Fully-Connected Deep Networks

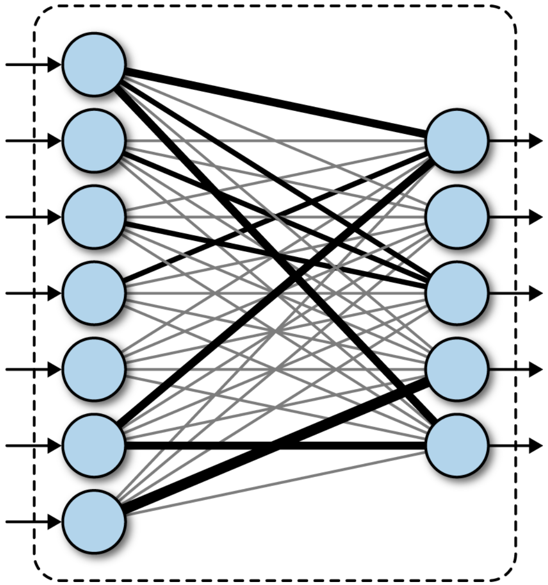
Hee-il Hahn

Professor
Department of Information and Communications Engineering
Hankuk University of Foreign Studies
hihahn@hufs.ac.kr

Fully-Connected Deep Networks

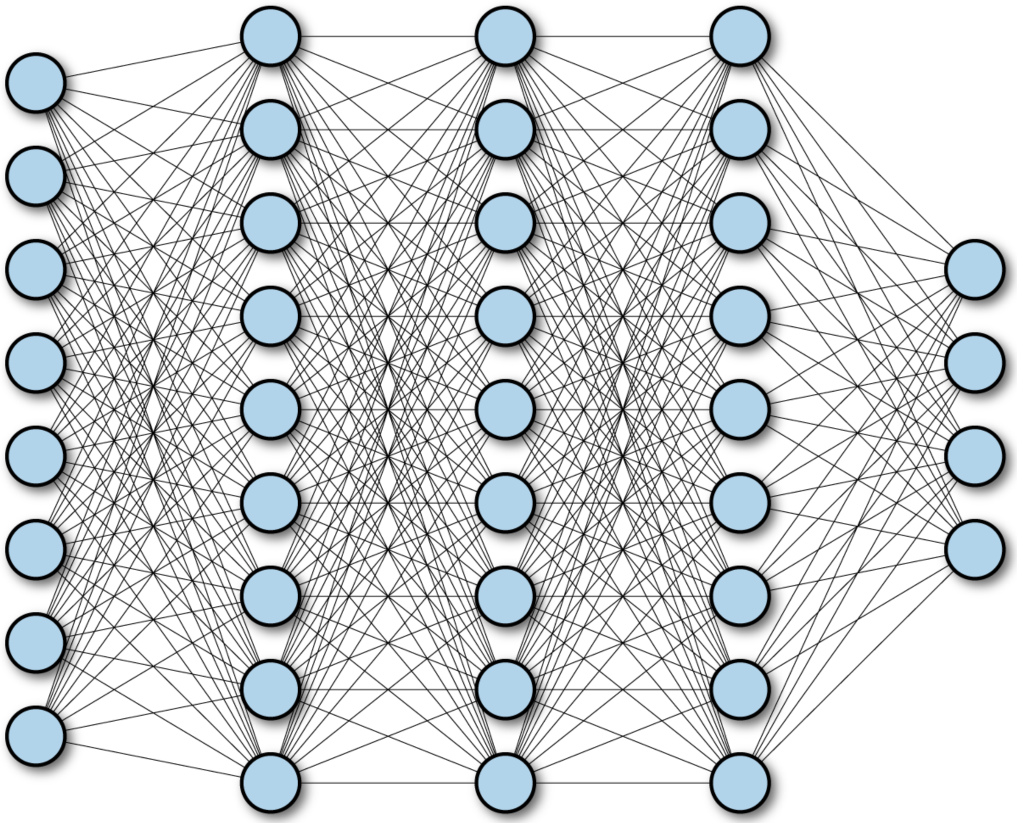
- Fully-connected NN은 일련의 fully-connected layer로 구성된다.
- fully-connected layer는 $\mathrm{F}: \mathbb{R}^m \longrightarrow \mathbb{R}^n$ 인 함수로 볼 수 있다.
- 각각의 출력차원은 입력차원에 종속적이다.

- $y_i = \sigma(w_1 x_1 + \cdots + w_m x_m)$
- $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \sigma\left(w_{1,1}x_1 + \cdots + w_{1,m}x_m\right) \\ \vdots \\ \sigma\left(w_{n,1}x_1 + \cdots + w_{n,m}x_m\right) \end{pmatrix} = \sigma(wx)$
- $w = \begin{pmatrix} w_{1,1} & \cdots & w_{1,m} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & w_{n,m} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$



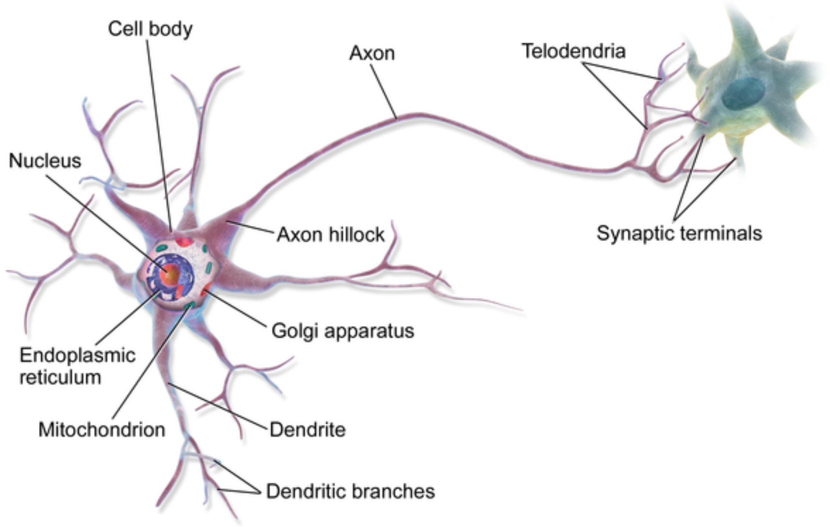
Fully-Connected Deep Networks – cont.

- Fully-connected network를 차례로 쌓아 deep network를 구성할 수 있다.



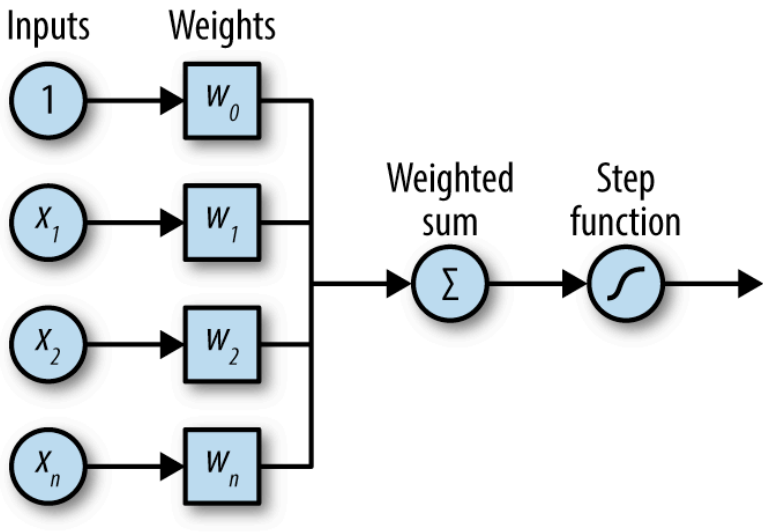
완전연결 네트워크에서의 뉴런

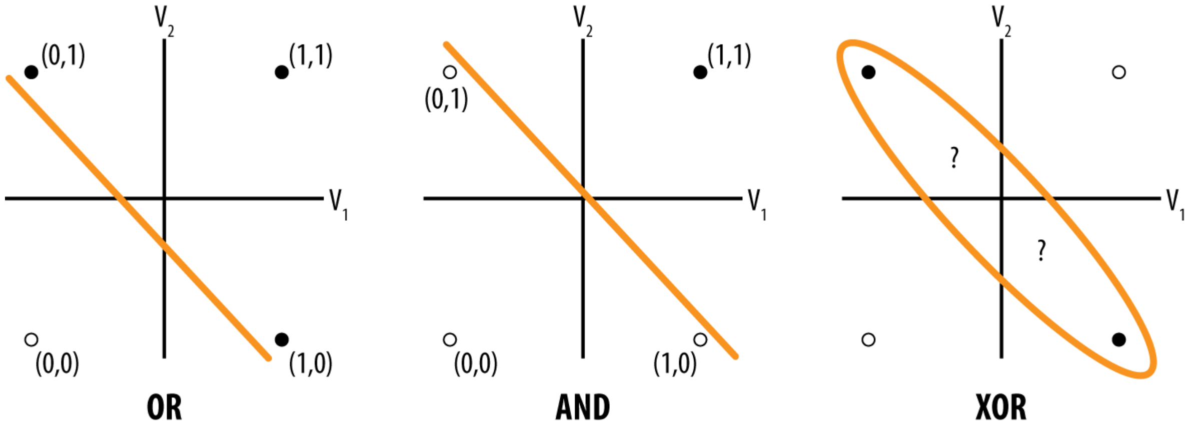
- 완전연결 네트워크를 "신경망"이라고 부른다.
- 완전연결 네트워크의 노드를 "뉴런"이라고 부른다.
- 1940년대 McCulloch와 Pitts는 뉴런이 Bullean-valued function을 임의로 계산할 수 있다고 주장하는 최초의 뇌수학 모델을 발표
- 그 후, 0에서 1 사이에 분포하는 수학적 뉴런 연속함수를 이용해 논리모델을 개선
- 연속함수의 입력이 충분히 커지면 뉴런은 fire(1을 출력)되고 그렇지 않으면 0을 출력



Error Back Propagation으로 FCN 학습

- 최초의 완전연결 신경망 : Rosenblatt이 만든 Perceptron



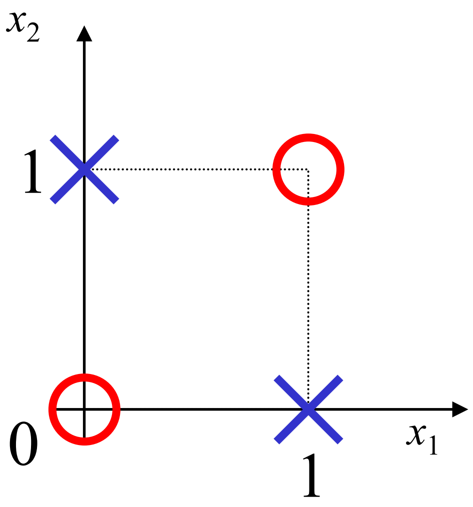


- Minsky와 Paypert : 단순 perceptron은 XOR 함수를 배울 수 없음을 증명
- MLP는 back propagation 알고리즘으로 학습 가능

MLP로 XOR 함수를 분류할 수 있는 이유

Example:

