

# 9장 신경망: 인간의 뇌를 컴퓨터로 모방하기

## HOW A DEEP NEURAL NETWORK SEES

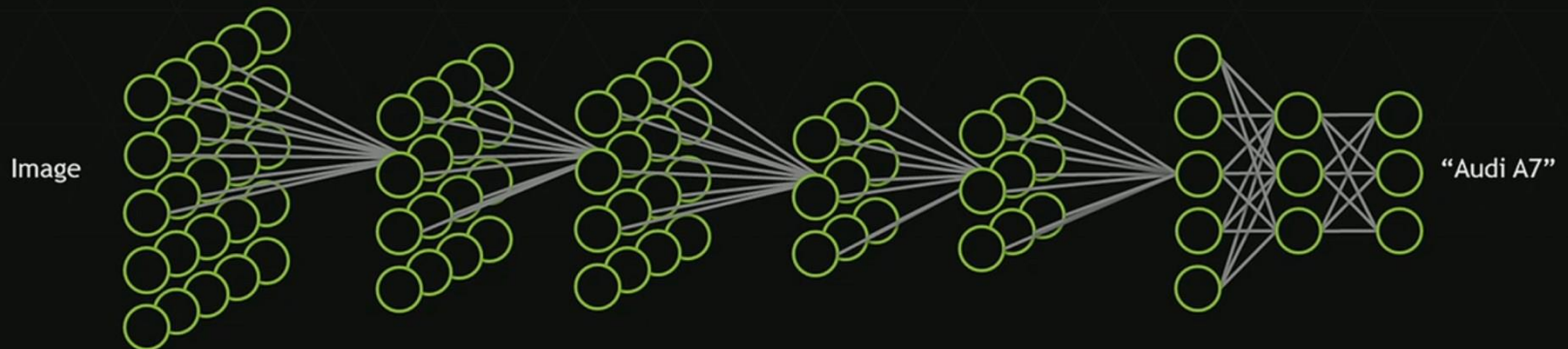
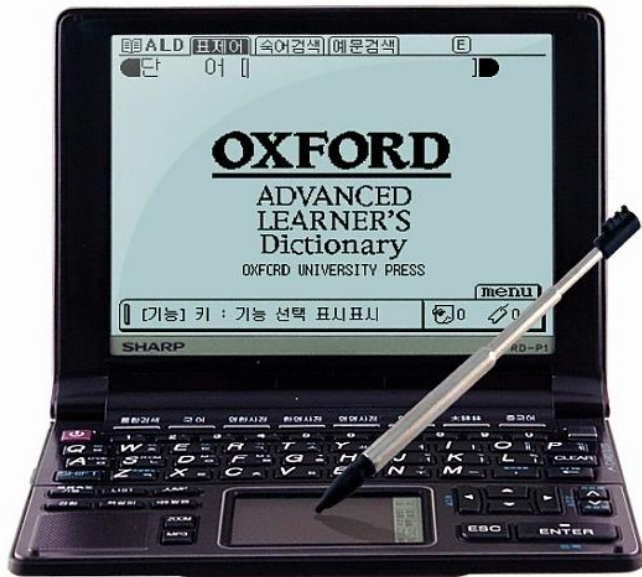


Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng.

출처: <https://www.extremetech.com/extreme/215170-artificial-neural-networks-are-changing-the-world-what-are-they>

## 9장 신경망: 적용 사례



### 네이버 일본어사전

+ Mobile



일본 한자

|   |   |   |   |
|---|---|---|---|
| 市 | 布 | 芋 | 折 |
| 艾 | 支 | 扱 | 扞 |

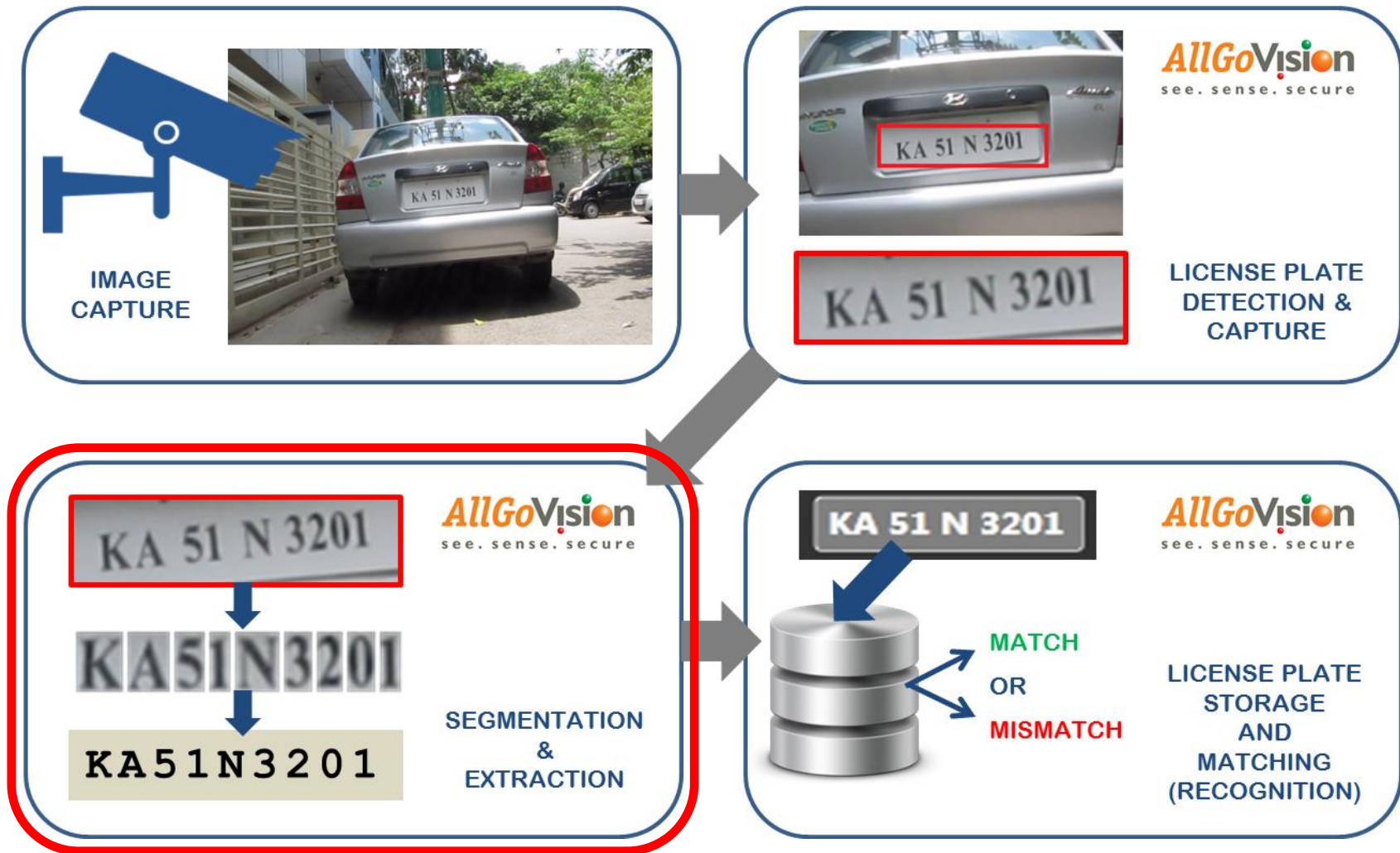
하라가나/가타가나

|   |   |   |   |
|---|---|---|---|
| あ | あ | ま | な |
|   |   |   |   |

1획 지우기    모두 지우기

\*마우스로 드래그하면 위치이동이 가능합니다.

