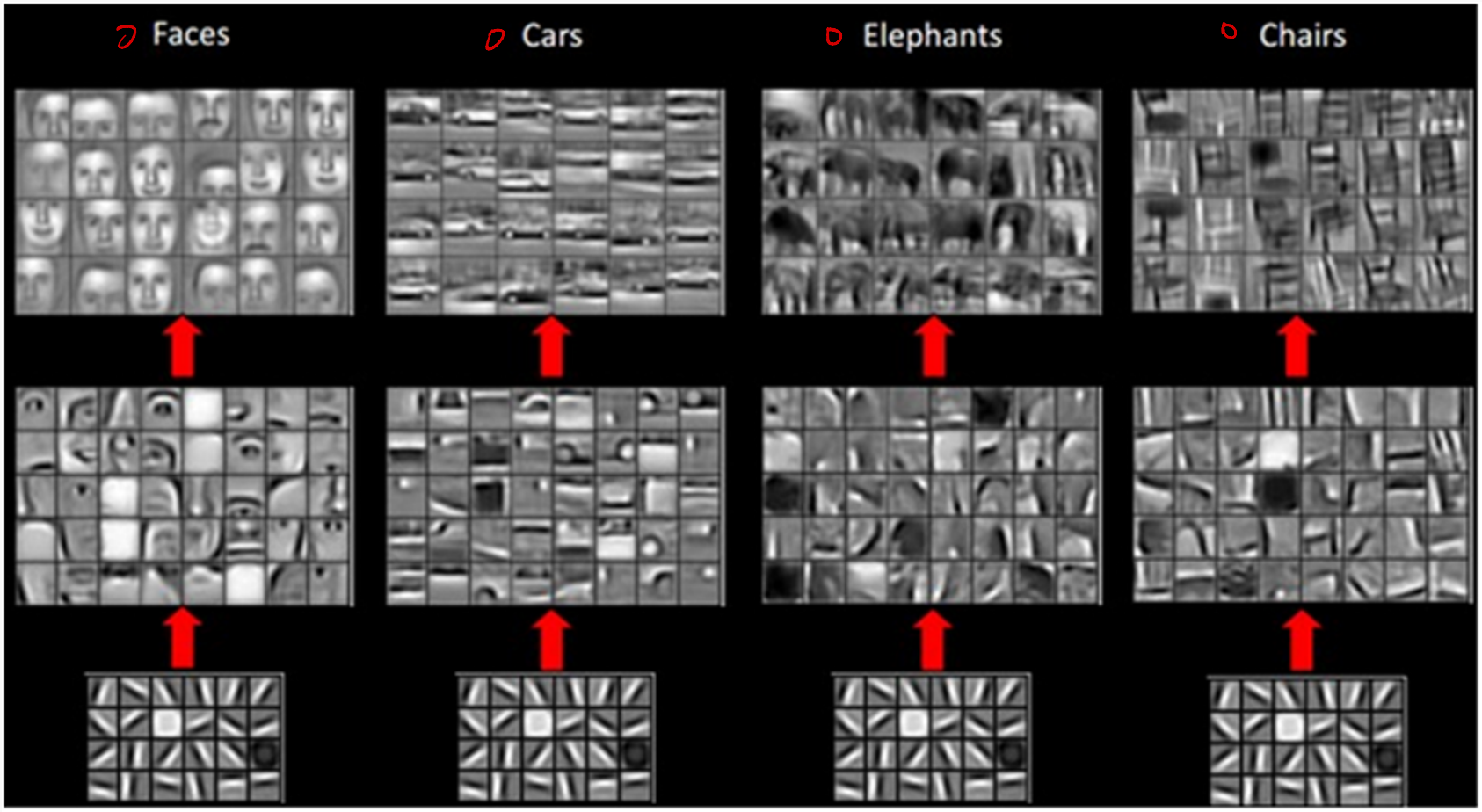
Figure 14-6. Convolution layers with multiple feature maps, and images with three color channels

CNN: hierarchical learning of features

CNN → layer by layer feature 기르다.