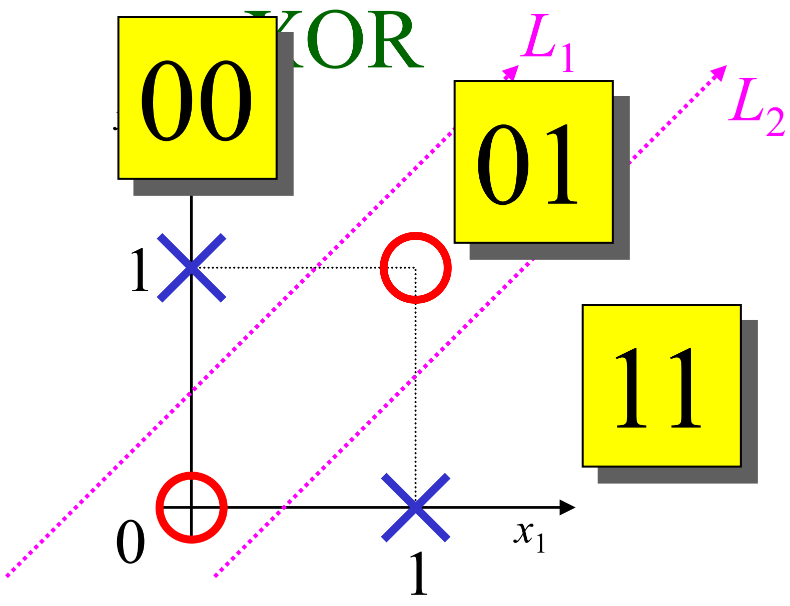
XOR

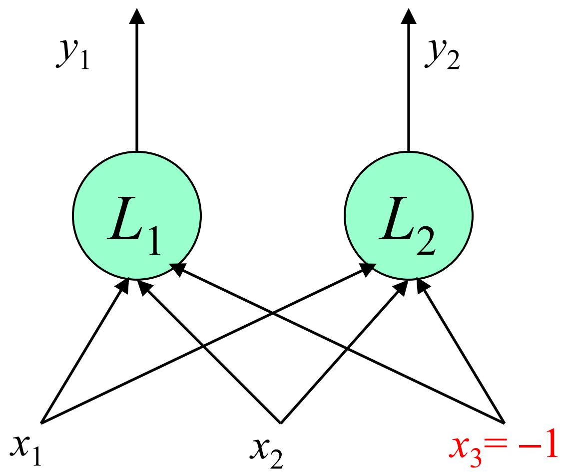


- Not linearly separable.
- Is a single layer perceptron workable?

MLP로 XOR 함수를 분류할 수 있는 이유 – cont.

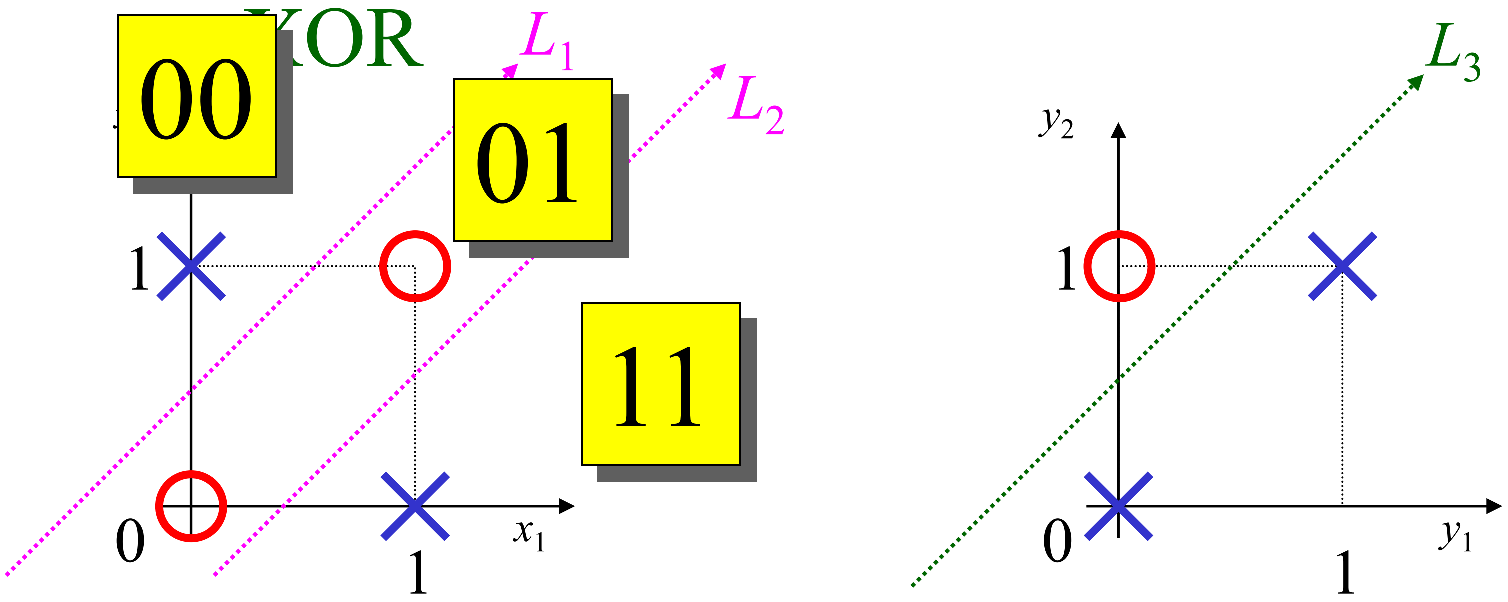
Example:





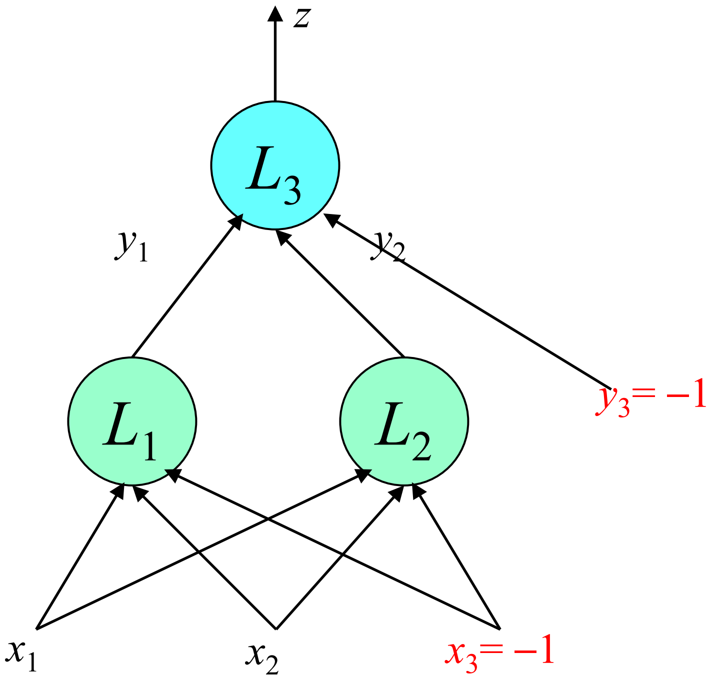
MLP로 XOR 함수를 분류할 수 있는 이유 – cont.

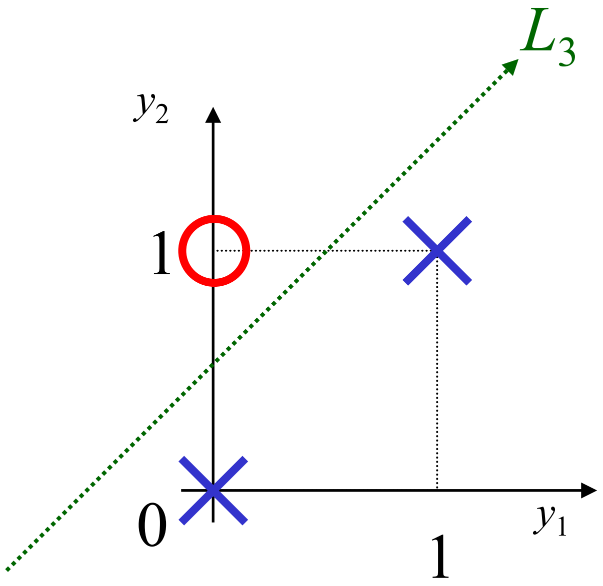
Example:



MLP로 XOR 함수를 분류할 수 있는 이유 – cont.

Example:

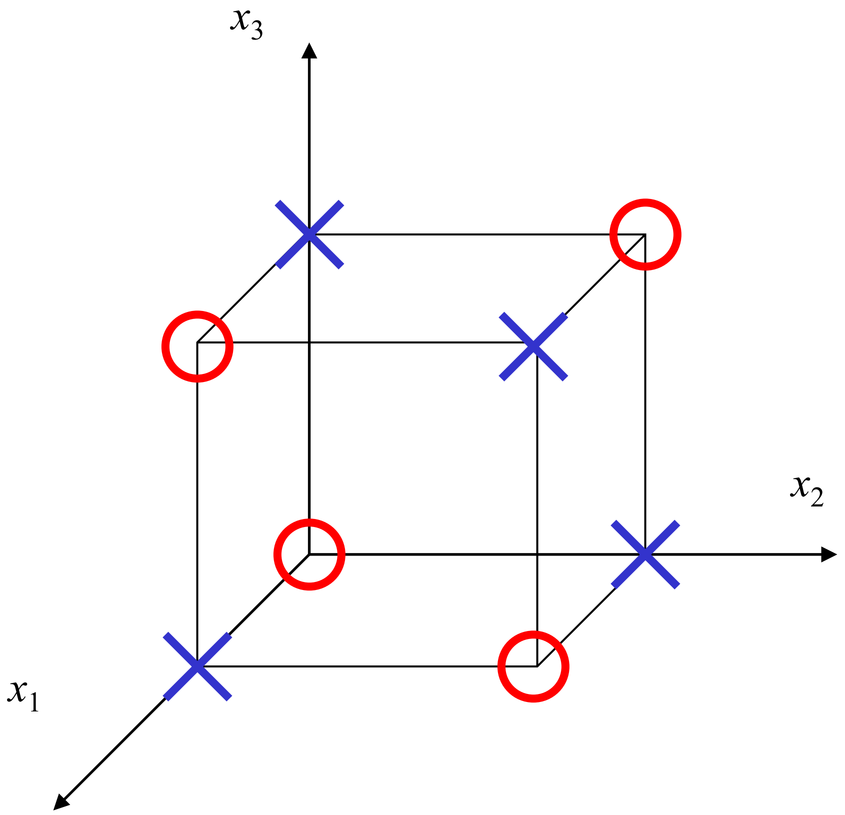




Parity Problem

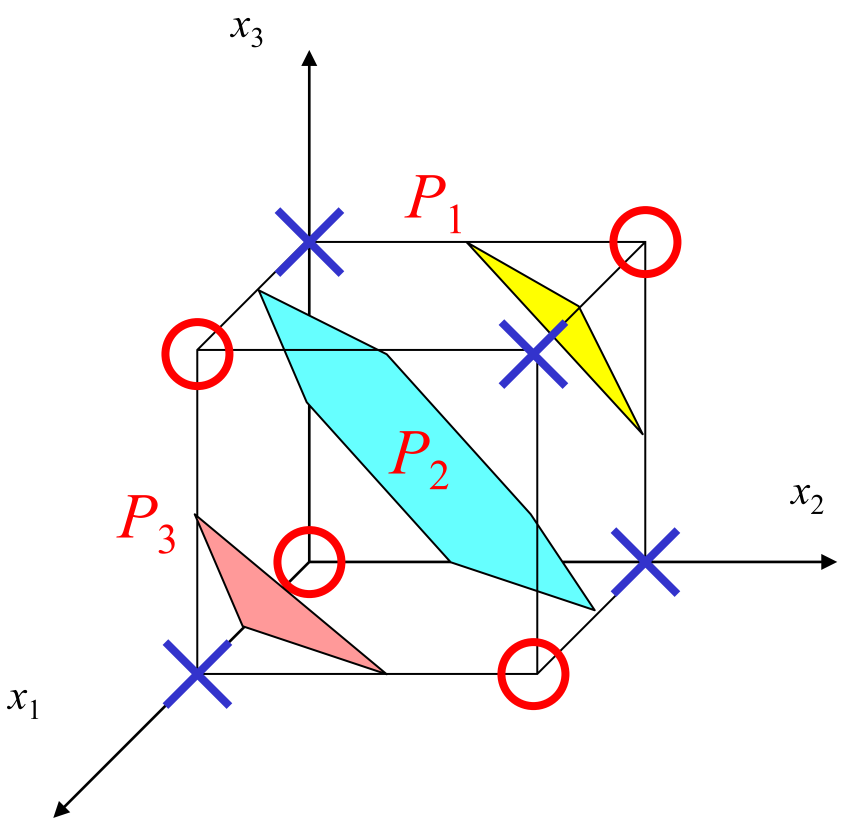
Is the problem linearly separable?