## 9장 신경망: 적용 사례



출처: <http://www.allgovision.com/license-plate-recognition.php>

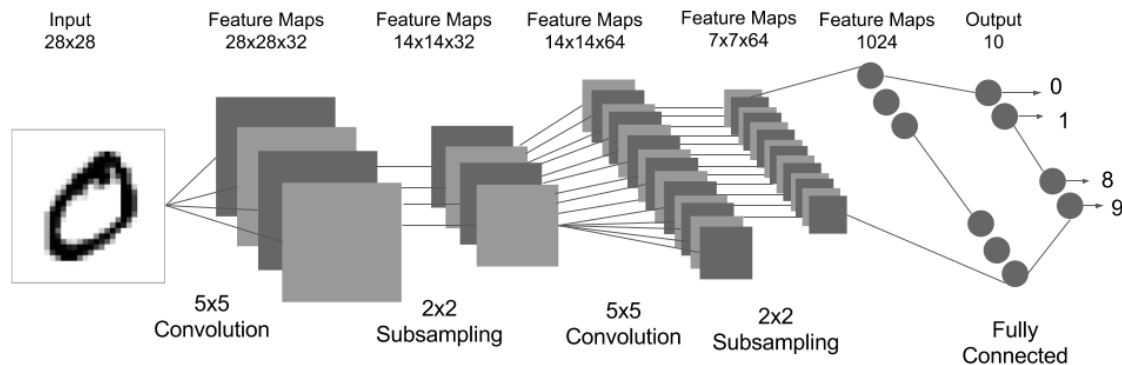
## 9장 신경망: 적용 사례



### MNIST 데이터 수

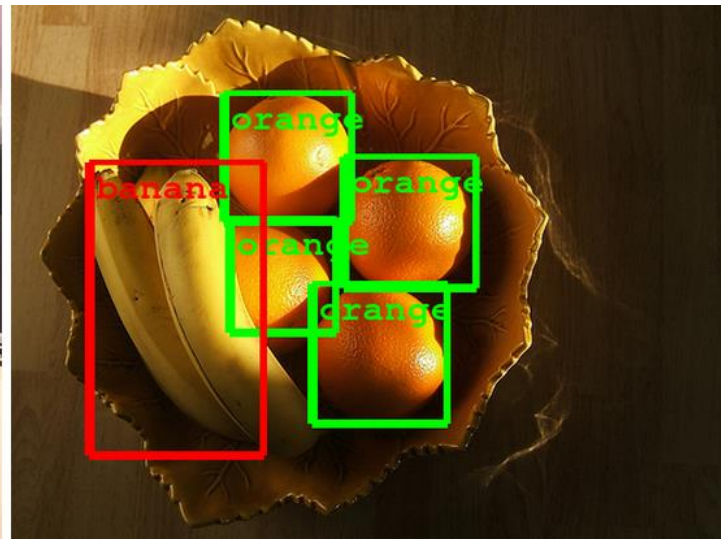
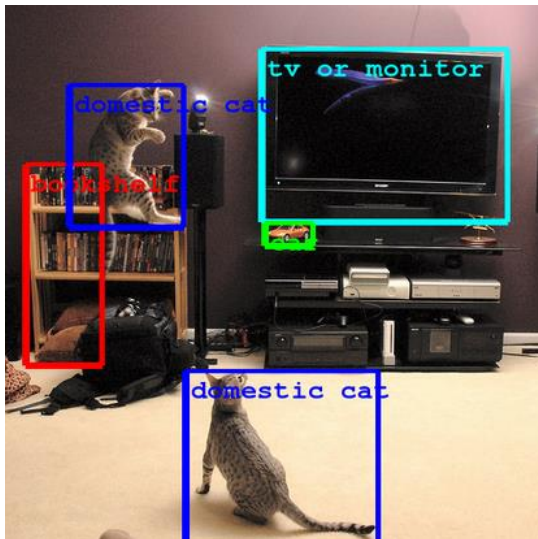
학습 셋 이미지 – 55,000개의 트레이닝 이미지,  
5,000개의 검증 이미지(28 x 28 해상도의 gray)

테스트 셋 이미지 – 10,000개의 이미지



출처: <http://solarisailab.com/archives/1308>

## 9장 신경망: 적용 사례





## 9장 신경망: 적용 사례

---

- **Convolutional Neural Networks(CNNs)**
  - 딥 러닝의 한 종류로 주로 이미지를 인식하는데 사용됨.
  - Variations of multilayer perceptron
- **Recurrent Neural Networks(RNNs)**
  - Sequence prediction을 위한 ANNs.
  - 자연어 처리, 음성인식에 뛰어난 성능을 보이는 인기 모델
  - 문장에서 다음에 나올 단어를 추측하고 싶다면 이전에 나온 단어들을 아는 것이 큰 도움이 됨.
  - 가장 많이 사용되는 모델 : LSTM(Long / Short Term Memory)

## 9장 신경망: 적용 사례

### Alpha Go Design

- ❖ According to paper in Nature
- ❖ Not yet known what changed over the last 5 months, other than much more self-play

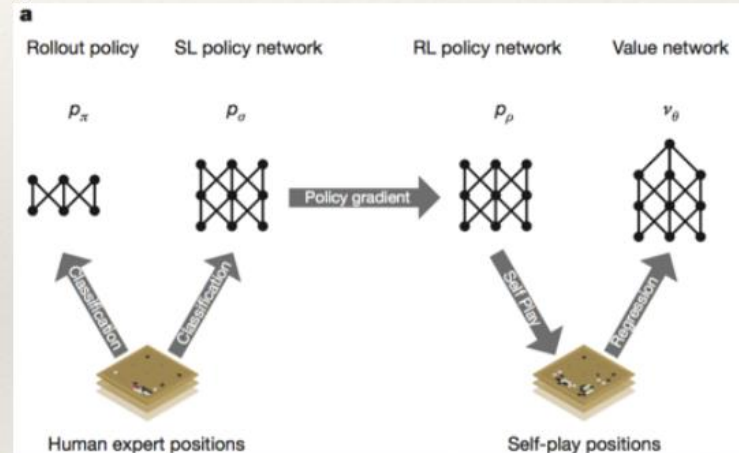
- ❖ Search: MCTS (normal)

- ❖ Simulation (rollout) policy: relatively normal

- ❖ Supervised Learning (SL) policy from master games: improved in details, more data

- ❖ **New: Reinforcement Learning (RL) from self-play for value network**

- ❖ **New: Reinforcement Learning (RL) from self-play for policy network**



## 9장 신경망: 적용 사례

# 콘텐츠 추천 알고리즘

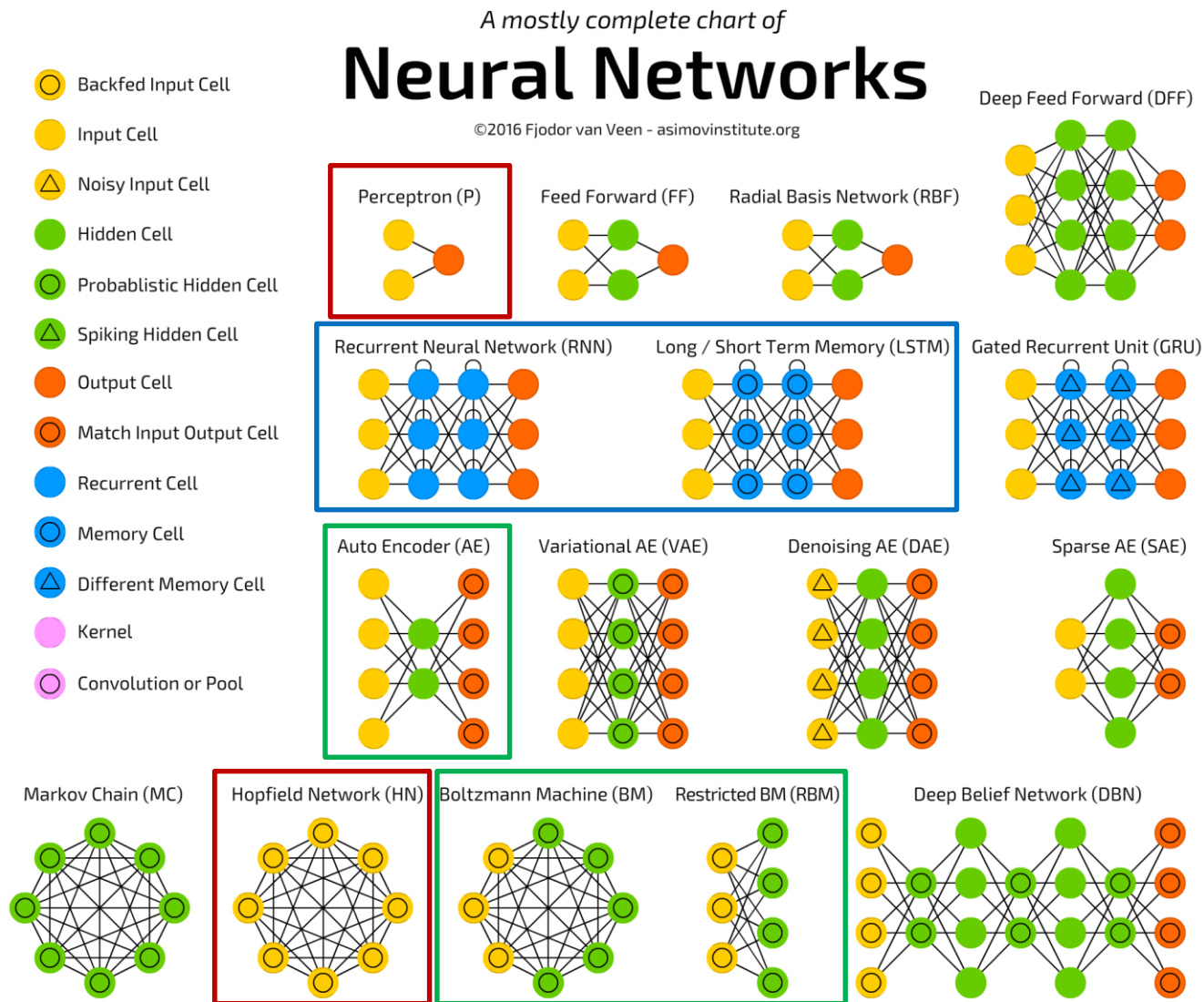
### 콘텐츠 기반 필터링

콘텐츠 기반 필터링은 위와 같은 협업 필터링과는 다른 방법으로 추천을 구현하는 방법이다. 협업 필터링이 사용자의 행동 기록을 이용하는 반면, 콘텐츠 기반 필터링은 항목 자체를 분석하여 추천을 구현한다. 예를 들어 음악을 추천하기 위해 음악 자체를 분석하여 유사한 음악을 추천하는 방식이다.

