Chapter3 텐서플로2.0 시작하기
- 딥러닝 기초 -

SeoKyeong University
Department of Computer Engineering
Professor. Lee Kwang Yeob

ESL
Embedded System Laboratory
Since 1995

Notice

- TensoFlow2.0 설치 결과 보고
- 교재 2장 설명에 따라 CUDA, 아나콘다 환경 설치(GPU 사용)
- colab 활용
- 보고 방법 : 이메일(kylee@skuniv.ac.kr)로 설치 결과 또는 문제점

- 매주 수업계획서에서 부여되는 [과제] 는 자습 으로 하며 제출하지 않음
- [레포트]로 부여 되는 경우 화일로 제출함
- 교재 내용 이외 자체 제작된 슬라이드 는 강의자료실에 업로드 됨

머신러닝 개념

- 2018년 9월 1일 날씨를 예측한다면?
- 데이터(예측 모델을 만들기 위한 학습 데이터) 필요

ESL Embedded System Laboratory Since 1995

- 내년 월별 평균 기온 데이터를 예측한다면?
- **데이터의 모델화 :** 데이터 수치를 있는 그대로가 아닌 원리를 생각하는 것

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

x : 각각의 달 (1, 2, ... , 12)

y : 예상 평균 기온 (ex 20℃)

w : 변경 가능한 파라미터

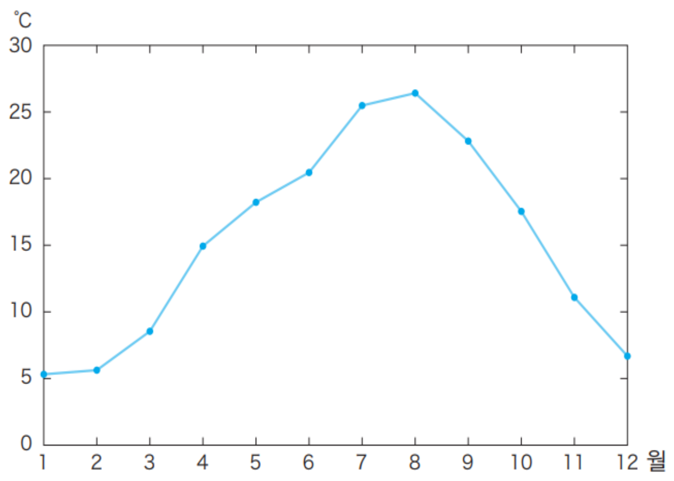


그림 1-3 월별 평균 기온 데이터

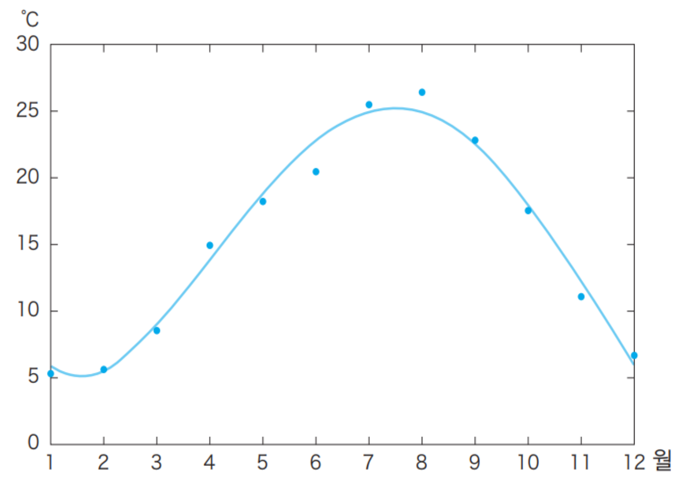
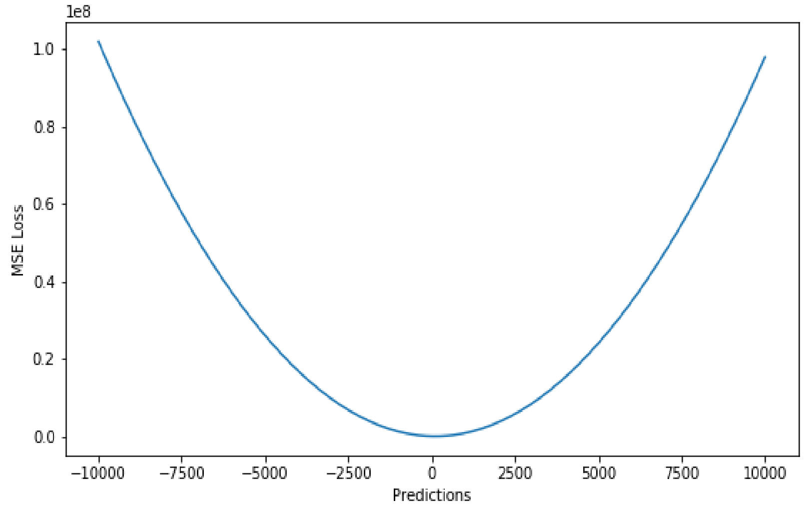