CNN: hierarchical learning of features



CNN: dimensional reduction by pooling layer

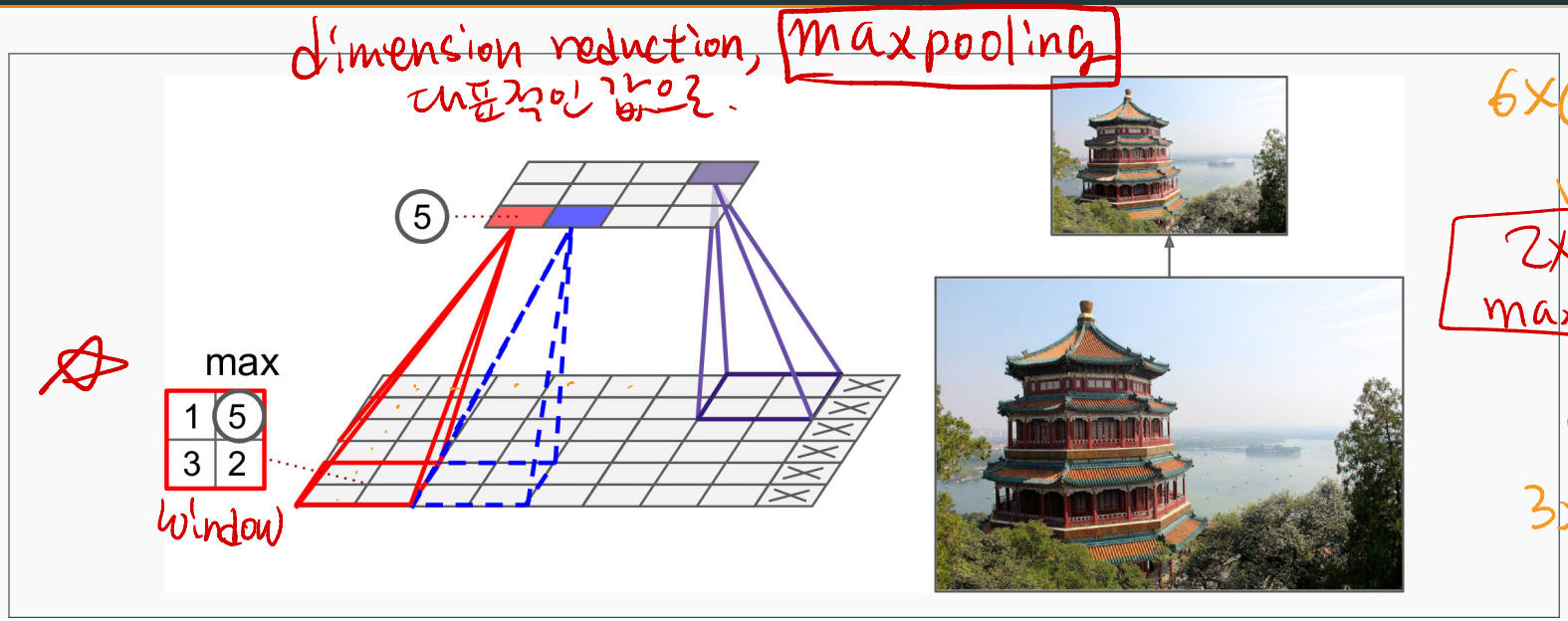


Figure 14-8. Max pooling layer (2 × 2 pooling kernel, stride 2, no padding)

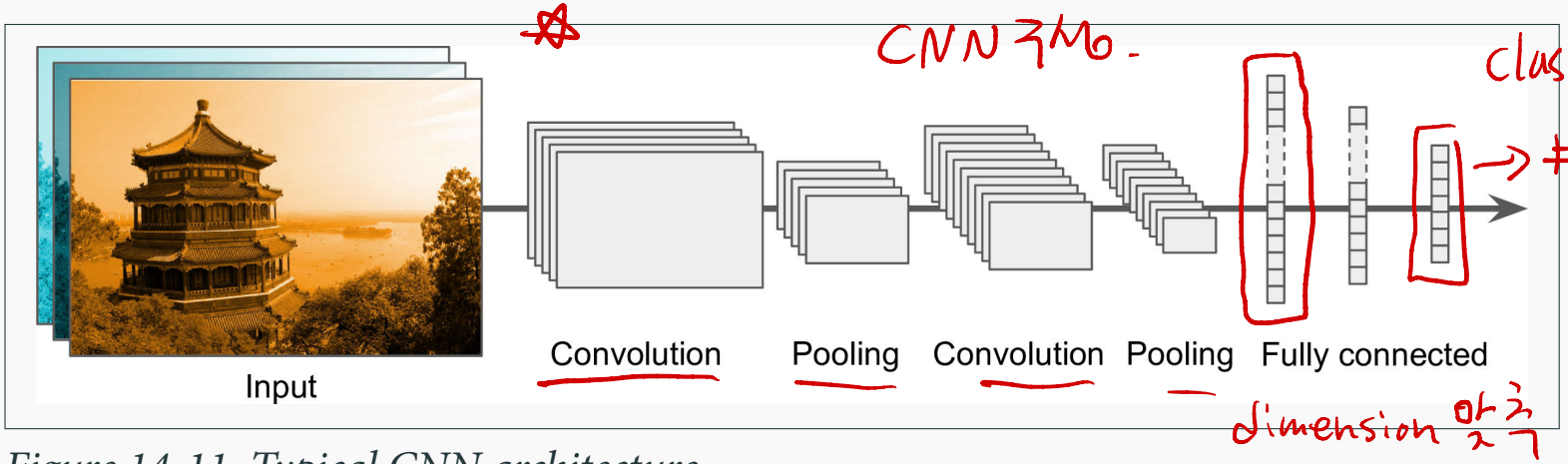


Figure 14-11. Typical CNN architecture

ML → SVM

convex optimization, 'Lagrangian'

'최적화'

objective, cost function

min / max

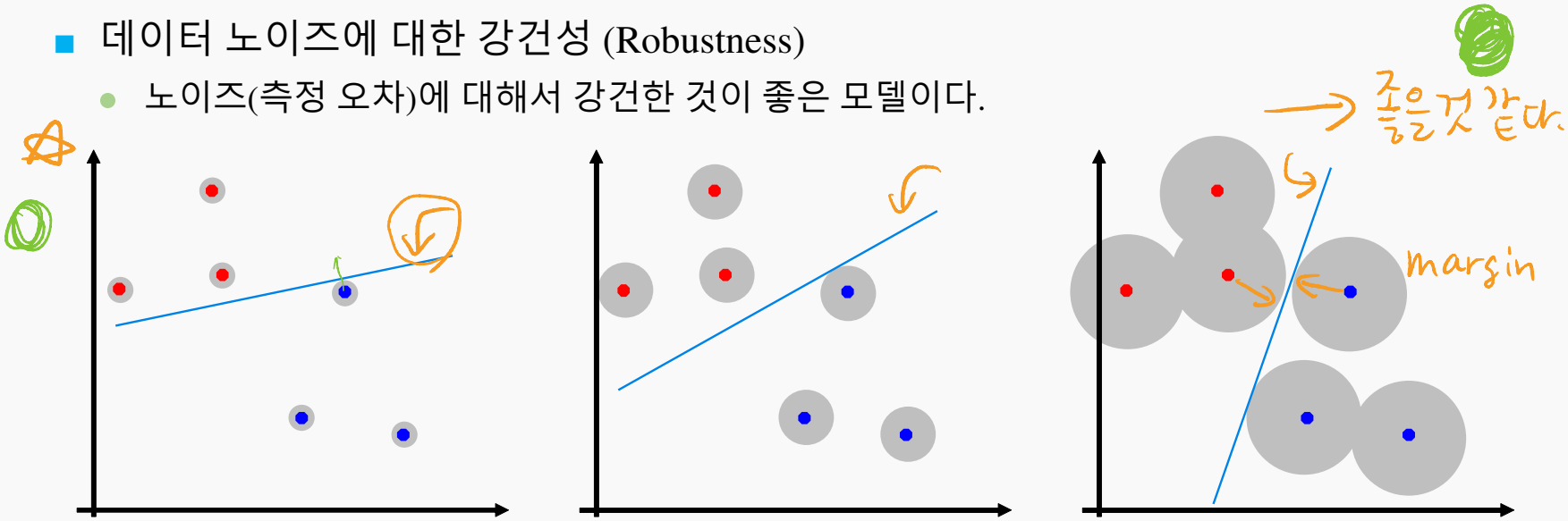
constraint.