이 기법은 콘텐츠의 내용을 분석해야 하므로 아이템 분석 알고리즘이 핵심적이며, 이를 위해 군집분석(Clustering analysis), 인공신경망(Artificial neural network), tf-idf(term frequency-inverse document frequency) 등의 기술이 사용된다.

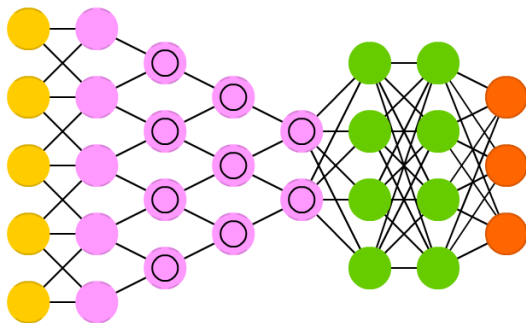


# 9장 신경망: 종류

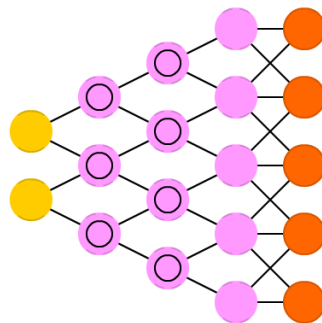


# 9장 신경망: 종류

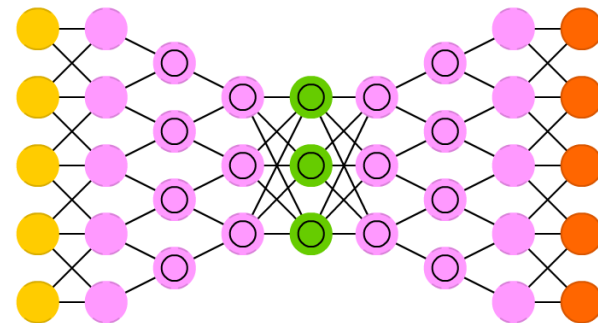
Deep Convolutional Network (DCN)



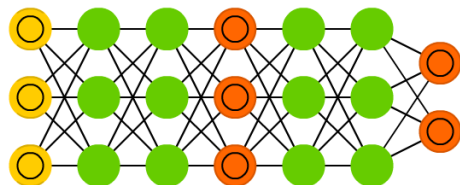
Deconvolutional Network (DN)



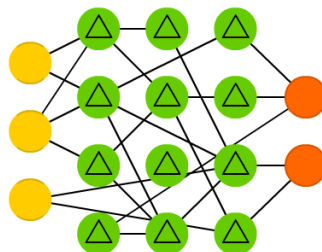
Deep Convolutional Inverse Graphics Network (DCIGN)



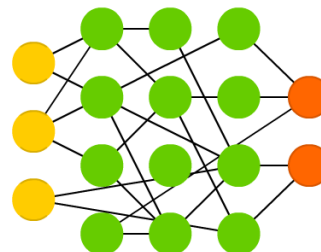
Generative Adversarial Network (GAN)



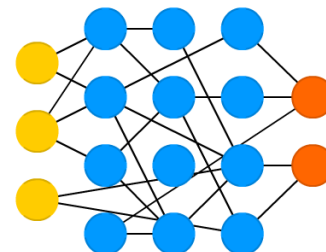
Liquid State Machine (LSM)



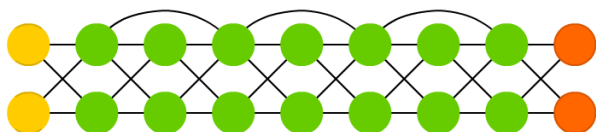
Extreme Learning Machine (ELM)



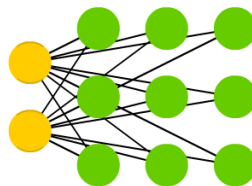
Echo State Network (ESN)



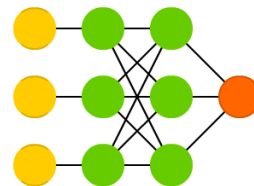
Deep Residual Network (DRN)



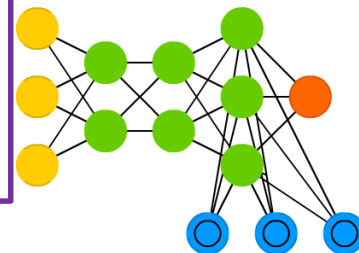
Kohonen Network (KN)



Support Vector Machine (SVM)



Neural Turing Machine (NTM)



# 9장 신경망: 인간의 뇌를 컴퓨터로 모방하기

---

- **학습 목표**

- 체험해 봅시다: Ex1\_Perceptron문자인식.xlsm

Ex2\_홉필드 네트워크 문자 인식.xlsm

비뚤어진 문자 인식

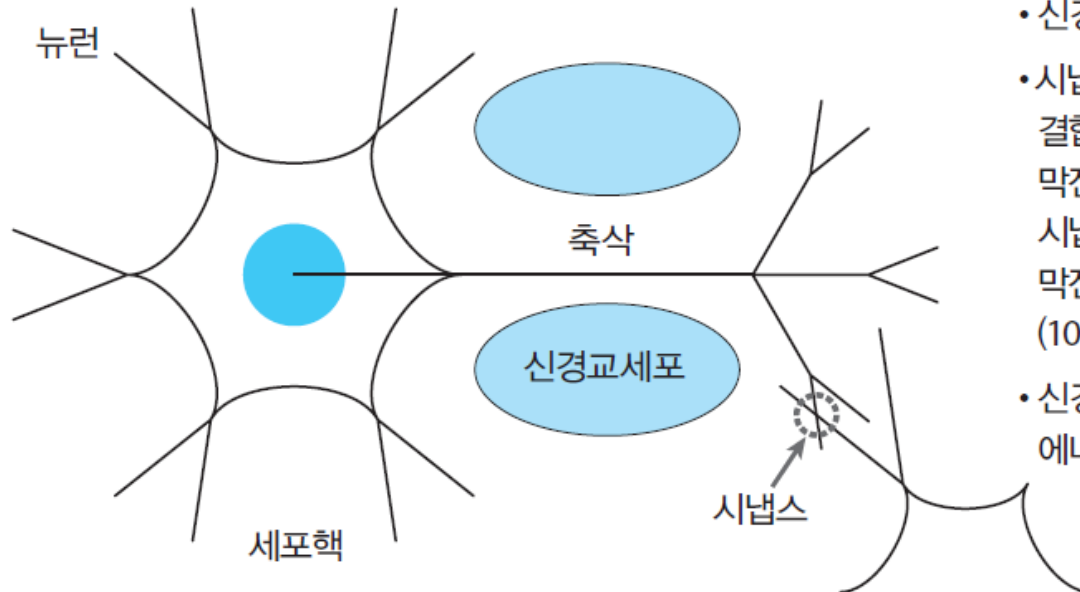
- 뇌의 모델과 신경망의 개념
  - 퍼셉트론
  - 홉필드 네트워크
  - 기타 신경망

## 9.1.1 뇌의 모델화

---

- 인간의 뇌는 뉴런이라는 신경 세포의 연결로 이루어져 있음.
- 이 절에서는 **인간의 뇌를 컴퓨터로 모방하기 위한 모델인 신경망**을 고찰.
- 인간의 뇌는 많은 신경 세포의 연결로 이루어져 있으며, 이것을 모델화한 것.
  - ✓ **맥컬러 · 피츠 모델(Warren McCulloch & Walter Pitts 1943).**
- 하나의 뉴런에 시냅스를 통하여 다수의 뉴런으로부터 입력 신호가 들어오며, 그 신호의 합이 일정한 강도를 초과하면 그 뉴런은 발화하고 축삭을 통하여 자기와 연결된 다른 뉴런에게 신호를 출력한다는 것.