그림 1-4 완만한 곡선으로 예측한 평균 기온

- 곡선이 올바름의 기준 : **데이터 오차(Error)** 로 판단
- **오차 함수 (error function)**

  : W로 구성된 함수가 실제 값과 일치하지 않는 정도를 나타내는 함수
- **제곱 오차**



$$E = \frac{1}{2}\sum_{n=1}^{12}(y_n - t_n)^2$$

y : 예상 평균 기온 (ex 20℃)

t_n : n월 평균 기온 데이터

ESL Embedded System Laboratory Since 1995

- **머신러닝 모델의 3단계**
  - 주어진 데이터 기반으로 미지의 데이터 예측

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

  - 파라미터의 좋고 나쁨을 판단(오차 함수)

$$E = \frac{1}{2}\sum_{n=1}^{12}(y_n - t_n)^2$$

  - 오차 함수를 최소화하도록 파라미터 수정

텐서플로를 이용한 파라미터 최적화

- y_n은 n월 $(n = 1 \sim 12)$의 기온을 예측한 결과

$$y_n = w_0 + w_1 n + w_2 n^2 + w_3 n^3 + w_4 n^4 = \sum_{m=0}^{4} w_m n^m$$

- 예측한 식의 오차함수

$$E(w_0, w_1, w_2, w_3, w_4) = \frac{1}{2} \sum_{n=1}^{12} \left(\sum_{m=0}^{4} w_m n^m - t_n \right)^2$$

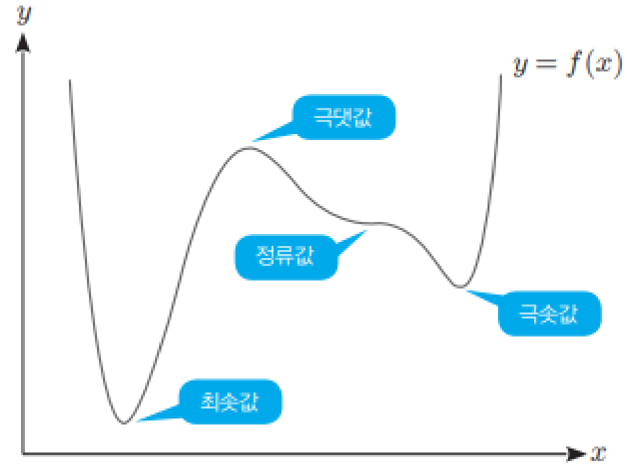


그림 1-13 그래프의 기울기가 0이 되는 지점

- 오차함수가 최소가 된 것이 데이터와 차이가 가장 적음

텐서플로를 이용한 파라미터 최적화

- **오차함수가 최소가 되도록 파라미터를 수정하기 위해서는 편미분이 필요**
  - **편미분 :** 복수의 변수를 갖는 함수에 대해 특정한 하나의 변수로 미분하는 것

- **2변수 함수의 편미분**
  - **임의의 함수** $h(x_1, x_2) = \frac{1}{4}(x_1^2 + x_2^2)$
  - **함수의 편미분**

$$\frac{\partial h}{\partial x_1}(x_1, x_2) = \frac{1}{2}x_1, \quad \frac{\partial h}{\partial x_2}(x_1, x_2) = \frac{1}{2}x_2$$

$$\nabla h(x_1, x_2) = \frac{1}{2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