Support Vector Machine (SVM):

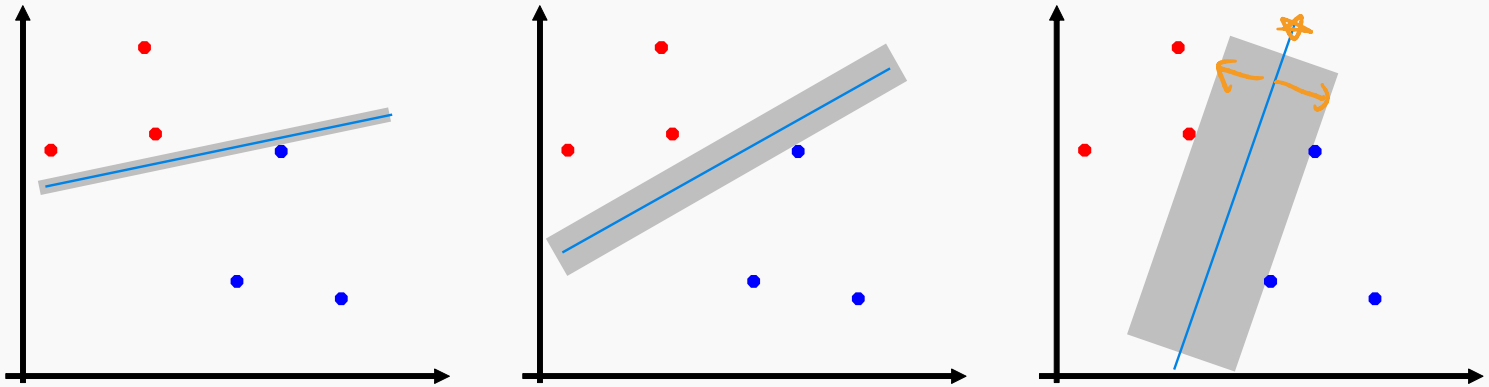
「Maximum Margin Classifier」 최적.

What is a good decision boundary?

- 데이터 노이즈에 대한 강건성 (Robustness)
  - 노이즈(측정 오차)에 대해서 강건한 것이 좋은 모델이다.



- 여유로운 것이 더 강건하다 $\Rightarrow$ 넓은 통로가 좋다 $\Rightarrow$ Large Margin Classification



What is a good decision boundary?