## 9.1.1 뇌의 모델화



- 신경 세포(뉴런): 140억 개
- 시냅스 결합: 1,000~200,000  
결합 상태가 정신적 활동을 나타냄.  
막전위: 뉴런 내는 외부보다 70mV 낮음.  
시냅스 경유 신호가 막전위를 상승시킴  
막전위 90mV에서 펄스 발생  
(100mV/1ms)
- 신경교세포: 1,000억 개  
에너지 보급, 발화 제어

그림 2-1 뉴런 모델



## 9.1.1 뇌의 모델화

매컬러·피츠 모델

$$\sum_{i=1}^n w_i x_i$$

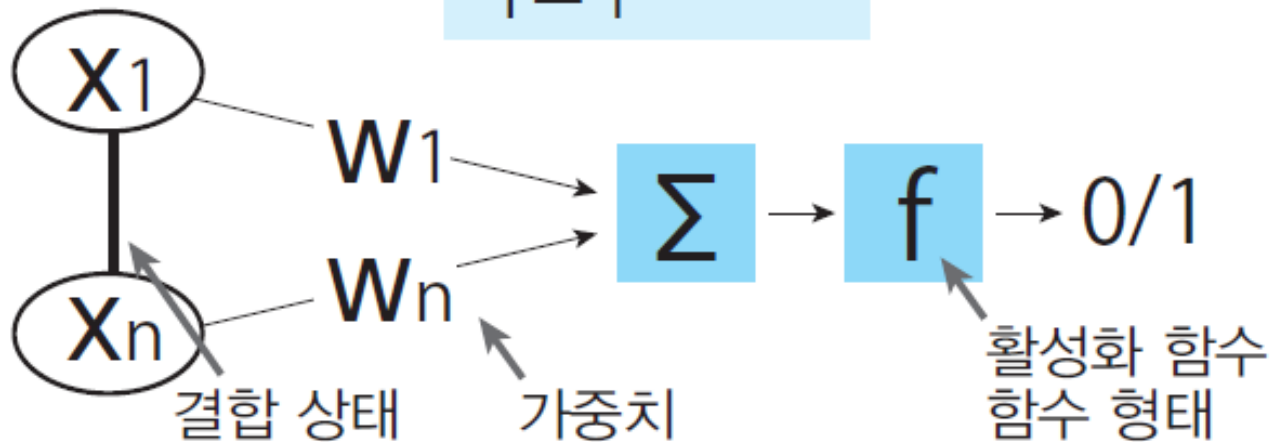


그림 2-2

매컬러·피츠 모델

## 9.1.1 뇌의 모델화

- 이것을 수식화하면 다음 식으로 표현

식 2-1

$$\text{출력} = f(\sum W_i X_i)$$

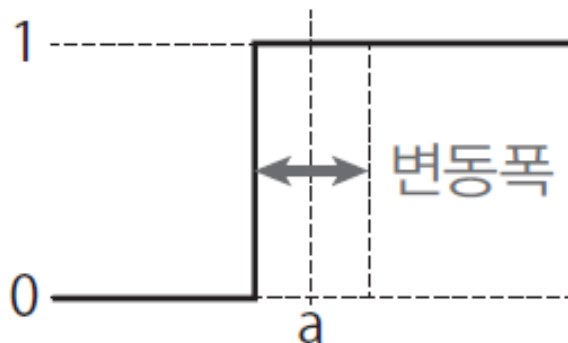
$W_i$ : 뉴런 간의 결합 강도       $X_i$ : 각 뉴런으로부터의 입력

- 여기서 **f**는 **활성화 함수**이며, 입력 신호의 총합으로부터 발화 여부를 결정.
- 가장 간단한 것이 **임계치 함수**로서 총합이 일정한 값을 초과하면 발화하고 일정한 값 이하라면 발화하지 않음.

## 9.1.1 뇌의 모델화

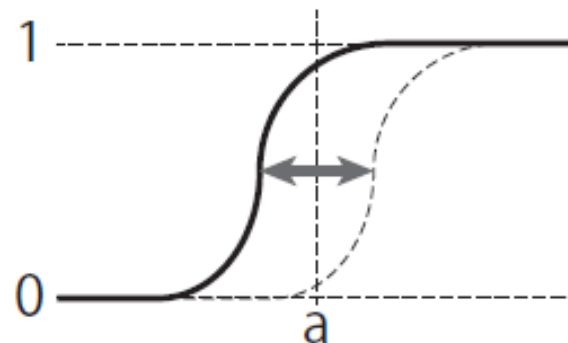
### 활성화 함수

임계치 함수



$$f : \begin{cases} \text{if } \sum wx \geq a \rightarrow 1 \\ \text{else } 0 \end{cases}$$

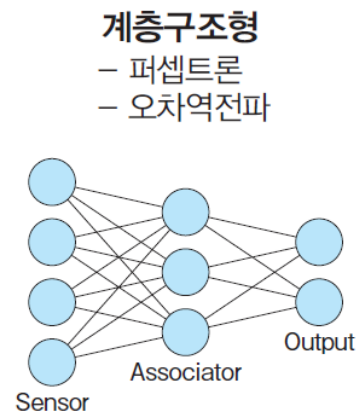
시그모이드



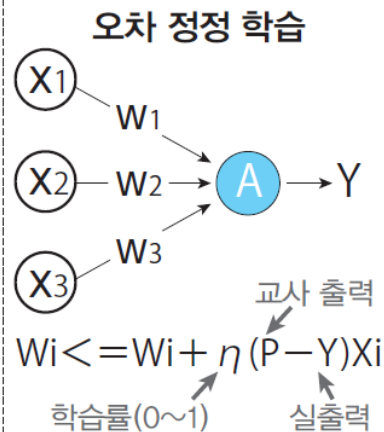
$$f : \frac{1}{1 + e^{\alpha(a - \sum wx)}} \quad (0 \leq \alpha < \infty)$$

## 9.1.2 신경망의 구조

- 신경망은 뉴런을 컴퓨터 소자에 의한 **노드로** 나타내고 **노드 간을 신호 선으로 연결한 것으로** 만들어짐.
- 노드를 연결하는 방법**에는 크게 두 가지 종류의 형태가 있음.



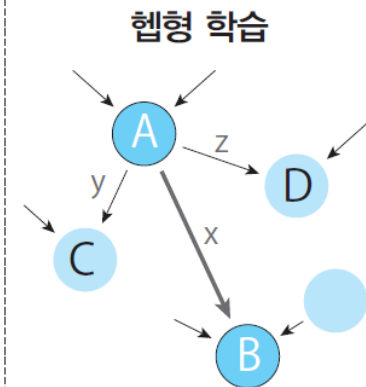
- 인접 계층 간 완전 결합, 계층 내 결합 없음
- 기대 출력이 되도록 가중치 변경



- 가중치  $W$ 가 변화하지 않을 때(즉,  $Y=P$ , 오차 0)까지 반복



- 노드 간 쌍방향 완전 결합
- 에너지 함수 극소화
- 어닐링(annealing)에 따른 최적화



- A, B가 함께 발화하면 양자 간의 가중치  $x$ 가 증가
- C, D가 발화하지 않으면  $y$ 와  $z$ 는 그대로 유지 (또는 감소)

그림 2-3    신경망의 형태와 학습법

## 9.1.2 신경망의 구조

---

### ❖ 계층구조형

- 노드를 계층적으로 나열하고 각 계층 간의 노드는 완전 결합으로 구성하지만 계층 내의 노드는 연결이 없음.
- 계층구조형은 선의 수가 적고 신호의 흐름에 방향성이 생기므로 인간 뇌의 뉴런 신호 흐름에 가까운 네트워크.
- 이 개념에 기초하여 최초로 만들어진 것이 퍼셉트론 (Frank Rosenblatt, 1958)이지만 적용에 한계가 있어, 그 이후 상호결합형 구조가 탄생.



## 9.1.2 신경망의 구조

---

### ❖ 상호결합형

- 모든 노드를 대등하게 연결. 완전 결합으로는 선이 너무 많아지기 때문에 선택적으로 연결하는 것도 있음.
- 신호 흐름에 방향성이 없고 각 노드의 값과 노드 간의 가중치(결합 강도)를 네트워크 전체적으로 안정화하는 방향으로 변화시켜 나간다는 개념.
- 이것은 뇌의 모방과는 관계없이, 연상이나 조합 최적화와 같은 **인간의 뇌 활동에 가까운 동작이 이루어지고** 적용 범위가 크게 넓어짐.
- 상호결합형의 대표는 **홉필드 네트워크(John Hopfield, 1982)**

## 9.1.3 신경망의 학습

---

- 신경망은 '**학습**'과 '**리콜**'이라는 연산을 반복하며 동작.
- 학습은 노드 간의 가중치와 활성화 함수의 형태를 변화시키는 것.
  - ✓ 네트워크에 기억해야 할 상태가 결정되어 있는 경우에는 이것을 교사 신호로 입력하고, 기대하는 출력을 얻을 수 있도록 가중치와 활성화 함수를 변경.
  - ✓ 교사 신호가 없는 경우에는 노드 간의 발화 상태가 네트워크 전체의 안정화에 따르도록 가중치 등을 변경.
- 교사 신호의 필요 여부에 따라 지도 학습, 비지도 학습으로 나뉨.
  - ✓ 오차 정정 학습: 교사 신호를 입력할 때의 기대 출력과 실제 출력이 일치하도록 변경한다(지도 학습, supervised learning).
  - ✓ 헵형 학습: 인접한 뉴런이 함께 발화하면 그 결합 가중치가 증가하도록 변경한다(비지도 학습, unsupervised learning).

## 9.1.4 신경망의 리콜

---

- 학습이 끝난 신경망에 임의의 데이터(노드의 초기 상태)를 입력하면 네트워크가 동작하고 어떤 출력(노드의 최종 상태)를 얻을 수 있음.