$x_1 x_2 x_3$	
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1



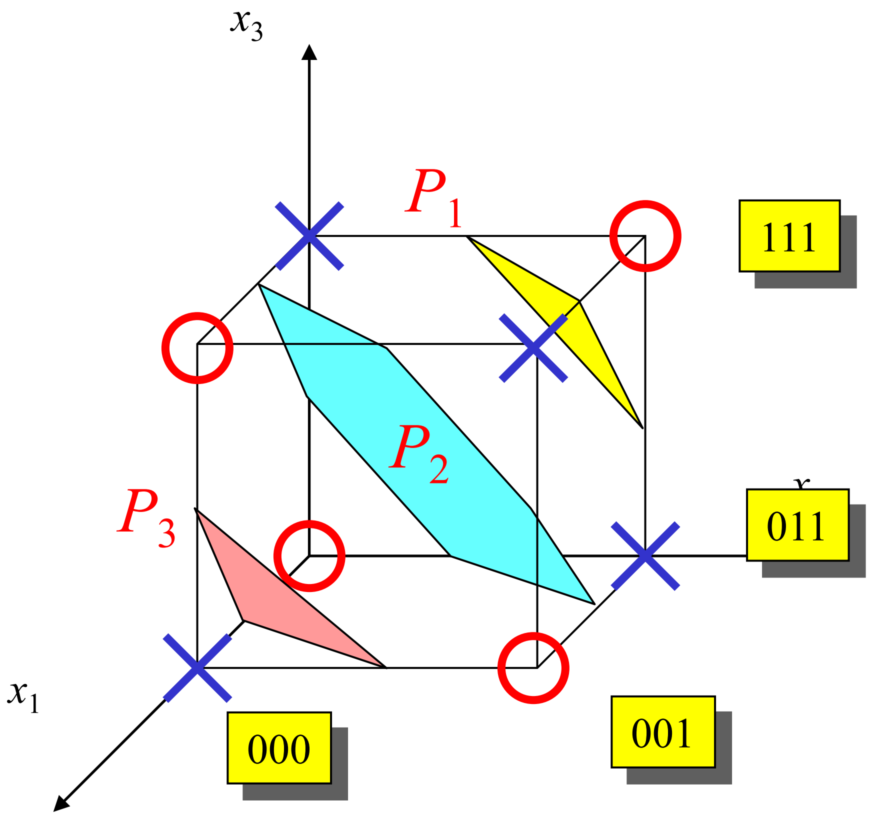
Parity Problem

$x_1 x_2 x_3$	
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

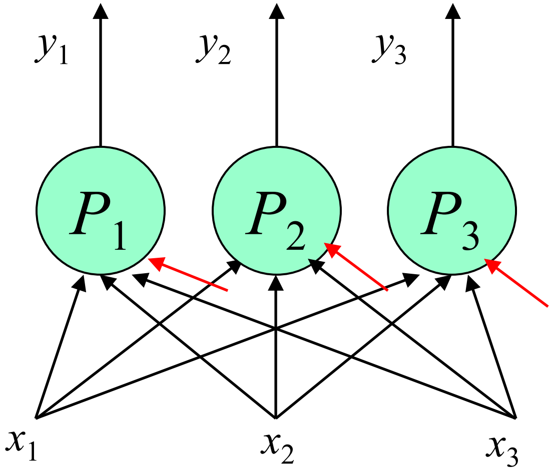


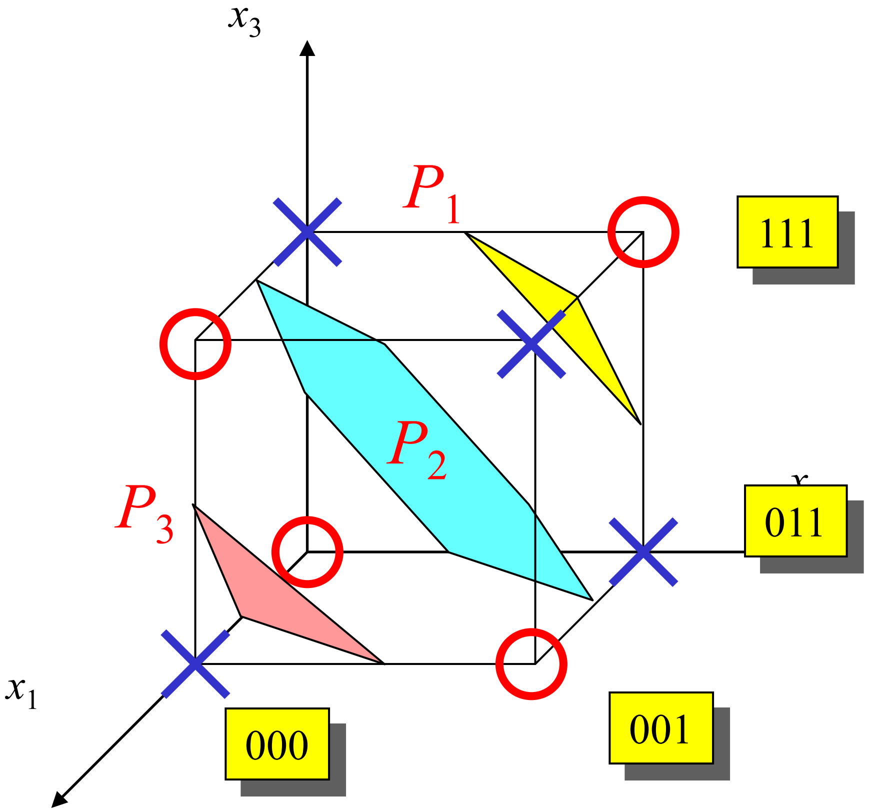
Parity Problem

$x_1 x_2 x_3$	