의사 결정은 경계의 데이터(support vectors)에 의해서 결정됨

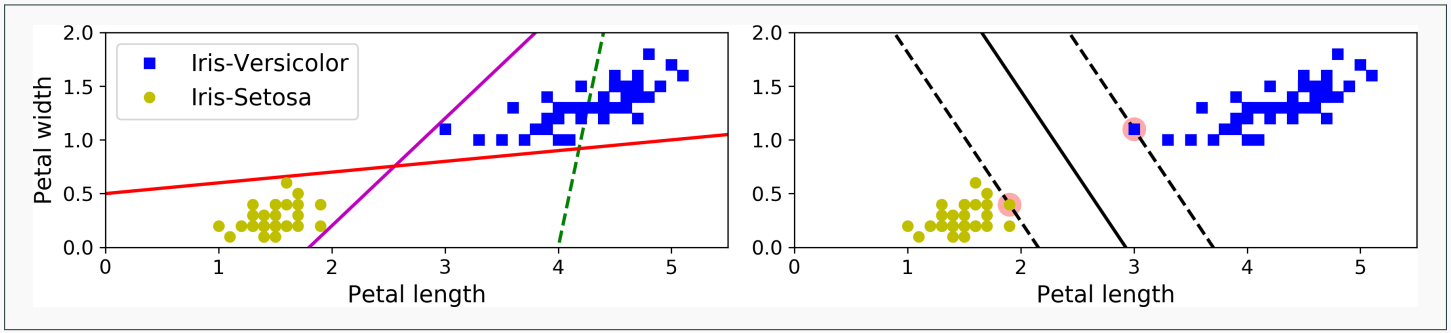


Figure 5-1. Large margin classification

Input feature의 scale에 민감한 support vector machine

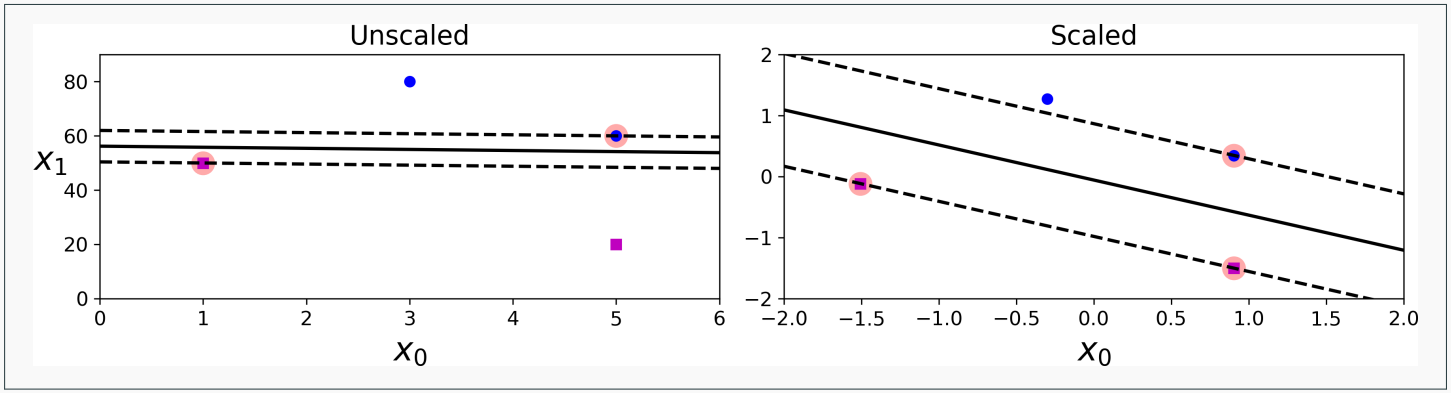
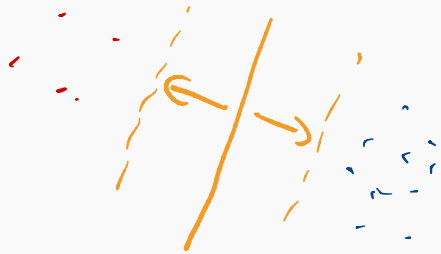


Figure 5-2. Sensitivity to feature scales

Hard margin vs. soft margin



- Hard margin classification (hard-SVM)
  - 모든 데이터들이 margin 밖에 위치하도록 boundary를 설정
  - 데이터가 선형적으로 분리 가능(linearly separable)할 때만 적용 가능
  - outlier에 매우 민감

- Soft margin classification (soft-SVM)
  - margin을 가능한 넓게 하면서도 margin 안쪽으로 들어오는 것을 허용
  - hyperparameter C: 클수록 좁아짐 (엄격), 작을 수록 넓어짐 (위반 허용)

margin의 침범을 덜 허용하겠다 = margin이 좁아짐 → 오른쪽

침범을 꽉꽉 눌러서 = margin도 넓히겠다. → 왼쪽

Hard margin vs. soft margin

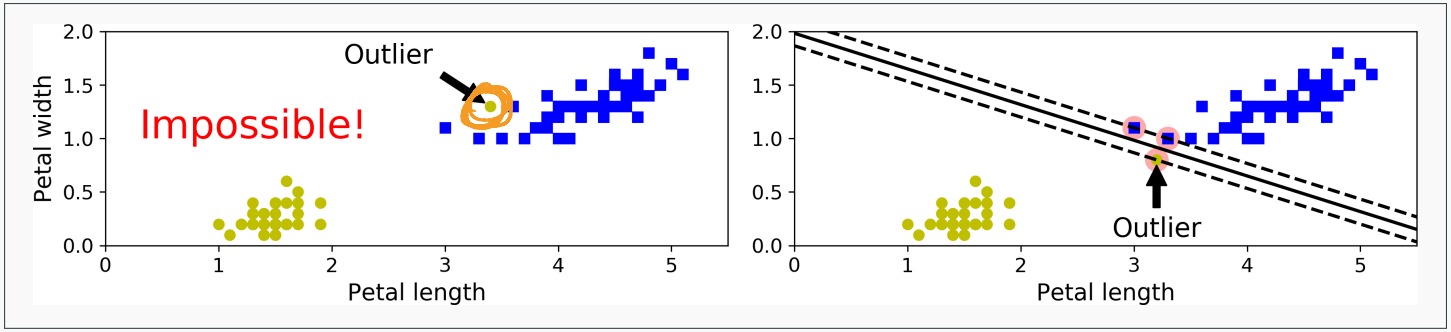


Figure 5-3. Hard margin sensitivity to outliers

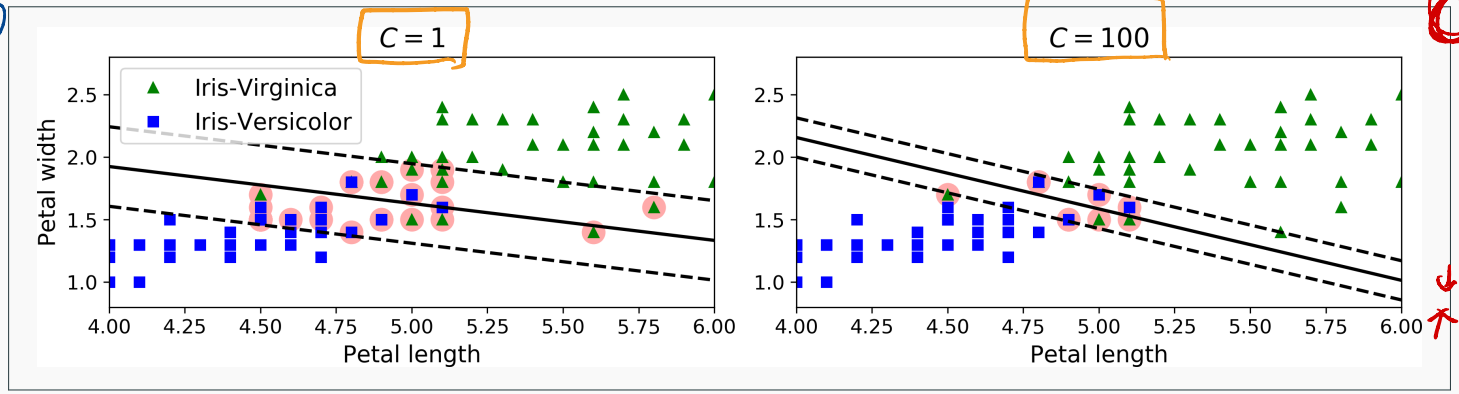


Figure 5-4. Large margin (left) versus fewer margin violations (right)