→이 과정을 **리콜(recall)**이라고 함.

- **계층형의 리콜은 1회 리콜 연산으로** 교사 신호대로 출력이 얻어지면 리콜 성공, 얻어지지 않으면 리콜 실패.
- 입력이 교사 신호와 완전히 일치하는 경우에는 반드시 교사 신호대로 출력이 얻어짐.

## 9.1.4 신경망의 리콜

---

- **입력이 교사 신호와 다른 경우**에는 어떤 하나의 교사 신호에 일치하는 출력이 얻어질 수도 있고, 예기치 않은 출력이 나올 수도 있음.
- 입력이 교사 신호와 달라도 출력이 교사 신호와 일치하는 경우  
→ **연상(association)**이라고 함.
- **상호결합형의 리콜은 일반적으로 네트워크 전체가 안정 상태가 될 때까지 리콜 연산을 반복.**
- **네트워크의 안정 상태는 리콜 연산을 반복해도 각 노드의 값이 변화하지 않는 상태이며, 그것이 기억 내용과 일치하면 리콜 성공, 그렇지 않으면 리콜 실패.**

## 9.1.4 신경망의 리콜

---

- 실패할 경우에는 리콜 연산이 수렴하지 않고 몇 번이나 동일한 노드 상태가 반복되는 상태에 빠질 수도 있음.
- 또 수렴은 하더라도 기대하는 노드 상태가 아닐 수도 있음. 이런 경우를 국소해라고 부르며, 보다 바람직한 최적해를 얻기 위한 연구를 수행한 **볼츠만 머신(Geoffrey Hinton, 1985)** 등이 고안.
- **상호결합형**은 리콜 연산을 반복하기 때문에 **연상이 아주 우수**하며 꽤 비틀어진 입력에 대해서도 어떤 연상 결과가 얻어지므로, **번거로운 패턴 인식이나 특징 추출 대신에 비슷한 것을 파악하는 용도로** 사용될 수 있음.



## 9.2 퍼셉트론

---

- 퍼셉트론(Perceptron)은 로젠블라트(Frank Rosenblatt, 1958)가 고안한 가장 초보적인 계층형 네트워크.
- 구조는 수용기(sensory unit), 연합기(association unit), 응답기(response unit)라는 노드 계층으로 구성된 3개 층의 구조로, 수용기와 연합기는 가중치 고정의 완전 결합, 연합기와 응답기는 가중치 가변의 완전 결합.
- 신호는 수용기로부터 연합기를 거쳐 응답기 쪽으로 한쪽 방향으로만 흐름.
- 수용기의 각 노드에 값을 설정하는 것이 네트워크로의 입력이며, 출력은 응답기의 각 노드 값으로 얻어짐.

## 9.2.1 퍼셉트론의 가중치 학습

---

- 퍼셉트론의 학습은 교사 신호 입력에 대해 기대 신호가 출력되도록 연합기와 응답기 간의 가중치를 변화시키는 것.
- 기대 출력이 1인데 0이 출력될 때는 가중치를 증가, 반대의 경우에는 가중치를 감소시키는 방식으로 처리를 반복하여 가중치 학습이 진행되고 여러 교사 신호에 대응하는 가중치가 결정.
- 그러나 복수의 교사 신호를 동시에 처리하는 것은 상호 간에 영향을 주기 때문에 가중치 조정이 어려움.
- 다행히 교사 신호가 선형 분리 가능(?)이라면 학습은 반드시 수렴한다는 학습 수렴 정리가 있음.
- 반대로 선형 분리가 불가능한 경우에는 가중치 변화가 계속 들쭉날쭉하여 학습이 수렴하지 않을 가능성이 높음.

## 9.2.2 퍼셉트론 가중치 학습의 구체적 예

- 연합기 3개 노드, 응답기 2개 노드의 퍼셉트론으로 가중치 학습

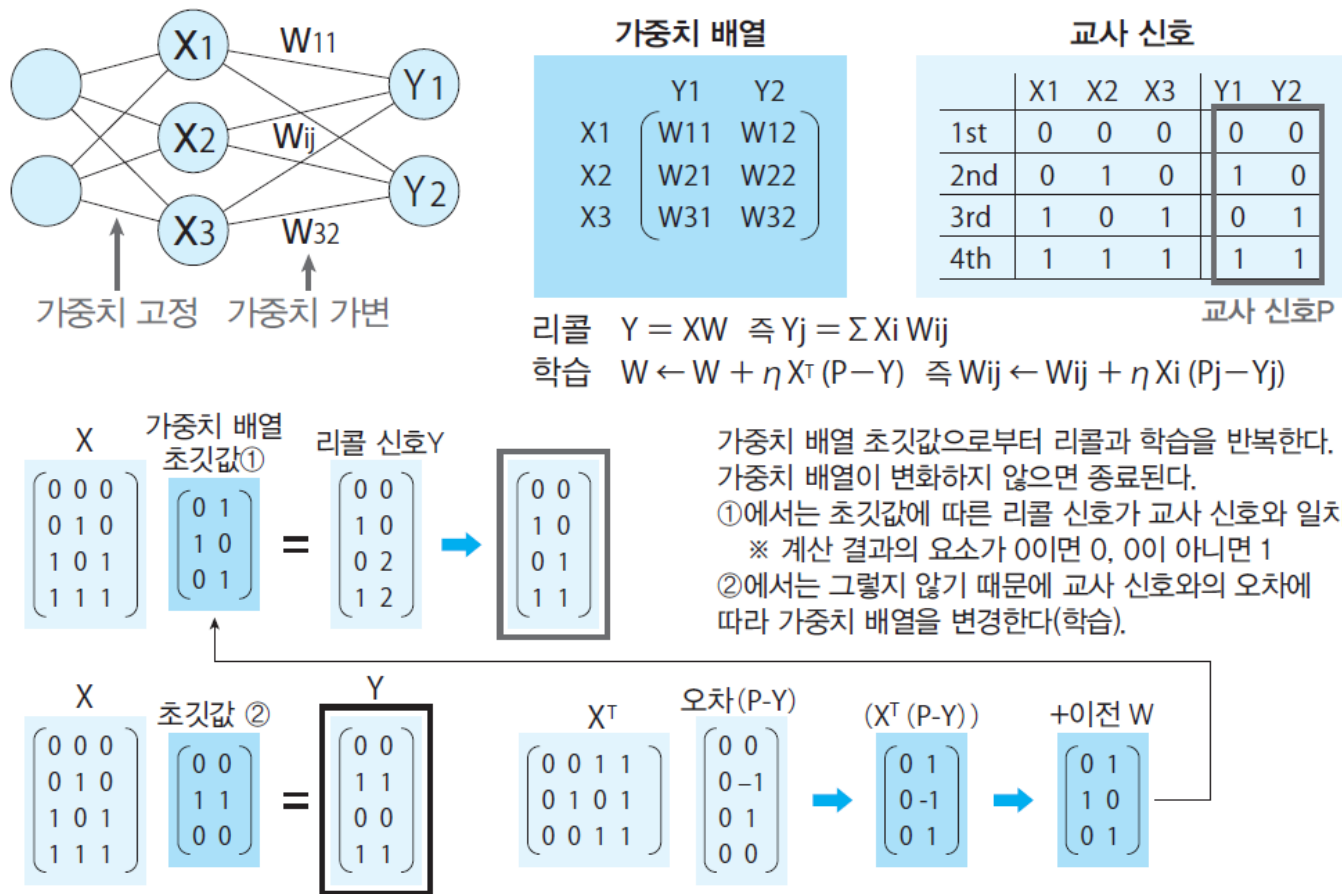


그림 2-4 퍼셉트론 학습

## 9.2.3 오차 정정 학습

- 가중치 학습을 수행해 보자. 우선 임의의  $3 \times 2$  행렬  $W$ 를 준비하고, 다음 식을 사용하여  $Y$ 를 구하면 교사 신호 출력( $P$ )과 다른 결과가 얻어질 수도 있음.

식 2-2

$$Y = f(XW)$$

식 2-2는 식 2-1을 교사 신호의 개수만큼 한 번에 계산하도록 확장한 형태다.

- 이 경우에는 두 행렬의 차( $P-Y$ )를 구하고 이것을  $W$ 에 반영하여 다시 위의 식으로 계산하여  $P=Y$ 가 될 때까지 반복.
- $P-Y$ 가 오차이므로 이것이 0이 되도록 가중치를 조정해 나간다는 의미에서 **오차 정정 학습**이라고 함.
- 오차 정정**에는 보통 다음과 같은 식을 사용.