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

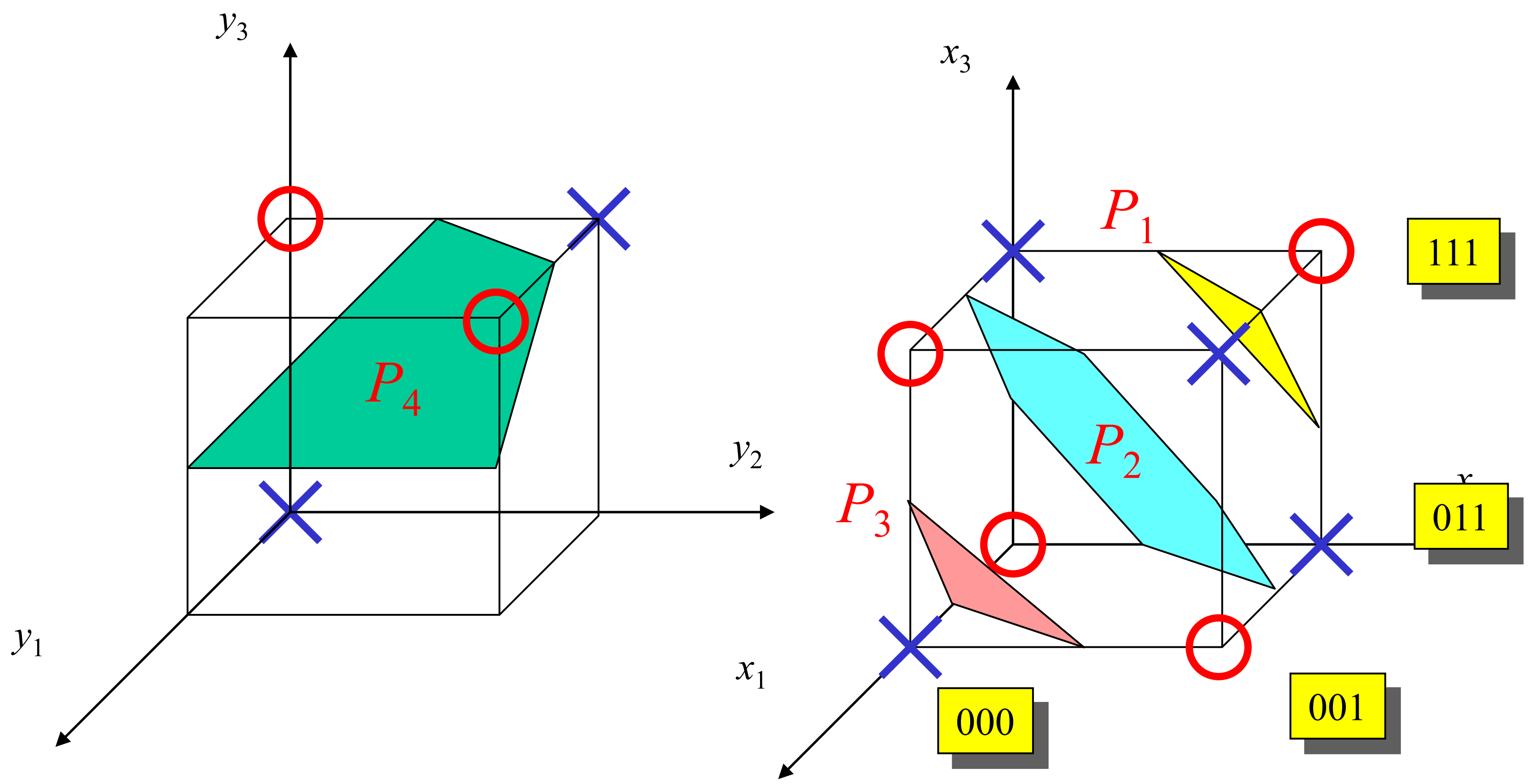


Parity Problem

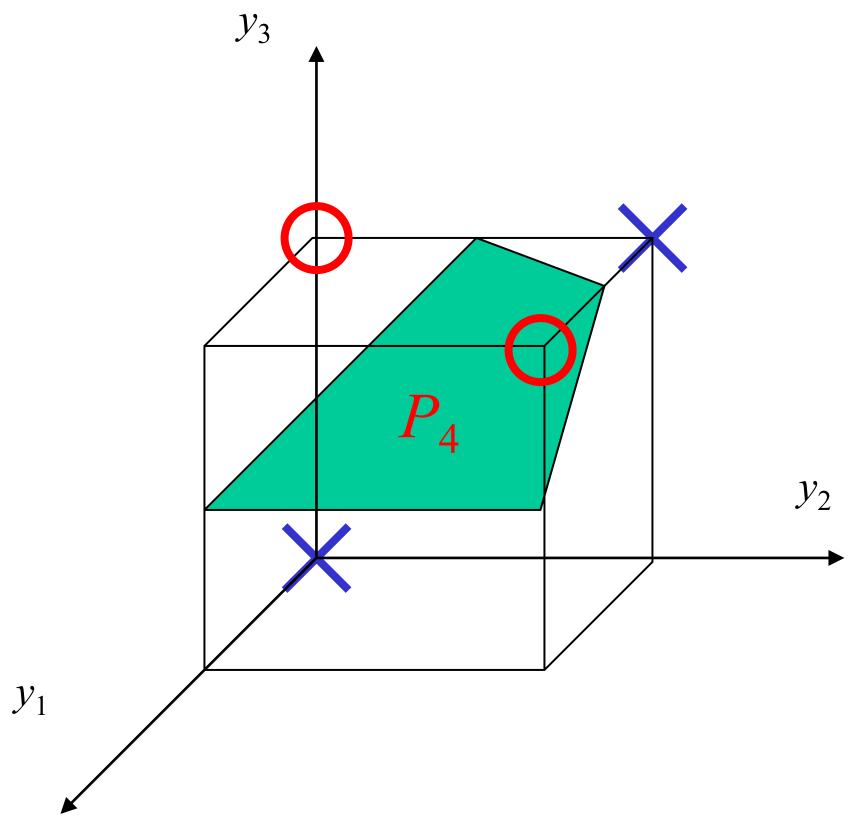


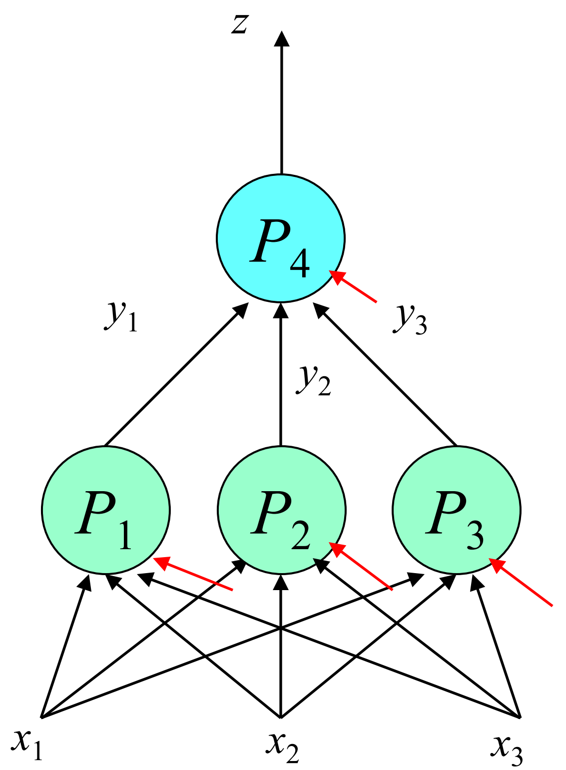


Parity Problem

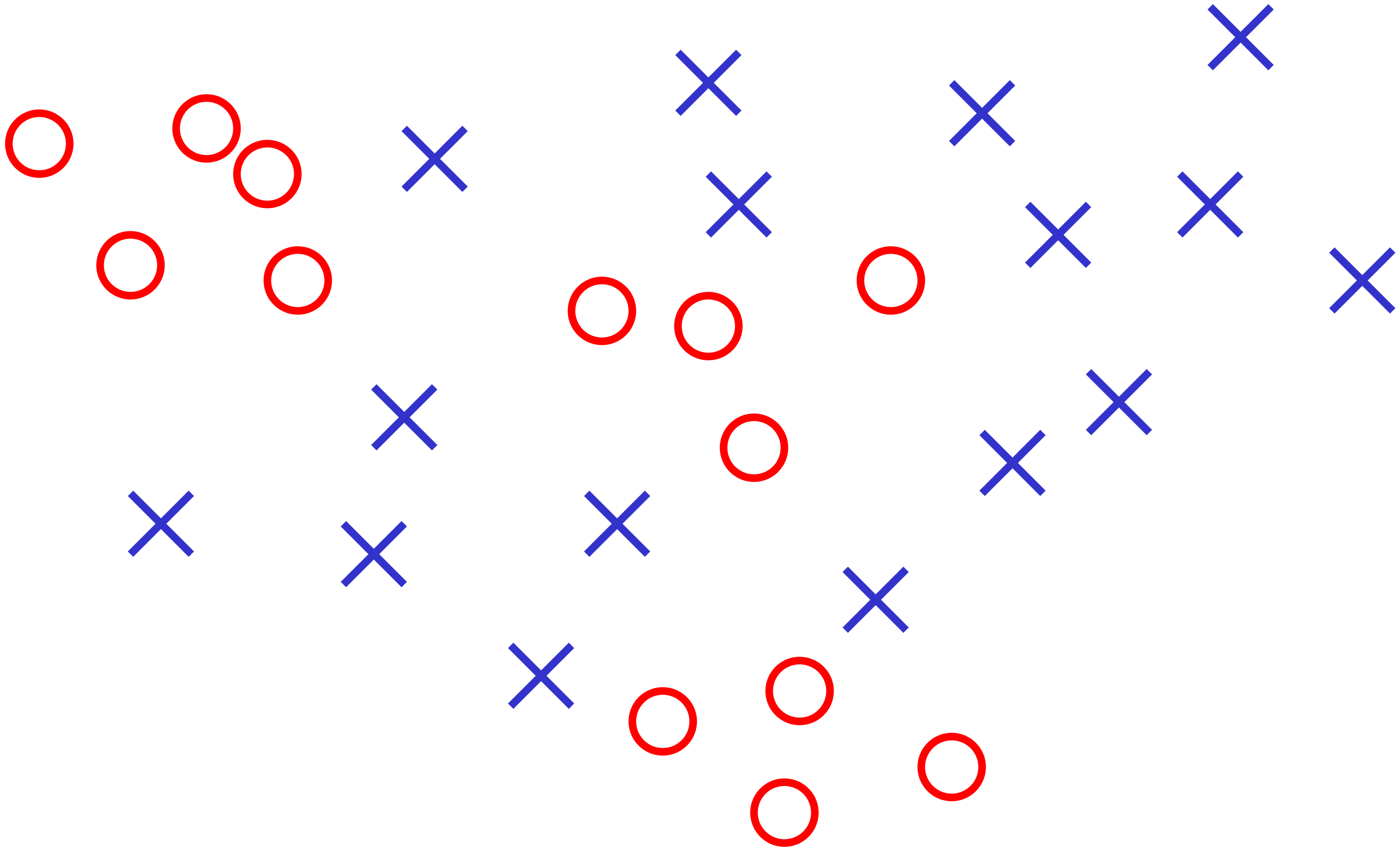


Parity Problem

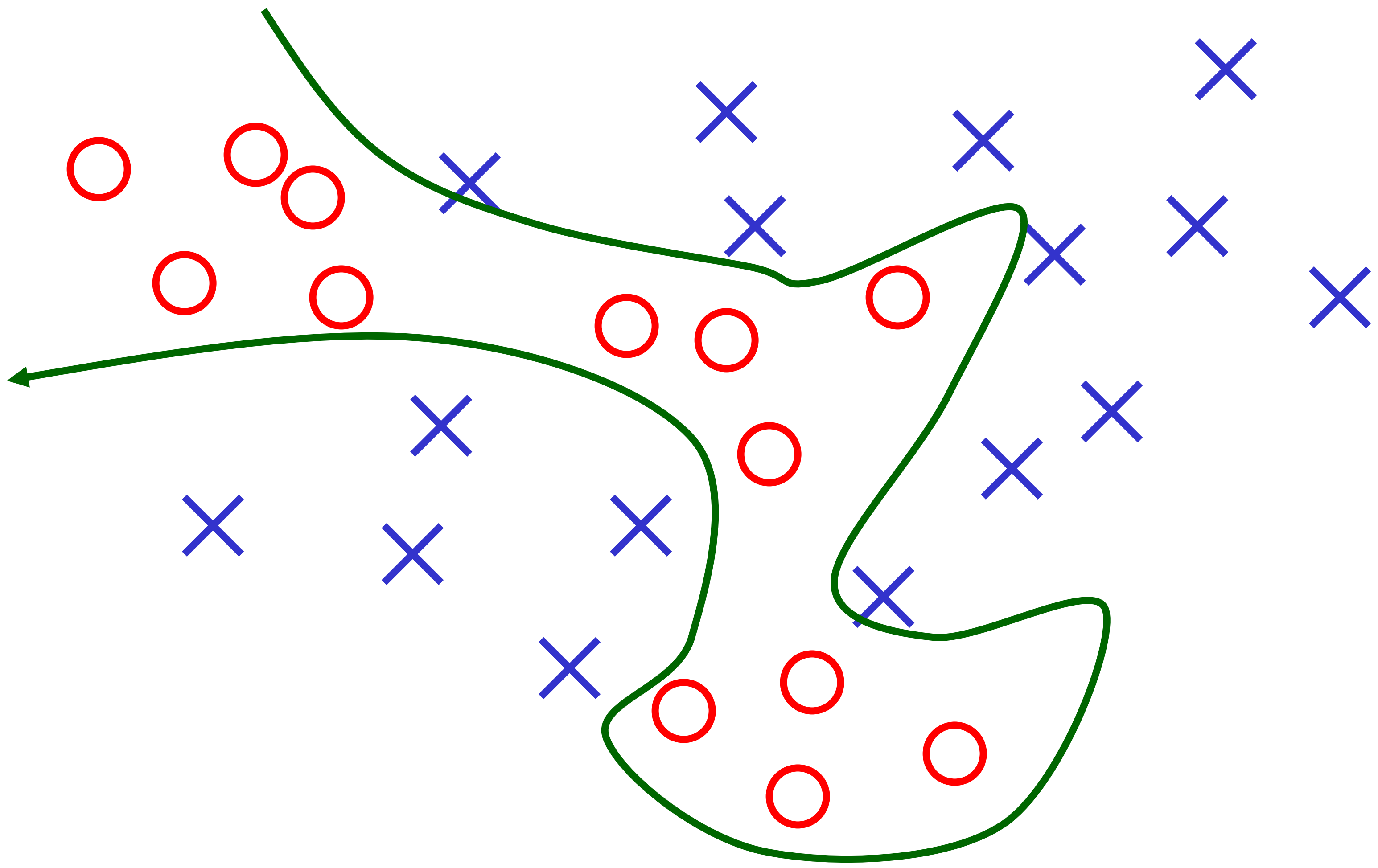




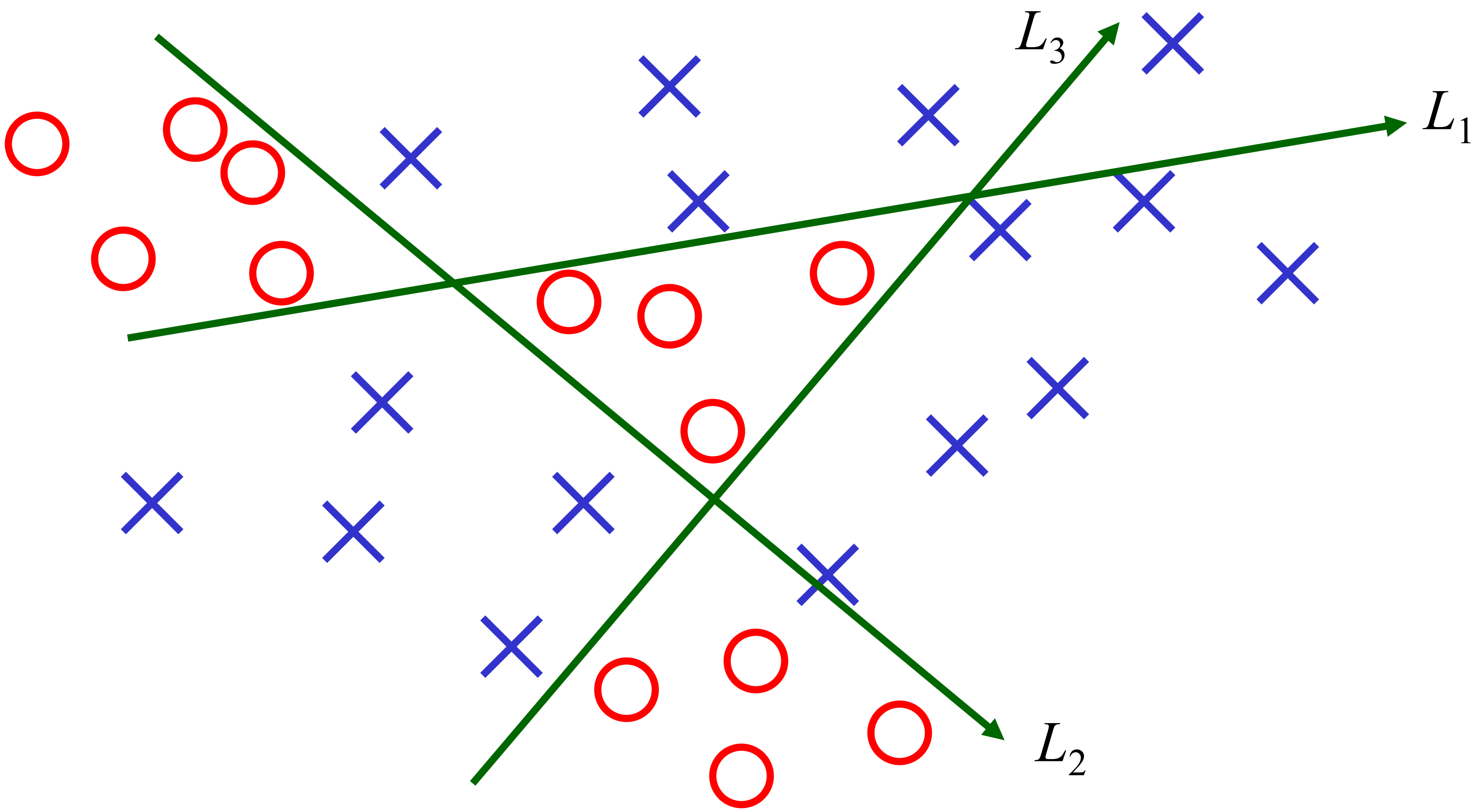