A brief history of SVM

- SVM은 1992년 Boser, Guyon and Vapnik에 의해서 소개됨
- Statistical Learning Theory에 이론적 바탕을 둔 알고리즘 (Vapnik Chervonenkis)
- 손글씨 숫자 인식에서 뛰어난 성능을 보이면서 널리 쓰이게 됨
- SVM으로 1.1% Test error rate ≈ 신중히 설계된 신경망(e.g., LeNet 4)과 맞먹음
- 실용적으로 우수한 성능 ← 최적화된 결과
- bioinformatics, text, image recognition 등을 포함한 많은 성공 사례
- 강력하고 다재 다능한 머신 러닝 모델
- 선형/비선형 분류 뿐 아니라 회귀, outlier detection 도 수행
- 복잡한 소규모/중규모 데이터셋의 분류에 특히 잘 맞음
- 머신 러닝에서 중요한 기법 중 하나인 Kernel 방법을 사용하는 대표적 알고리즘
- 머신 러닝에서 가장 널리 쓰이는 모델이며 머신 러닝을 하며 반드시 알아야 할 기법 중 하나
- Liblinear libsvm: Scikit-Learn 에서 liblinear 및 libsvm 을 사용하여 구현

Convex Optimization and Duality

Optimization problem in standard form

$$\begin{aligned} &\text{minimize} && f_0(x) && && (32)\\ &\text{subject to} && f_i(x) \leq 0, && i = 1, 2, ..., m && (33)\\ &&& h_i(x) = 0, && i = 1, 2, ..., p && (34) \end{aligned}$$

- $x \in \mathcal{R}^n$ is the optimization variable
- $f_0 : \mathcal{R}^n \to \mathcal{R}$ is the objective or cost function
- $f_i : \mathcal{R}^n \to \mathcal{R}, i = 1, 2, ..., m$ are the inequality constraint functions
- $h_i : \mathcal{R}^n \to \mathcal{R}$ are the equality constraint functions

Convex optimization problem

Standard form convex optimization problem

$$\begin{aligned} \text{minimize} \quad & f_0(x) && && (35)\\ \text{subject to} \quad & f_i(x) \le 0, && i = 1, 2, ..., m && (36)\\ & a_i^T x = b_i, && i = 1, 2, ..., p && (37) \end{aligned}$$

- $f_0, f_1, ..., f_m$ are convex
- equality constraints are affine

Often written as

$$\begin{aligned} \text{minimize} \quad & f_0(x) && && (38)\\ \text{subject to} \quad & f_i(x) \le 0, && i = 1, 2, ..., m && (39)\\ & Ax = b && && (40) \end{aligned}$$

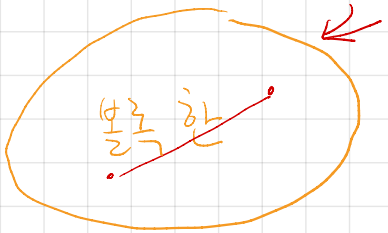
Important property: feasible set of a convex optimization problem is convex

「Convex set, S」

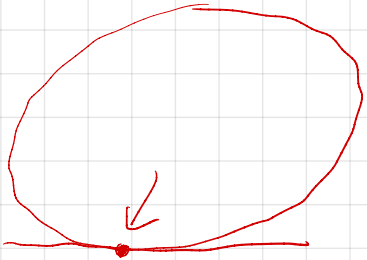
if $x, y \in S$, then $\lambda x + (1-\lambda) y \in S$ $0 \le \lambda \le 1$ 를 만족하면

$\lambda x + (1-\lambda) y, \ 0 \le \lambda \le 1$

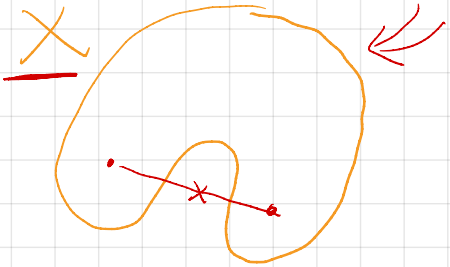
$\Rightarrow$



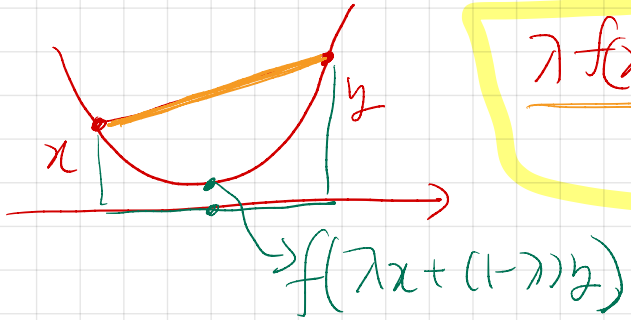
Convex → optimization 하기 좋다. (global)