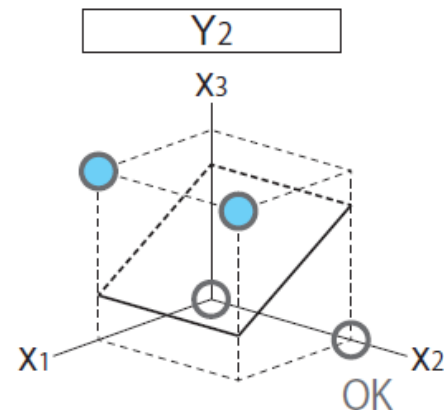
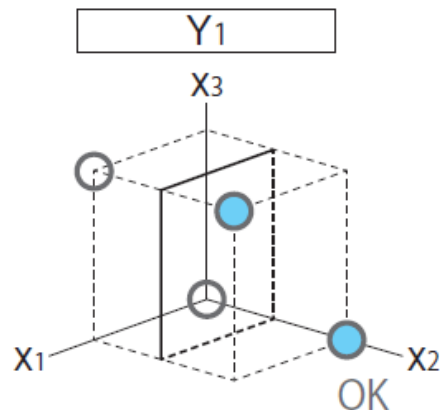
식 2-3

$$W \leftarrow W + \eta X^T(P-Y) \quad \eta: \text{학습률}(0 < \eta \leq 1), X^T: X \text{의 전치행렬}$$

## 9.2.4 선형 분리 가능성

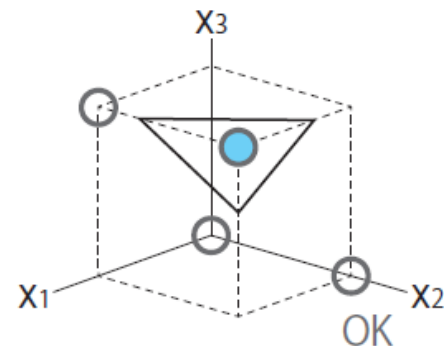
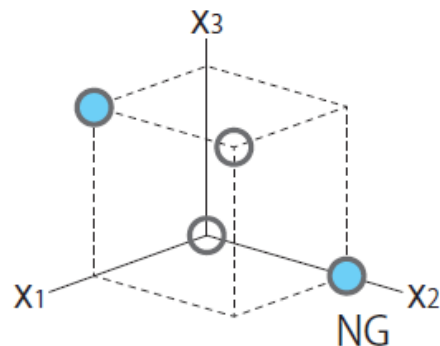
a. 선형 분리가 가능한 교사 신호

|     | X1 | X2 | X3 | Y1 | Y2 |
|-----|----|----|----|----|----|
| 1st | 0  | 0  | 0  | 0  | 0  |
| 2nd | 0  | 1  | 0  | 1  | 0  |
| 3rd | 1  | 0  | 1  | 0  | 1  |
| 4th | 1  | 1  | 1  | 1  | 1  |



b. 선형 분리가 불가능한 교사 신호

|     | X1 | X2 | X3 | Y1 | Y2 |
|-----|----|----|----|----|----|
| 1st | 0  | 0  | 0  | 0  | 0  |
| 2nd | 0  | 1  | 0  | 1  | 0  |
| 3rd | 1  | 0  | 1  | 1  | 0  |
| 4th | 1  | 1  | 1  | 0  | 1  |



a의 교사 신호는 X를 3차원 공간으로 두었을 때 Y의 각 요소가 평면으로 분리 가능하므로 선형 분리 가능.

b의 교사 신호는 동일하게 생각할 경우 Y1의 요소가 평면으로 분리되지 않으므로 선형 분리 불가능.

b의 교사 신호로 가중치 학습을 수행하면 오차 정정을 반복해도 오차가 0이 되도록 하는 W가 구해지지 않음.

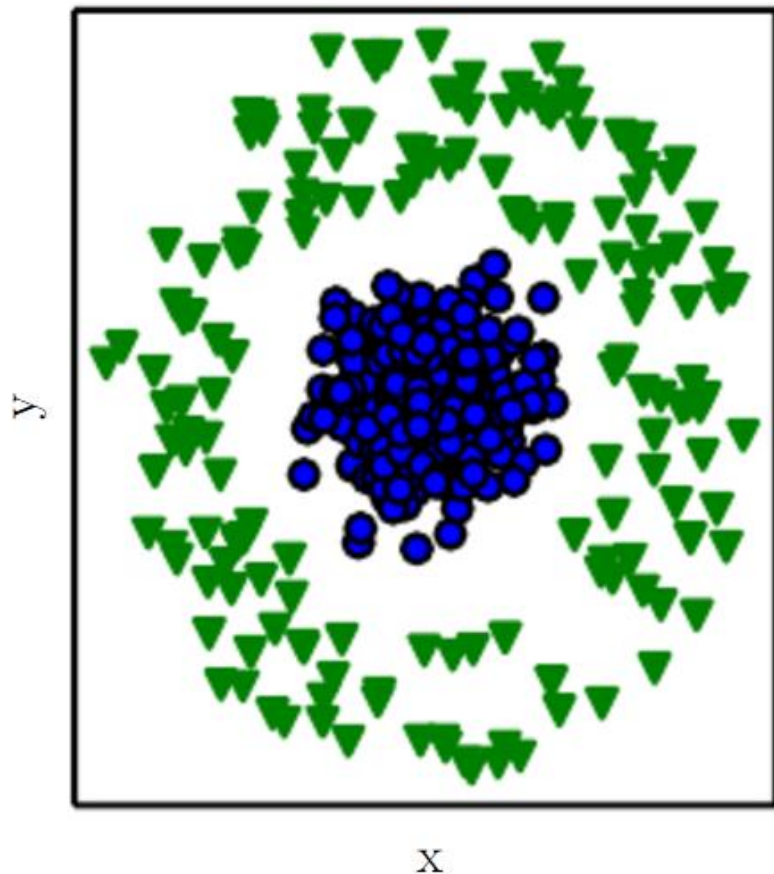
이 경우에는 교사 신호를 기억하는 것이 불가능하다.