General Problem



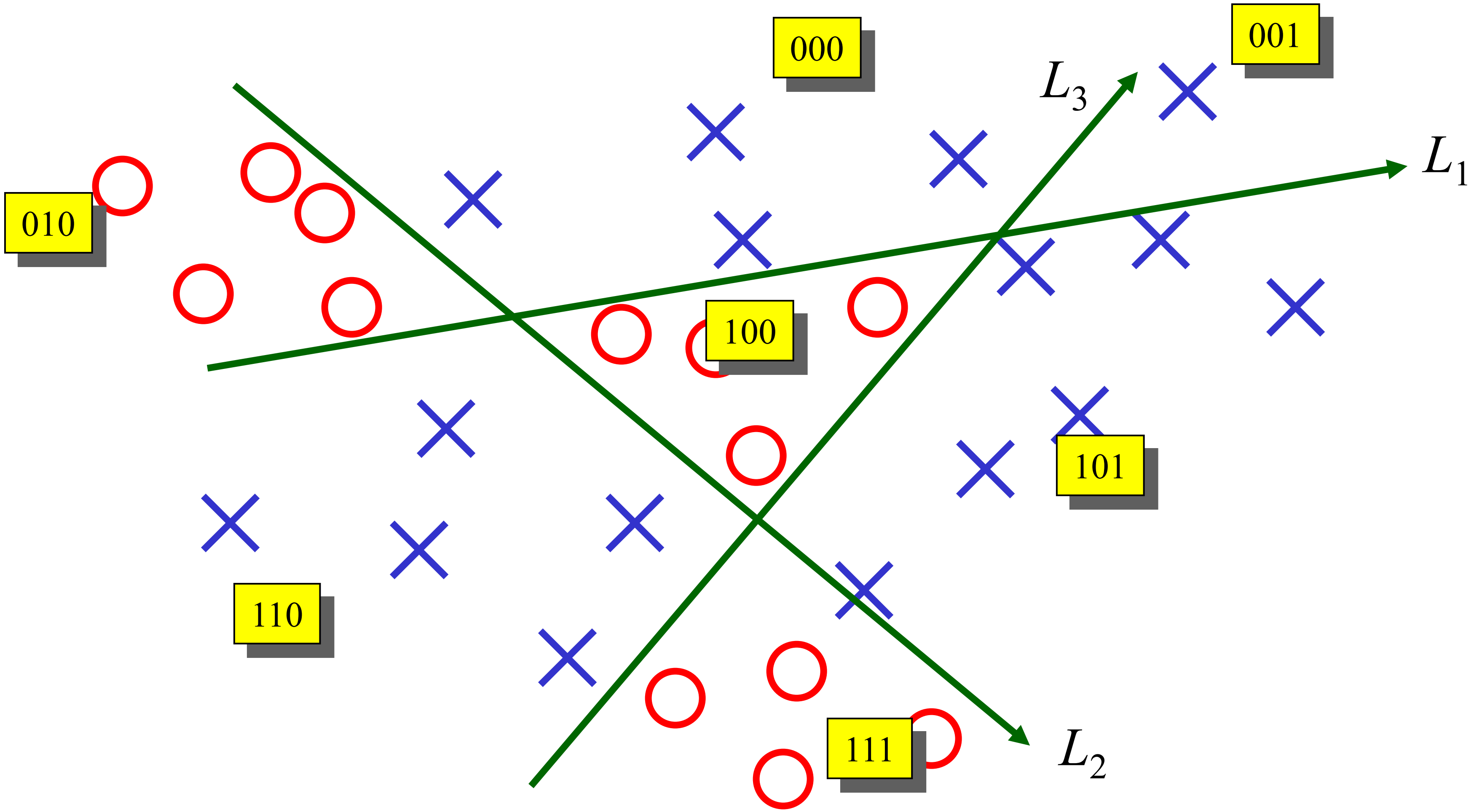
General Problem



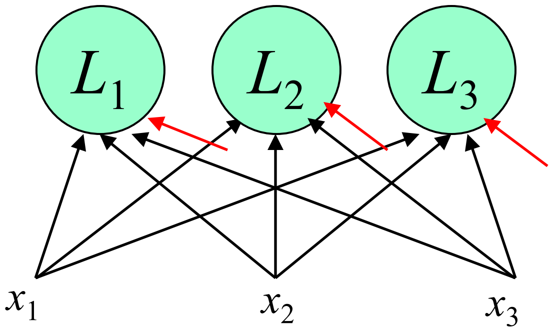
Hyperspace Partition

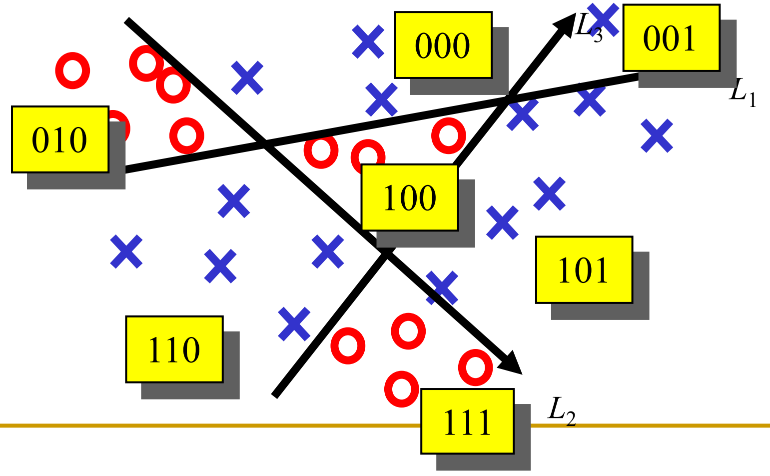


Region Encoding

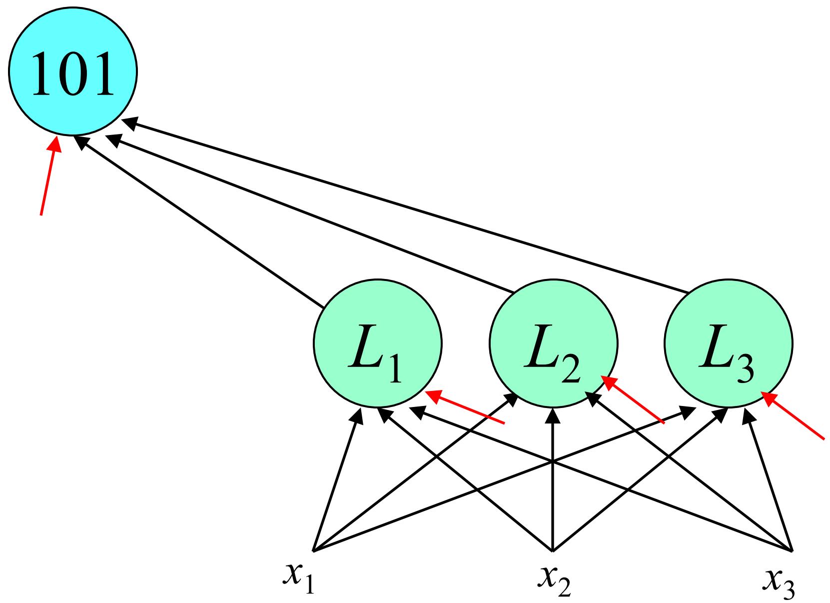


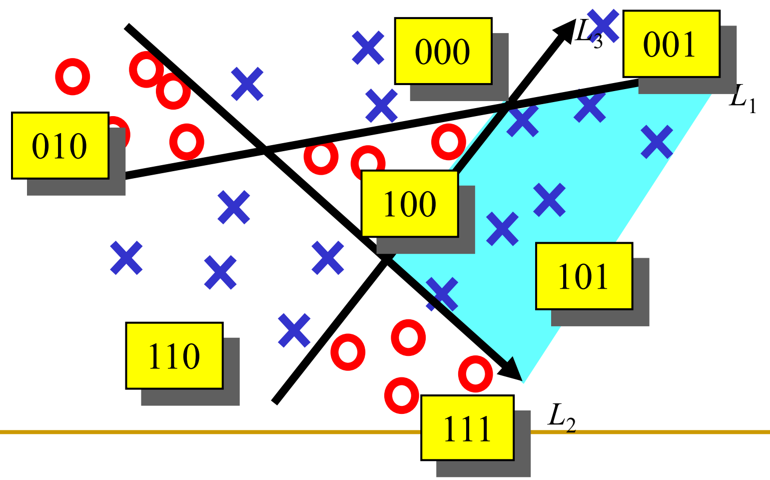
Hyperspace Partition & Region Encoding Layer