X



Convex function, f,

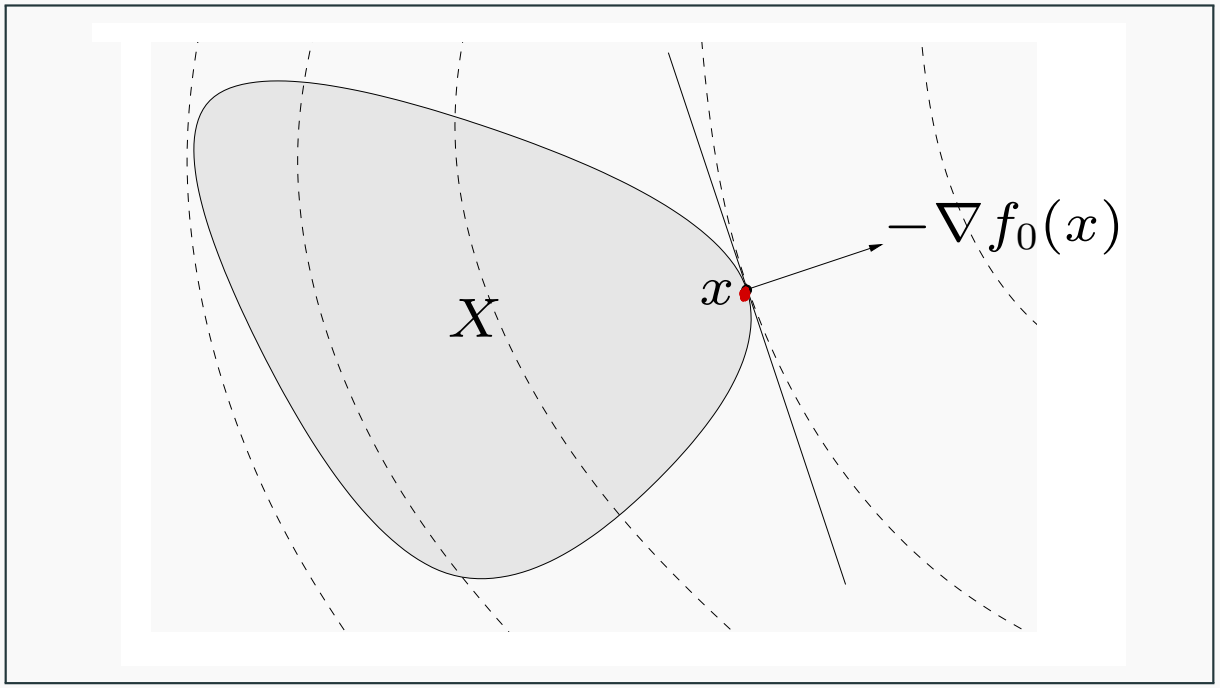


$$\lambda f(x) + (1-\lambda) f(y) \geq f(\lambda x + (1-\lambda) y)$$

Optimality criterion for differentiable f_0

x is optimal if and only if it is feasible and

$$\nabla f_0(x)^T(y-x) \geq 0 \quad \text{for all feasible } y \tag{41}$$



if nonzero, $\nabla f_0(x)$ defines a supporting hyperplane to feasible set X at x

Lagrangian

standard form problem

Primal problem

$$\begin{aligned} \text{minimize} \quad & f_0(x) && (42)\\ \text{subject to} \quad & f_i(x) \le 0, \qquad i = 1, 2, ..., m && (43)\\ & h_i(x) = 0, \qquad i = 1, 2, ..., p && (44) \end{aligned}$$

variable $x \in \mathcal{R}^n$, domain $\mathcal{D}$, optimal value p^*

Lagrangian: $L : \mathcal{R}^n \times \mathcal{R}^m \times \mathcal{R}^p \to \mathcal{R}$ with $\mathbf{dom}\, L = \mathcal{D} \times \mathcal{R}^m \times \mathcal{R}^p$

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \qquad (45)$$

(x: 변수, λ: inequality, ν: equality)

L을 정의하는 방법.

- weighted sum of objective and constraint functions
- λ_i is Lagrange multiplier associated with $f_i(x) \le 0$
- ν_i is Lagrange multiplier associated with $h_i(x) = 0$

Lagrange dual function

Lagrange dual function: $g : \mathcal{R}^m \times \mathcal{R}^p \to \mathcal{R}$

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \qquad (46)$$

$$= \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right) \qquad (47)$$

$=$ g is concave, can be $-\infty$ for some λ, ν

lower bound property: if $\lambda \geq 0$, then $g(\lambda, \nu) \leq p^*$

proof: if $\tilde{x}$ is feasible and $\lambda \geq 0$, then

$$f_0(\tilde{x}) \geq L(\tilde{x}, \lambda, \nu) \geq \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = g(\lambda, \nu) \qquad (48)$$

minimizing over all feasible $\tilde{x}$ gives $p^* \geq g(\lambda, \nu)$

The dual problem

Lagrange dual problem

primal problem → $\mathcal{L}$ → $g(\lambda, \nu)$ → dual problem

$$\text{maximize} \quad g(\lambda, \nu) \tag{49}$$

$$\text{subject to} \quad \lambda \geq 0 \tag{50}$$

f_0, f_i, h_i

inequality const.

Lagrangian multiplier.

- finds best lower bound on p^*, obtained from Lagrange dual function
- a convex optimization problem; optimal value denoted d^*
- λ, ν are dual feasible if $\lambda \geq 0$, $(\lambda, \nu) \in \operatorname{dom} g$
- often simplified by making implicit constraint $(\lambda, \nu) \in \operatorname{dom} g$ explicit

Weak and strong duality

- weak duality: $d^* \leq p^*$
  - always holds (for convex and nonconvex problems)
  - can be used to find nontrivial lower bounds for difficult problems
- strong duality: $d^* = p^*$
  - does not hold in general
  - holds for convex problems*
  - conditions that guarantee strong duality in convex problems are called constraint qualifications

d^* 가 lower bound 다.

d^* : dual prob. solution
p^* : primal prob. solution

* ← SVM

Karush-Kuhn-Tucker (KKT) conditions

the following four conditions are called KKT conditions (for a problem with differentiable f_i, h_i)

1. primal constraints: $f_i(x) \leq 0$ i = 1, ..., m, $h_i(x) = 0$, i = 1, ..., p
2. dual constraints: $\lambda \geq 0$
3. complementary slackness: $\lambda_i f_i(x) = 0, i = 1, ..., m$
4. gradient of Lagrangian with respect to x vanishes:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0 \tag{52}$$

if strong duality holds and x, λ, ν are optimal, then they must satisfy the KKT conditions