그림 2-5 선형 분리 가능성

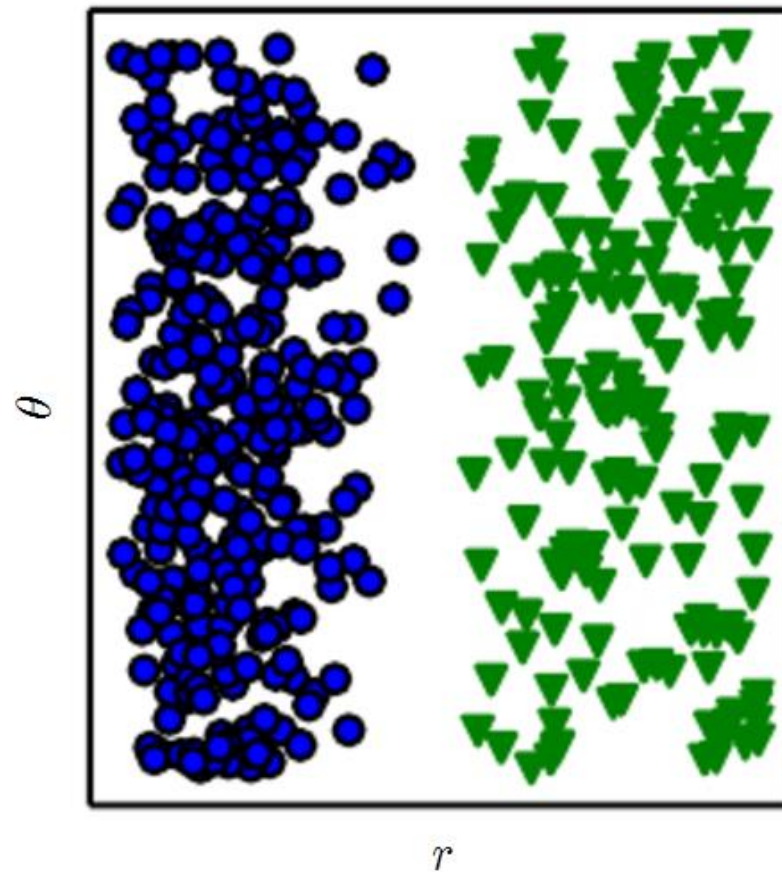


## 9.2.4 선형 분리 가능성

Cartesian coordinates

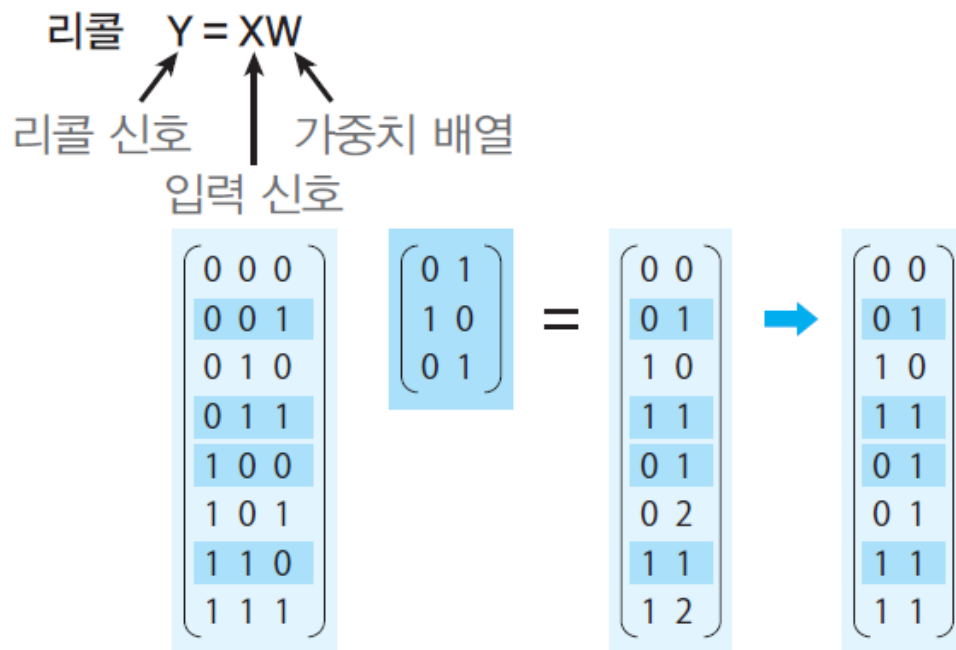


Polar coordinates



출처: <http://www.deeplearningbook.org/contents/intro.html>

## 9.2.5 퍼셉트론의 리콜



교사 신호

|     | X1 | X2 | X3 | Y1 | Y2 |
|-----|----|----|----|----|----|
| 1st | 0  | 0  | 0  | 0  | 0  |
| 2nd | 0  | 1  | 0  | 1  | 0  |
| 3rd | 1  | 0  | 1  | 0  | 1  |
| 4th | 1  | 1  | 1  | 1  | 1  |

교사 신호 이외의 입력 신호에 대해서는 가까운 신호를 리콜한다.

그림 2-6 퍼셉트론의 리콜

## 9.2.6 퍼셉트론에 의한 문자 인식

- 퍼셉트론을 보다 구체적으로 살펴보기 위한 간단한 문자 인식

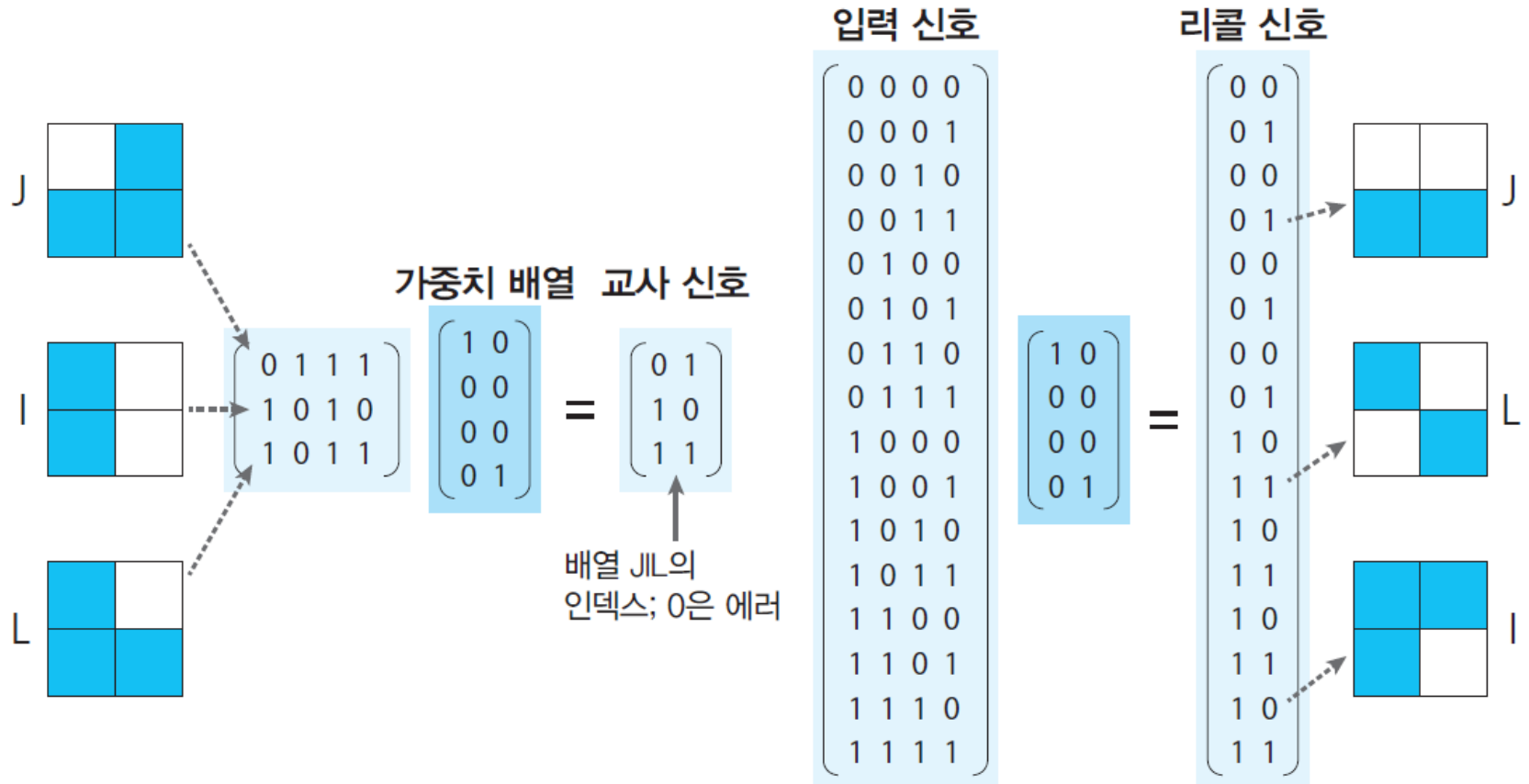


그림 2-7 퍼셉트론에 의한 문자 인식

## 9.2.6 퍼셉트론에 의한 문자 인식

---

- 체험해 봅시다: [Ex1\\_Perceptron문자인식.xlsx](#)  
-> 재수행을 통해 처리 원리를 이해

## 9.3 홉필드 네트워크

- **홉필드 네트워크(Hopfield network)**는 홉필드(John Hopfield, 1982)가 고안한 상호결합형 네트워크.
- 구조는 각 노드를 쌍방향으로 완전 결합한 것으로 노드 값에 따라서 다음과 같은 에너지 함수를 정의.

식 2-4

$$E = -1/2 \sum_{ij} w_{ij} X_i X_j$$

$w_{ij}$ : 노드  $i$ 와  $j$  간의 가중치       $X_i, X_j$ : 노드  $i, j$ 의 값

### 9.3.1 홉필드 네트워크의 가중치 배열

---

- 에너지 함수는 각 노드가 네트워크의 기억 항목에 해당하는 값이 될 때 최소(또는 극소)가 되도록 정의.
- 이 때문에 가중치 배열  $W = [w_{ij}]$ 는 다음과 같은 헵형 학습에 따라 얻어짐.
- 즉, 기억 항목에 해당하는 각 노드 값을 보고 다음 사항을 실행.
  - 값이 동일한 노드 간의 가중치를 증가
  - 값이 다른 노드 간의 가중치를 감소
- $W$ 의 각 요소  $w_{ij}$ 는 노드  $i$ 와  $j$  간의 가중치를 나타내며 신호의 방향은 무관하므로  $w_{ij}=w_{ji}$ 가 되어 **가중치 배열은 대칭 행렬**이 됨.
- 또한, 가중치 배열의 대각 성분은 동일한 노드 간의 관계이기 때문에  **$w_{ii}=0$ 으로 정의.**

## 9.3.2 홉필드 네트워크의 리콜

- 리콜 시에는 기본적으로는 식 2-5에 따라 리콜 연산을 실행하지만 퍼셉트론과는 달리 **리콜 연산이 반복**.
- 이 과정에서 에너지 함수를 극소화시키는 방향으로 각 노드 값이 변경되며, **에너지 함수 값이 극솟값이 되어 노드 값이 변화하지 않으면 종료**.
- **최종적인 노드 상태가 기억 항목에 일치하면 리콜 성공, 그렇지 않으면 리콜 실패**.
- 리콜 연산은 시계열적인 의미를 포함하며 다음과 같음.

식 2-5

$$X(t+1) = f(W \cdot X(t))$$

$W$ : 가중치 배열,  $X(t)$ : 시계열  $t$ 에서의 노드 상태

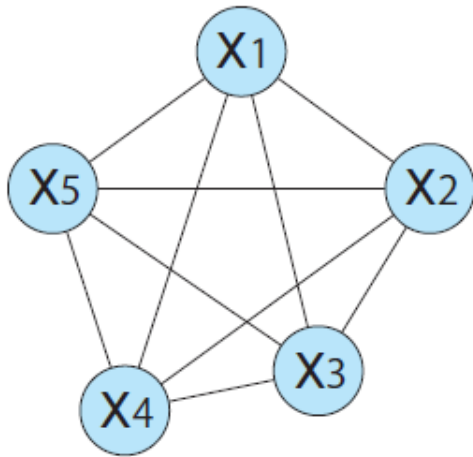
## 9.3.2 홉필드 네트워크의 리콜

---

- **홉필드 네트워크**는 리콜 연산을 반복하기 **때문에 연상과 조합 최적화 문제에 적합**.
- 퍼셉트론에서의 선형 분리 가능성이라는 제약이 없어서 적용 범위도 넓지만 **에너지 함수의 정의에는 연구가 필요**.
- **단점**
  - ✓ 리콜 연산에서 에너지 함수의 극솟값이 일단 얻어지면, 더 좋은 해 (최적해)가 있다고 해도 그 해를 얻을 수 없을 가능성이 있음.
  - ✓ 노드 수에 비해 기억 항목 수가 너무 많으면 잘 동작하지 않음.
- 이런 단점들을 해결하는 것이 뒷부분에서 설명하는 **볼츠만 머신**.