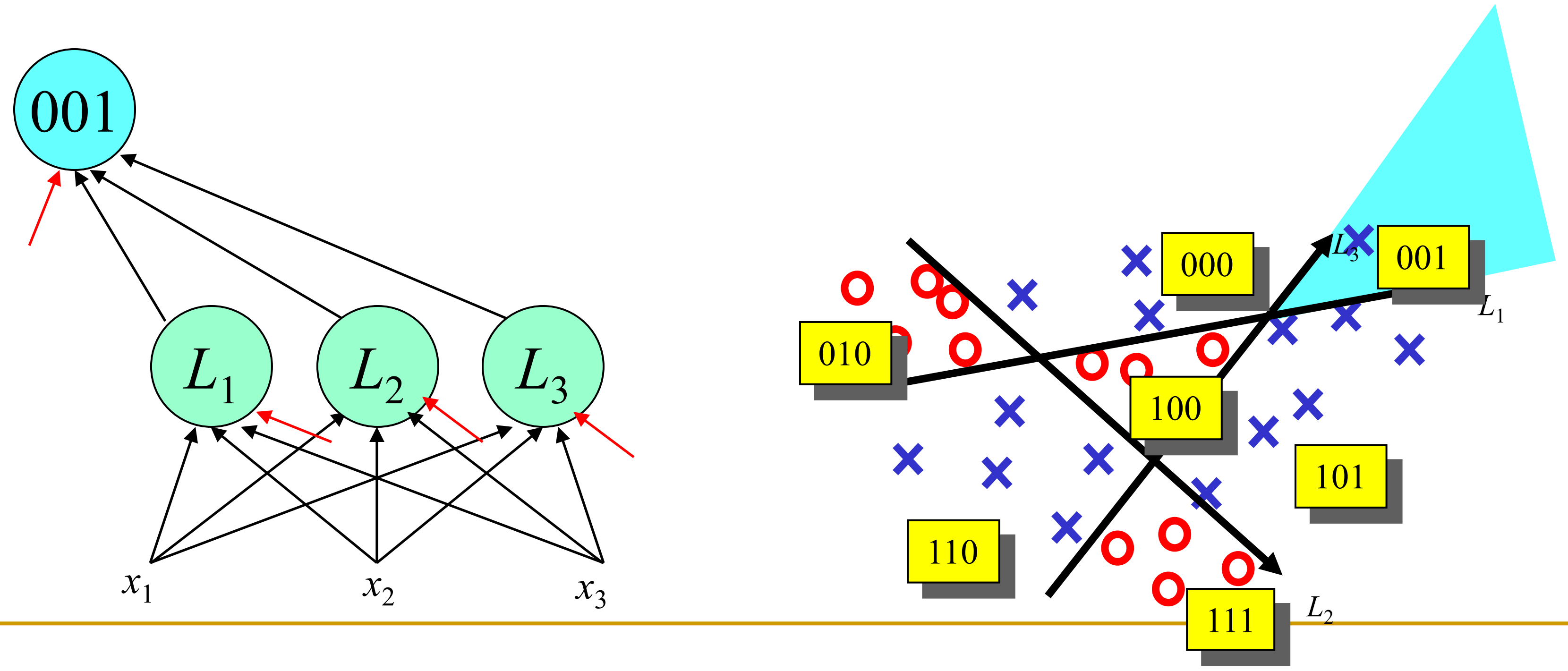


Region Identification Layer

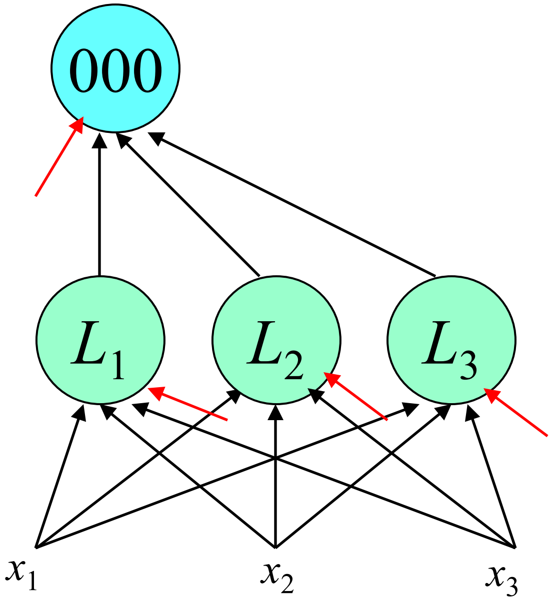


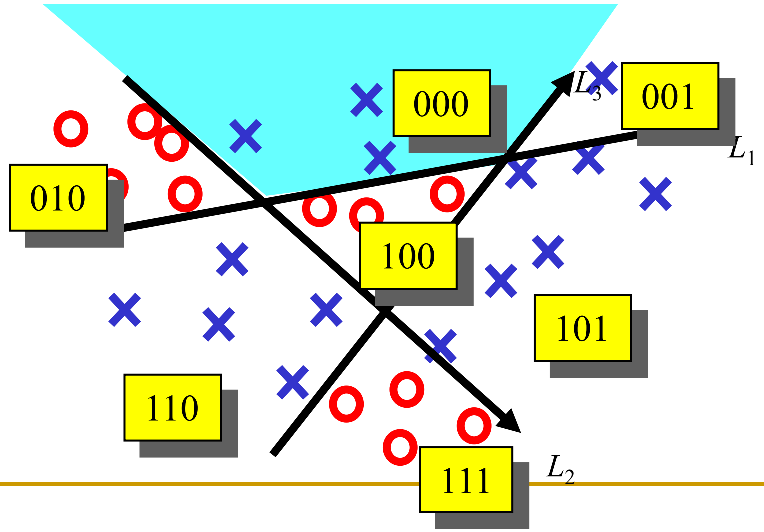


Region Identification Layer

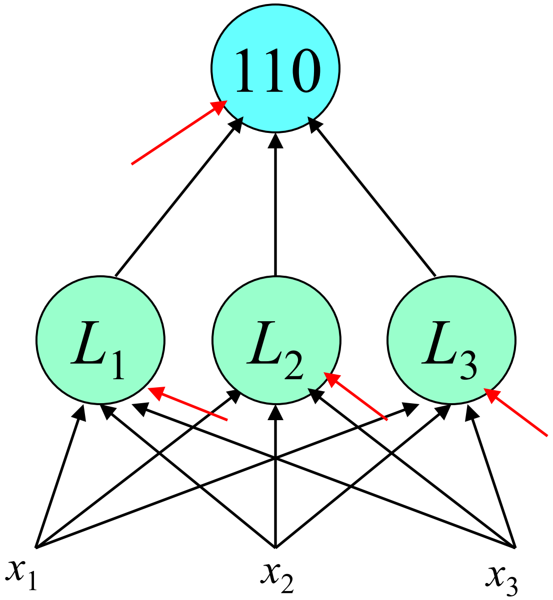


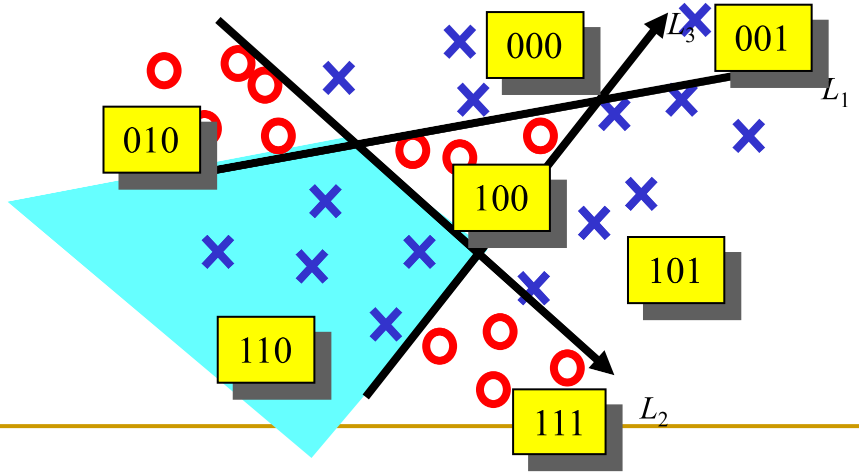
Region Identification Layer



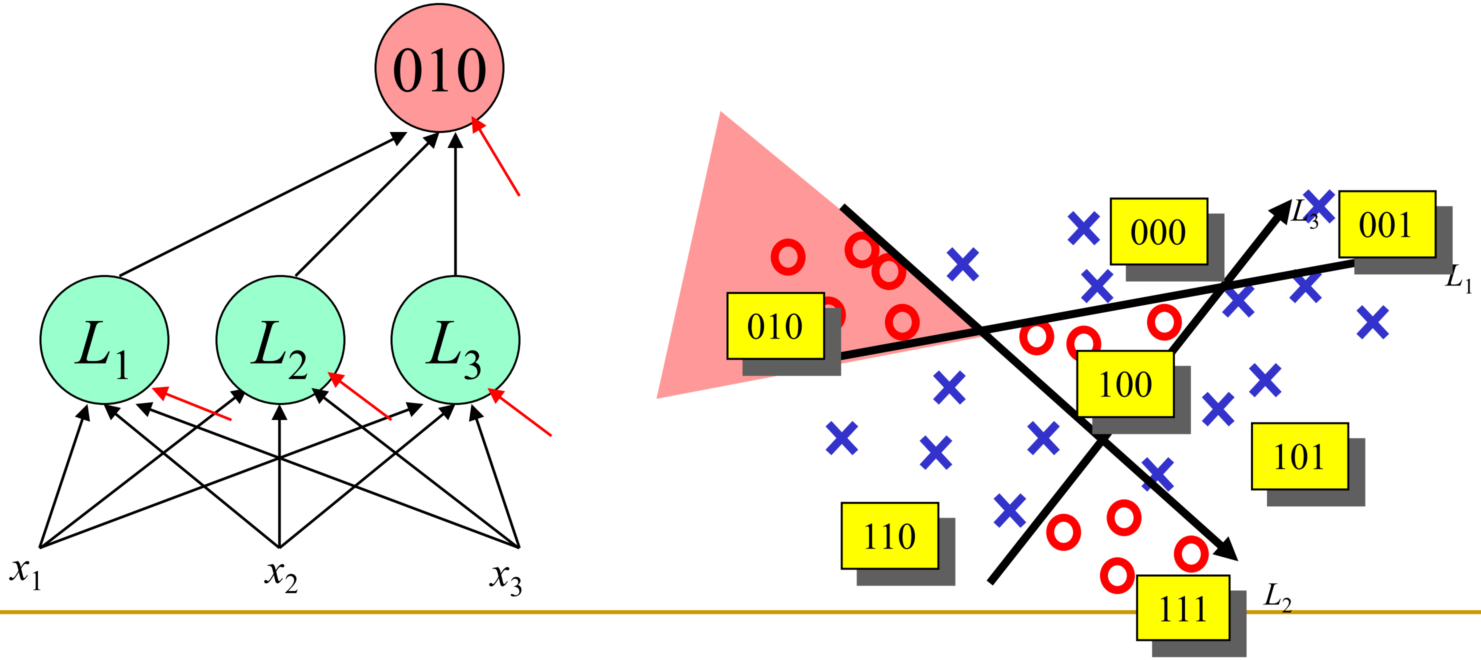


Region Identification Layer

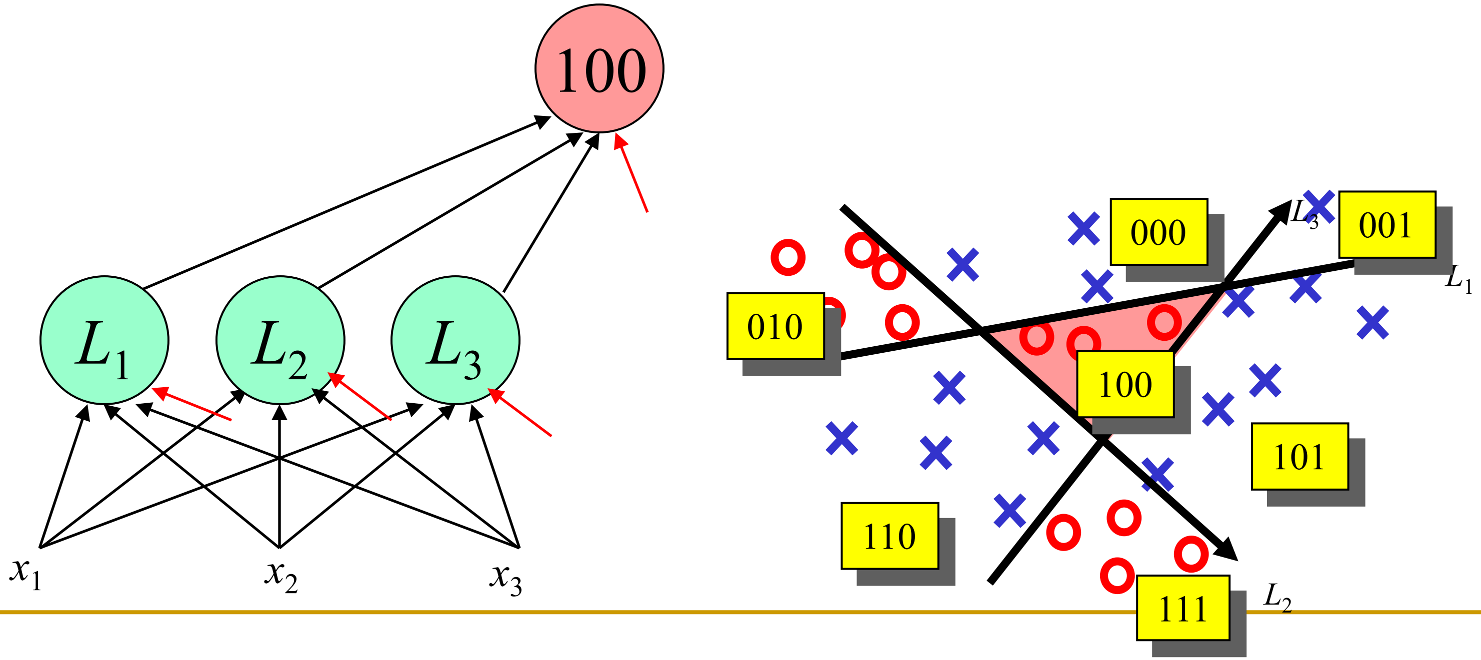




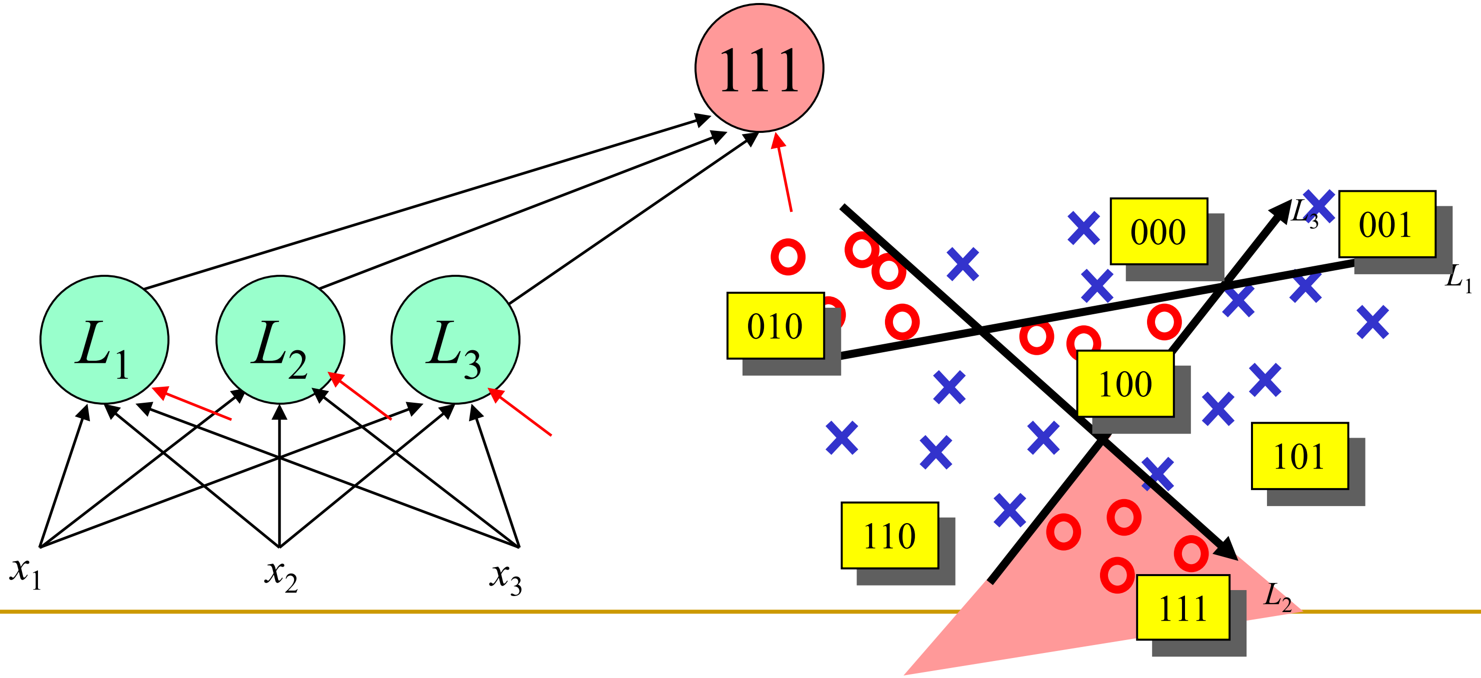
Region Identification Layer



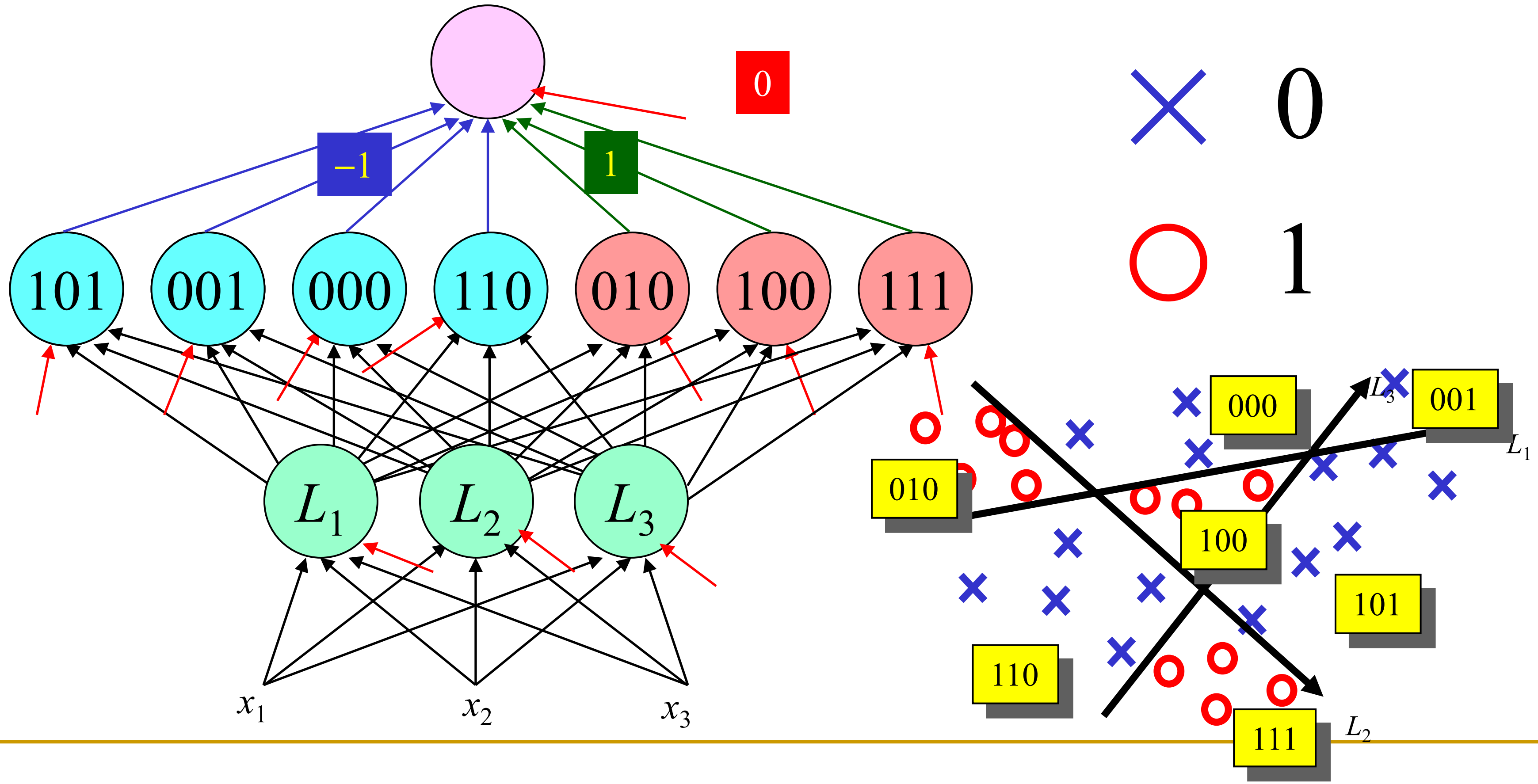
Region Identification Layer



Region Identification Layer



Classification



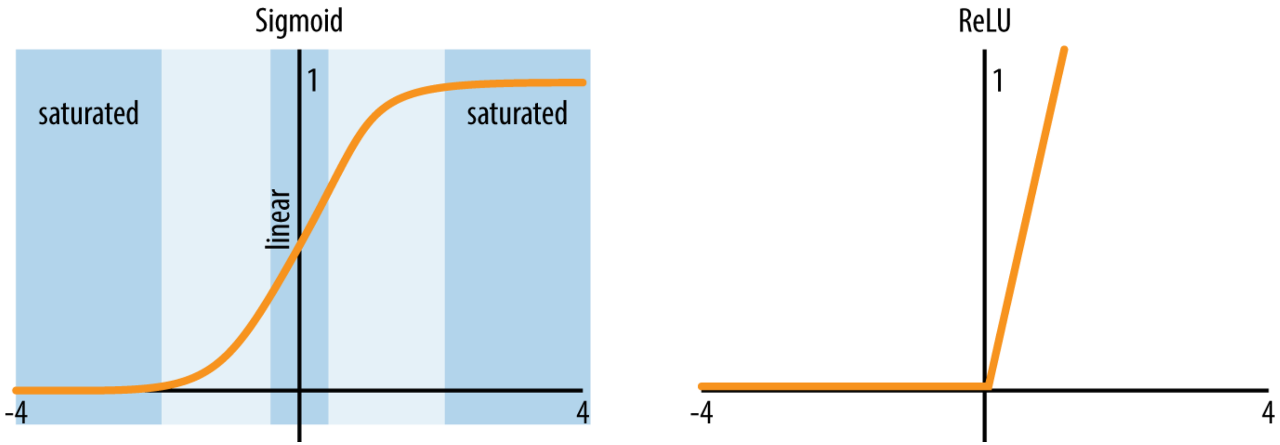
완전연결 신경망 학습

- 학습 가능한 표현
  - 각 FCN layer는 문제가 존재하는 특징공간을 변환한다.
  - 문제의 표현을 변환해서 더 유연한 시스템을 구성한다. ⟶ Representation Learning
  - 예를 들어, Fourier Transform, Laplace Transform 등을 이용해 복잡한 식과 함수를 단순화해서 분석하기 적합한 형태로 변환하는 기법 ⟶ 고전적인 공학적 문제해결 방법 (explicit)