조성진 악습 등
+ best polonaise '영웅' 별칭: 조성진 비전

Support Vector Machine (SVM): Theory

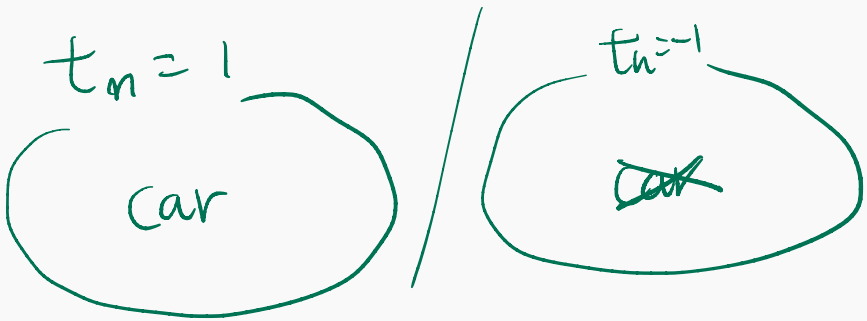
Maximum margin classifier

We begin our discussion of support vector machines to the two-class classification problem using linear models of the form

$$y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b \qquad (53)$$

where $\phi(\mathbf{x})$ denotes a fixed feature-space transformation, and we have made the bias parameter b explicit.

The training data set comprises N input vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ with corresponding target values $\{t_1, t_2, ..., t_N\}$ where $t_n \in \{-1, 1\}$, and new data points $\mathbf{x}$ are classified according to the sign of $y(\mathbf{x})$



Maximum margin classifier

→ convex optimization { objective, constraint

We shall assume that the training data set is linearly separable in feature space, so that by definition there exists at least one choice of the parameters $\mathbf{w}$ and b such that a function satisfies $y(\mathbf{x}_n) > 0$ for points having $t_n = +1$ and $y(\mathbf{x}_n) < 0$ for points having $t_n = -1$, so that $t_n y(\mathbf{x}_n) > 0$ for all training data points.

hard margin, ⇒ 전부다 나눠야 한다.

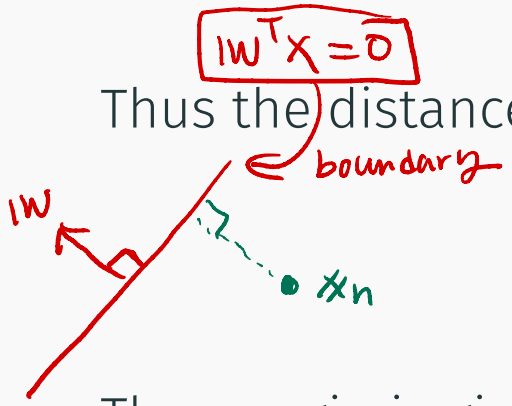
$t_n = 1$, $\mathbf{w}^T \mathbf{x}_n > 0$

$t_n = -1$, $\mathbf{w}^T \mathbf{x}_n < 0$

⇒ $t_n (\mathbf{w}^T \mathbf{x}_n) > 0$ ← $y(\mathbf{x}_n)$

constraint

Maximum margin classifier: optimality criterion



Thus the distance of a point $\mathbf{x}_n$ to the decision surface is given by

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}. \quad (54)$$

The margin is given by the perpendicular distance to the closest point $\mathbf{x}_n$ from the data set, and we wish to optimize the parameters $\mathbf{w}$ and b in order to maximize this distance. Thus the maximum margin solution is found by solving

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n [t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)] \right\} \quad (55)$$

where we have taken the factor $1/\|\mathbf{w}\|$ outside the optimization over n because $\mathbf{w}$ does not depend on n.

Dual problem for convex optimiztion

$$\left(\begin{array}{ll} \max & \frac{1}{\|w\|}\min\{t_n(w^T\phi(x_n)+b)\} \\ \text{s.t.} & t_n(w^T\phi(x_n)+b) > 0 \end{array}\right) \Rightarrow$$

primal problem

dual problem

Convex hold

Strong duality

Solution

Direct solution of this optimization problem would be very complex, so we shall convert it into an equivalent problem that is much easier to solve.

Lagrangian function with constraint

In order to solve this constrained optimization problem, we introduce Lagrange multipliers $a_n \geq 0$, with one multiplier a_n for each of the constraints, giving the Lagrangian function

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - 1\} \tag{56}$$

where $\mathbf{a} = (a_1, ..., a_N)^T$. Note the minus sign in front of the Lagrange multiplier term, because we are minimizing with respect to $\mathbf{w}$ and b, and maximizing with respect to $\mathbf{a}$.

Setting the derivatives of $L(\mathbf{w}, b, \mathbf{a})$ with respect to $\mathbf{w}$ and b equal to zero, we obtain the following two conditions

$$\begin{cases} \mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) & (57) \\ 0 = \sum_{n=1}^{N} a_n t_n & (58) \end{cases}$$

$$y(x) = \sum a_n t_n \phi(x_n)^T \phi(x)$$

Lagrangian function with constraint

Eliminating $\mathbf{w}$ and b from $L(\mathbf{w}, b, \mathbf{a})$ using these conditions then gives the *dual representation* of the maximum margin problem in which we maximize

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{59}$$

with respect to $\mathbf{a}$ subject to the constraints

$$a_n \geq 0, \qquad n = 1, ..., N \tag{60}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{61}$$

Here the kernel function is defined by $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$.

Prediction for a new sample: support vector machine

SVM ← (Support vector / data point.) ← 새로운 데이터 비교해보고 결정을내리는.

In order to classify new data points using the trained model, we evaluate the sign of $y(\mathbf{x})$. This can be expressed in terms of the parameter $\{a_n\}$ and the kernel function by substituting for $\mathbf{w}$ to give

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \tag{62}$$

$$y(x) = \sum a_n t_n \phi(x_n)^T \phi(x) + b$$

prediction, classification → $y(x) = \sum a_n t_n x_n^T x + b$

Support vector (O)

$a_n = 0$, margin 밖 → x_n은 여측에 영향X

$a_n > 0$, margin에걸침 → x_n은 여측에 영향O

KKT condition: complementary slackness

convex. dual problem.