### 9.3.3 홉필드 네트워크의 가중치 배열과 리콜의 구체적 예



에너지 함수

$$\varepsilon = -\frac{1}{2} \sum_{i,j=1}^5 W_{ij} X_i X_j$$

$W_{ij}$ : 노드,  $i, j$  간의 가중치  
 $X_i, X_j$ : 노드  $i, j$  의 값

가중치 배열

$$W_{ij} = \sum_{k=1}^n X_i(k) X_j(k)$$

$W_{ij}$ : 노드,  $i, j$  간의 가중치  
 $X_i(k)$ : 노드  $i$  의  $k$  번째 값  
 $n$ : 기억 패턴 수

기억 패턴

|     | X1 | X2 | X3 | X4 | X5 |
|-----|----|----|----|----|----|
| 1st | 1  | -1 | 1  | -1 | 1  |
| 2nd | -1 | 1  | 1  | -1 | -1 |

가중치 배열

$$W = \begin{bmatrix} 0 & -2 & 0 & 0 & 2 \\ -2 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ 2 & -2 & 0 & 0 & 0 \end{bmatrix}$$

리콜 방법

$$X_i(t+1) = f \left( \sum_{j=1}^5 W_{ij} X_j(t) \right)$$

↑  
임계치 함수

※ 결과가 양수이면 1, 0 이하이면 -1을 출력  
 기억 패턴과 일치할 때까지 반복 계산

## 9.3.3 홉필드 네트워크의 가중치 배열과 리콜의 구체적 예

### 리콜 예

| 초기 패턴                         | 초기 에너지             | 리콜 연산  | 양수이면 1, 0 또는 음수이면 -1 | 리콜 후 에너지                         |
|-------------------------------|--------------------|--|----------------------|----------------------------------|
| $A = [1 \ -1 \ 1 \ -1 \ 1]$   | $\varepsilon = -8$ | $WA = [4 \ -4 \ 2 \ -2 \ 4] \Rightarrow [1 \ -1 \ 1 \ -1 \ 1] = A$     |                      | $\varepsilon = -8$               |
| $B = [-1 \ 1 \ 1 \ -1 \ -1]$  | $\varepsilon = -8$ | $WB = [-4 \ 4 \ 2 \ -2 \ -4] \Rightarrow [-1 \ 1 \ 1 \ -1 \ -1] = B$   |                      | $\varepsilon = -8$               |
| $C = [-1 \ -1 \ 1 \ -1 \ -1]$ | $\varepsilon = 0$  | $WC = [0 \ 4 \ 2 \ -2 \ 0] \Rightarrow [-1 \ 1 \ 1 \ -1 \ -1] = B$     |                      | $\varepsilon = -8$ 연상            |
| $D = [1 \ 1 \ 1 \ -1 \ 1]$    | $\varepsilon = 0$  | $WD = [0 \ -4 \ 2 \ -2 \ 0] \Rightarrow [-1 \ -1 \ 1 \ -1 \ -1] = E$   |                      | $\varepsilon = 2$ 다시 리콜          |
|                               |                    | $WE = [0 \ 4 \ 2 \ -2 \ 0] \Rightarrow [-1 \ 1 \ 1 \ -1 \ -1] = B$     |                      | $\varepsilon = -8$ 국소해, 최적해는 A   |
| $F = [-1 \ 1 \ -1 \ -1 \ 1]$  | $\varepsilon = 4$  | $WF = [0 \ 0 \ 2 \ 2 \ -4] \Rightarrow [-1 \ -1 \ 1 \ 1 \ -1] = G$     |                      | $\varepsilon = 4$ 다시 리콜          |
|                               |                    | $WG = [0 \ 4 \ -2 \ -2 \ 0] \Rightarrow [-1 \ 1 \ -1 \ -1 \ -1] = H$   |                      | $\varepsilon = -4$ 다시 리콜         |
|                               |                    | $WH = [-4 \ 4 \ 2 \ 2 \ -4] \Rightarrow [-1 \ 1 \ 1 \ 1 \ -1] = I$     |                      | $\varepsilon = -4$ 다시 리콜         |
|                               |                    | $WI = [-4 \ 4 \ -2 \ -2 \ -4] \Rightarrow [-1 \ 1 \ -1 \ -1 \ -1] = H$ |                      | $\varepsilon = -4$ 극솟값 정지로 리콜 실패 |

그림 2-8 홉필드 네트워크의 가중치 배열과 리콜

## 9.2.6 홉필드 네트워크에 의한 문자 인식

---

- 체험해 봅시다: Ex2\_홉필드 네트워크 문자 인식.xlsx  
-> 재수행을 통해 처리 원리를 이해

## 2.4 기타 신경망

---

### 2.4.1 볼츠만 머신

- **볼츠만 머신(Boltzmann Machine)**
  - 힌튼과 세즈노프스키(Geoffrey Hinton & Terry Sejnowski, 1985)에 의해 고안된 상호결합형 네트워크.
- 홉필드 네트워크의 단점을 다음과 같은 기법으로 개선하고 있음.  
(단, 리콜 연산에 확률 처리를 포함하기 때문에 시간이 더 걸린다는 단점도 있음)
  - ① 노드 간의 결합을 완전 결합으로 하지 않고, 적당히 제거하여 노드 수를 늘림.
  - ② 어닐링(annealing)에 의해 리콜 연산이 극솟값으로 떨어지는 것을 피하고 최적해를 얻음.

## 2.4.2 오차역전파 네트워크

---

- 오차역전파(Back Propagation) 기법은 룬멜하트(1986)에 의해 고안된 계층형 네트워크로, 퍼셉트론의 단점을 다음과 같은 기법으로 개선하고 있음.
  - ✓ 연합기층을 다단계층으로 개선함으로써 선형 분리가 불가능한 경우에도 가중치를 잘 조정할 수 있음.
  - ✓ 학습은 교사 신호와의 차이를 출력 측부터 역순으로 줄여 나감으로써 각 층의 가중치를 조정함.
- 오차역전파 네트워크는 1980년대에 산업계에서 가장 주목 받는 방법이었으나, 학습에 시간이 걸리고 입력을 인간이 주의해서 주지 않으면 안 되는 등 사용하기 어렵다는 단점이 있었음.

## 2.4.3 자기조직화 맵

---

- 자기조직화 맵(SOM: Self-Organizing Map)은 코호넨(Teuvo Kohonen, 1981)에 의해 고안된 2계층의 네트워크로, 입력 데이터를 유사도에 따라 분류하는, 클러스터링 수행을 목적으로 함.
- 학습 방법으로는 기존의 오차 정정이나 헵형이 아니고 경쟁 학습이라는 비지도 학습을 수행.
- 자기조직화 맵에는 다음과 같은 특징이 있음.
  - ✓ 입력층과 경쟁층이라는 2개 층으로 구성.
  - ✓ 입력층의 각 노드 값(입력)과의 차이가 최소인 경쟁층 노드를 발화시켜 주변 노드의 가중치를 갱신.
  - ✓ 네트워크의 전체 이득을 최대화하는 방향으로 동작하며 비지도 학습을 수행.

## 2.4.4 자기부호화기

---

- **자기부호화기(Autoencoder)** : 딥 러닝을 위한 **다층 신경망**
- **볼츠만 머신**을 고안한 제프리 힌튼(Geoffrey Hinton, 2006)에 의해 제안.
- 초기에 영상 인식에 있어서 **다른 기법을 압도적으로 능가하는 인식 정밀도**를 나타냄으로써 각광을 받았으며, 현재에는 학습의 본질에 가까운 기법으로 다방면에 응용이 확대되고 있음.
- 학습 방법은 각 층에서 입력과 동일한 것을 출력하도록 다음 층을 결정하는 **순전파 기법**
- 보통 입력 데이터가 너무 많아 모든 데이터에 대해 동일한 것을 출력하기 위한 계산에는 시간이 걸리므로, 이를 개선하기 위한 계산 기법도 연구되고 있음.

## 9장 신경망: 적용 사례

---

- [https://www.youtube.com/watch?v=C\\_VUtPNFqYw](https://www.youtube.com/watch?v=C_VUtPNFqYw)
- <https://www.youtube.com/watch?v=qv6UVOQ0F44>
- <https://www.youtube.com/watch?v=BBLJFYr7zB8>