  - 딥러닝 기법도 당면한 문제에 맞춰 데이터 중심으로 변환하는 기법 채용 (implicit)
  - 일반적인 변환기술로는 이미지나 음성분석 문제를 풀수 없지만 딥 네트워크는 학습가능한 표현에 대해 유연하기 때문에 문제해결 가능
  - 딥 러닝을 이용한 기법은 Fourier 기법같은 수학적 변환보다 일반적이지는 않지만 특정문제에 보다 강력한 해결도구가 될 수 있다.

- 회귀데이터셋 생성(다른 예)

완전연결 신경망 학습 – cont.

- Activation
  - sigmoid
  - ReLu(rectified linear activation) : $\sigma(x) = max(x, 0)$
  - Relu는 vanishing gradient 문제를 피할 수 있는 장점이 있다.



완전연결 신경망 학습 – cont.

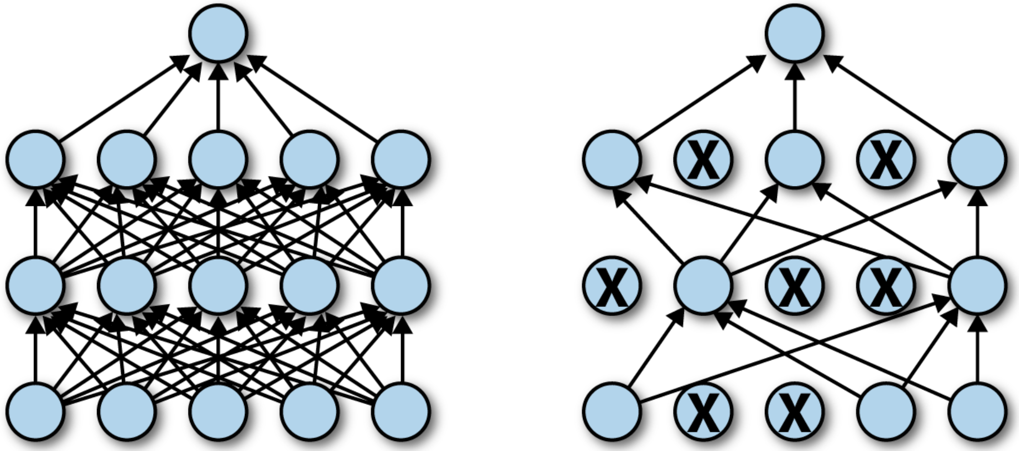
- FCN의 기억능력
  - FCN Learning에서 수렴은 크게 의미 있는 지표는 아니다.
  - 즉, learning error가 0에 근사하게 수렴했다고 해서 반드시 잘 학습되었다는 의미는 아니다.
  - Training dataset을 기억해서 잘 동작하고 그 외의 데이터에서는 동작하지 않을 수 있다.

완전연결 신경망 학습 – cont.

- Regularization
  - Generalizable learning으로 나아가도록 기억을 제한할 수 있는 수학적 기법

Dropout

- 완전연결 계층으로 입력되는 노드 일부를 무작위로 제거한다.
- 각 경사 하강 단계에서 무작위로 제거할 노드를 선정한다.
- 네트워크가 학습데이터 기억을 방지한다.
- 새로운 데이터에 대한 모델의 예측력을 증가시킨다.
- 예측할 때는 드롭아웃을 꺼야 한다.