We show that a constrained optimization of this form satisfies the *Karush-Kuhn-Tucker* (KKT) conditions, which in this case require that the following three properties hold

$$a_n \geq 0 \tag{63}$$

$$t_n(\mathbf{x}_n) - 1 \geq 0 \tag{64}$$

$$a_n\{t_n y(\mathbf{x}_n) - 1\} = 0 \tag{65}$$

Thus for every data point, either $a_n = 0$ or $t_n y(\mathbf{x}_n) = 1$. Any data point for which $a_n = 0$ will not appear in the sum and hence plays no role in making predictions for new data points. The remaining data points are called *support vectors*, and because they satisfy $t_n y(\mathbf{x}_n) = 1$, they correspond to points that lie on the maximum margin hyperplanes in feature space.

KKT condition: complementary slackness

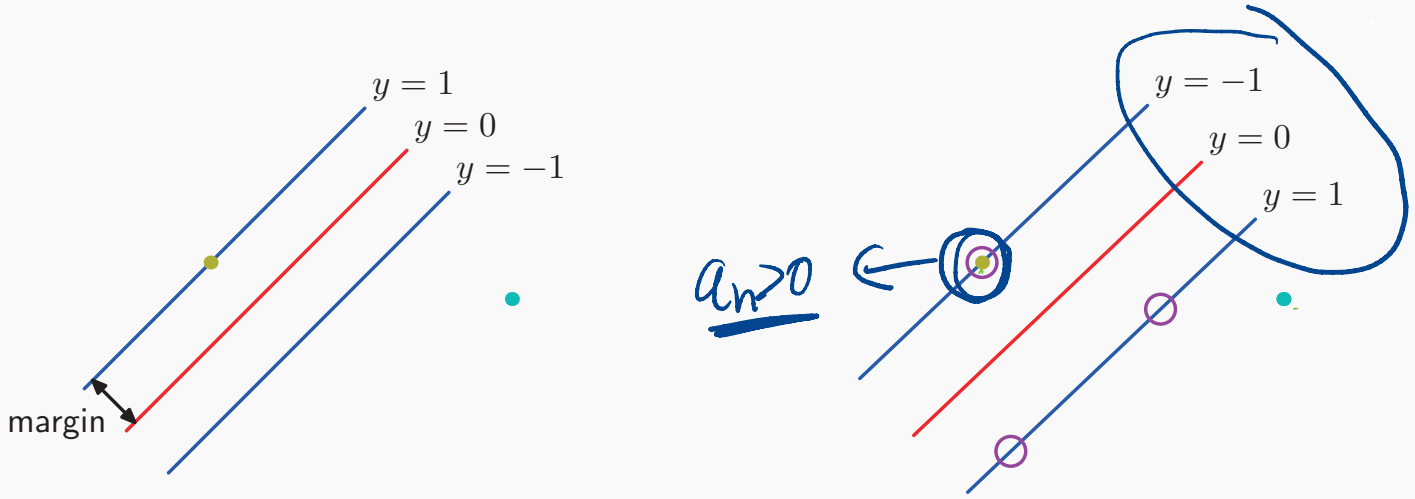
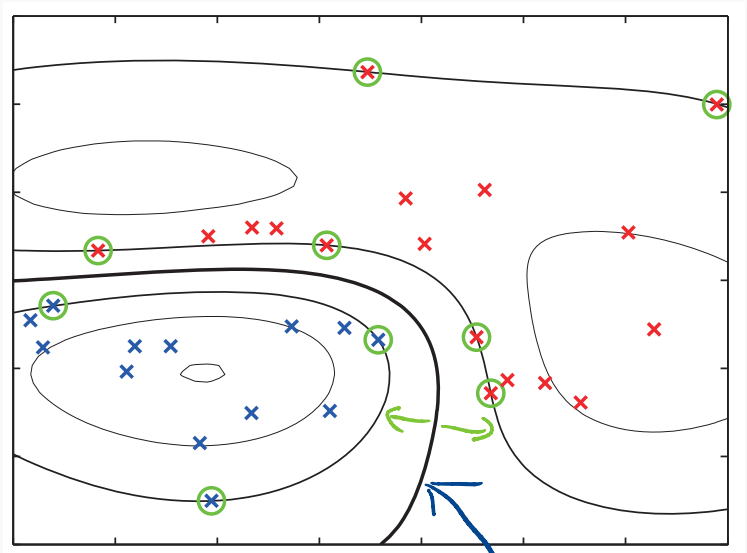


Figure 7.1 The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by a subset of the data points, known as support vectors, which are indicated by the circles.

KKT condition: complementary slackness

$\phi(x)$: feature mapping

Figure 7.2 Example of synthetic data from two classes in two dimensions showing contours of constant $y(\mathbf{x})$ obtained from a support vector machine having a Gaussian kernel function. Also shown are the decision boundary, the margin boundaries, and the support vectors.



0 = support vector

Appendix

Reference and further reading

- “Chap 14 | Recognition” of R. Szeliski, Computer Vision: Algorithms and Applications
- “Chap 15 | Learning to Classify”, “Chap 16 | Classifying Images” and “Chap 17 | Detecting Objects in Images” of Forsyth and Ponce, Computer Vision: A Modern Approach
- “Lecture18 | Object Recognition I” and “Lecture19 | Object Recognition II” of Bohyung Han, CSED441: Introduction to Computer Vision, POSTECH (2011)
- “Chap 7 | Sparse Kernel Machines” of C. Bishop, Pattern Recognition and Machine Learning
- “Chap 5 | Support Vector Machines”, “Chap 10 | Introduction to Artificial Neural Network with Keras” and “Chap 14 | Deep Computer Vision Using Convolutional Neural Network” of A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
- “Chap 4 | Convex Optimization Problems” and “Chap 5 | Duality” of S. Boyd, Convex Optimization
- “Lecture 6 | Support Vector Machines” and “Lecture 10 | Convolutional Neural Networks” of Kwang Il Kim, Machine Learning (2019)