- 그렇지 않으면 예측시 많은 잡음이 발생하여 유용성이 떨어진다.



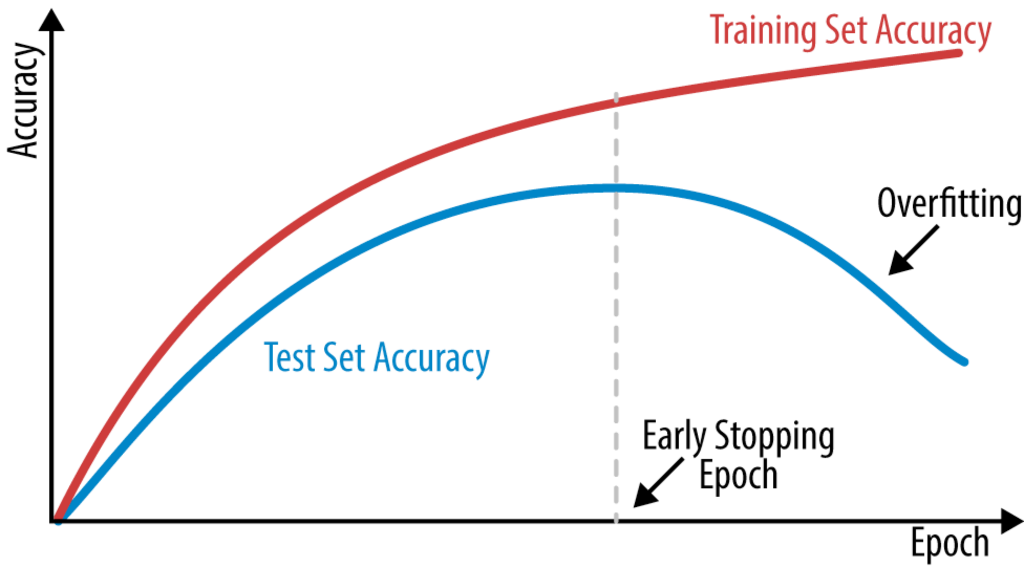
(a) Standard Neural Net

(b) After applying dropout

완전연결 신경망 학습 – cont.

Early Stopping

- FCN은 그 전까지 입력된 모든 것을 기억하려는 경향이 있다.
- Validation dataset에 대하여 네트워크 성능을 추적해서 만족하는지 추적하고,
- 성능이 떨어지기 시작할 때엔 학습을 중단한다.



완전연결 신경망 학습 – cont.

<u>가중치 정규화</u>

- $\mathcal{L}'(x, y) = \mathcal{L}(x, y) + \alpha\|\theta\| \quad \begin{pmatrix} \|\theta\|_2 = \sqrt{\sum_{i=1}^{N} \theta_i^2} \\ \|\theta\|_1 = \sum_{i=1}^{N} |\theta_i| \end{pmatrix}$

완전연결 신경망 학습 – cont.

가중치 정규화

- $\mathcal{L}'(x,y) = \mathcal{L}(x,y) + \alpha\|\theta\|$

$$\left(\begin{array}{l} \|\theta\|_2 = \sqrt{\sum_{i=1}^{N} \theta_i^2} \\ \|\theta\|_1 = \sum_{i=1}^{N} |\theta_i| \end{array}\right)$$

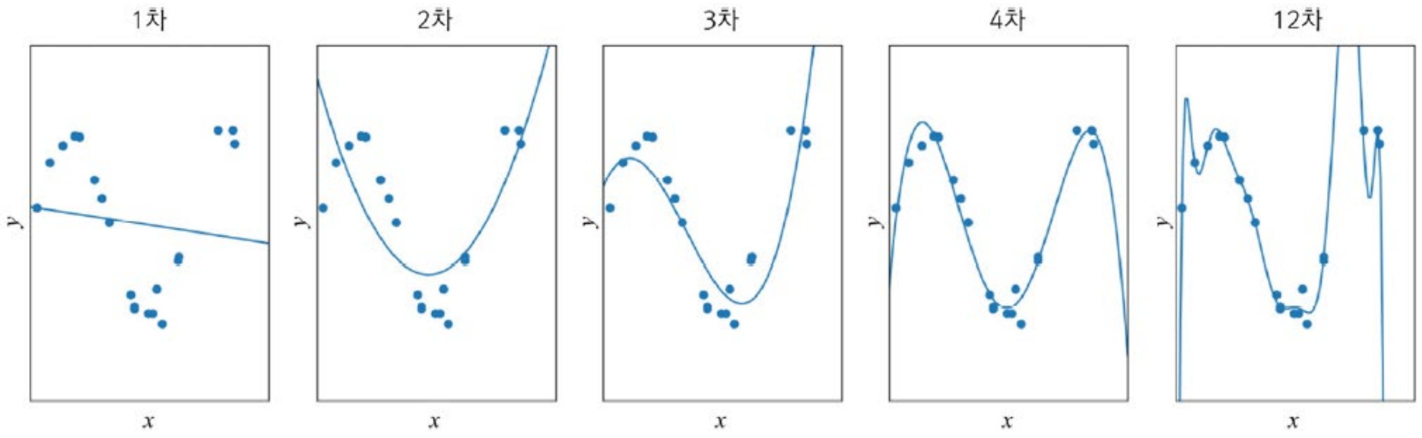


그림 1-13 과소적합과 과잉적합 현상

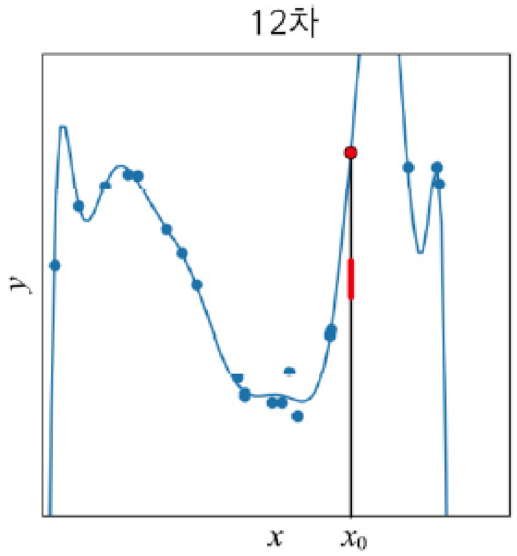


그림 1-14 과잉적합되었을 때 부정확한 예측 현상

수고하셨습니다.