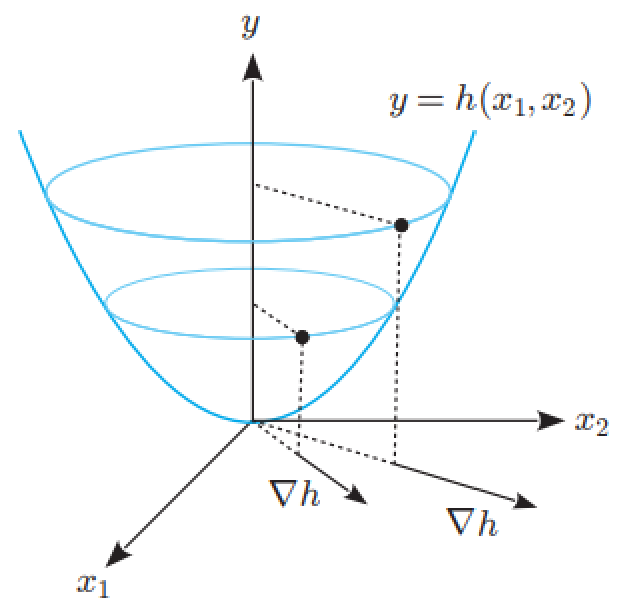
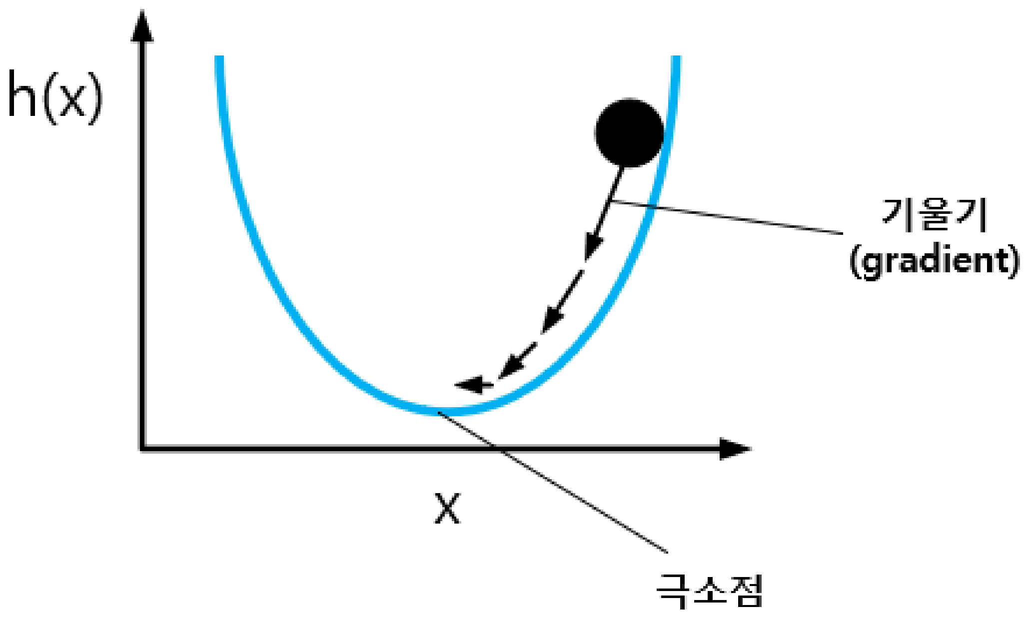


그림 1-14 2변수 함수의 기울기 벡터

텐서플로를 이용한 파라미터 최적화

- **경사 하강법(gradient descent)**
  - 오차함수가 최소가 되게 하기 위한 방법
  - 현재 위치에서 기울기와 반대방향으로 이동
  - 기울기 값을 계속 빼서 기울기가 0이되도록 한다.
  - 값 갱신 $x^{new} = x - \nabla h$



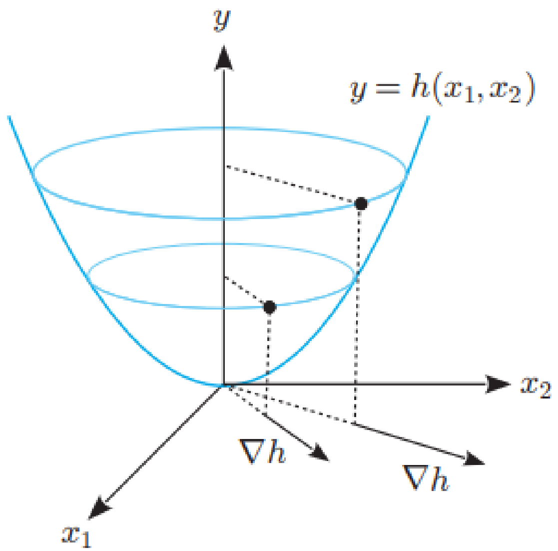


그림 1-14 2변수 함수의 기울기 벡터

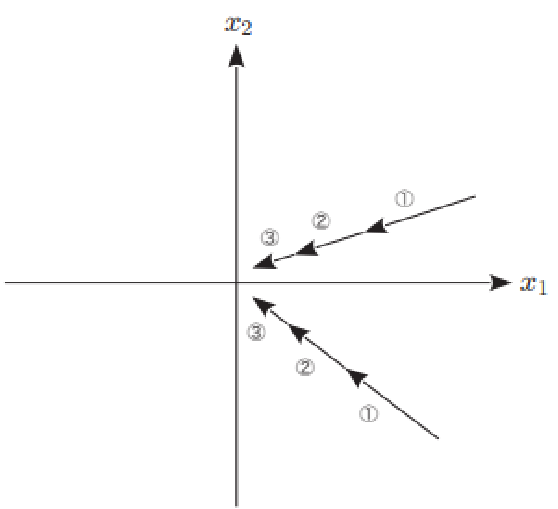
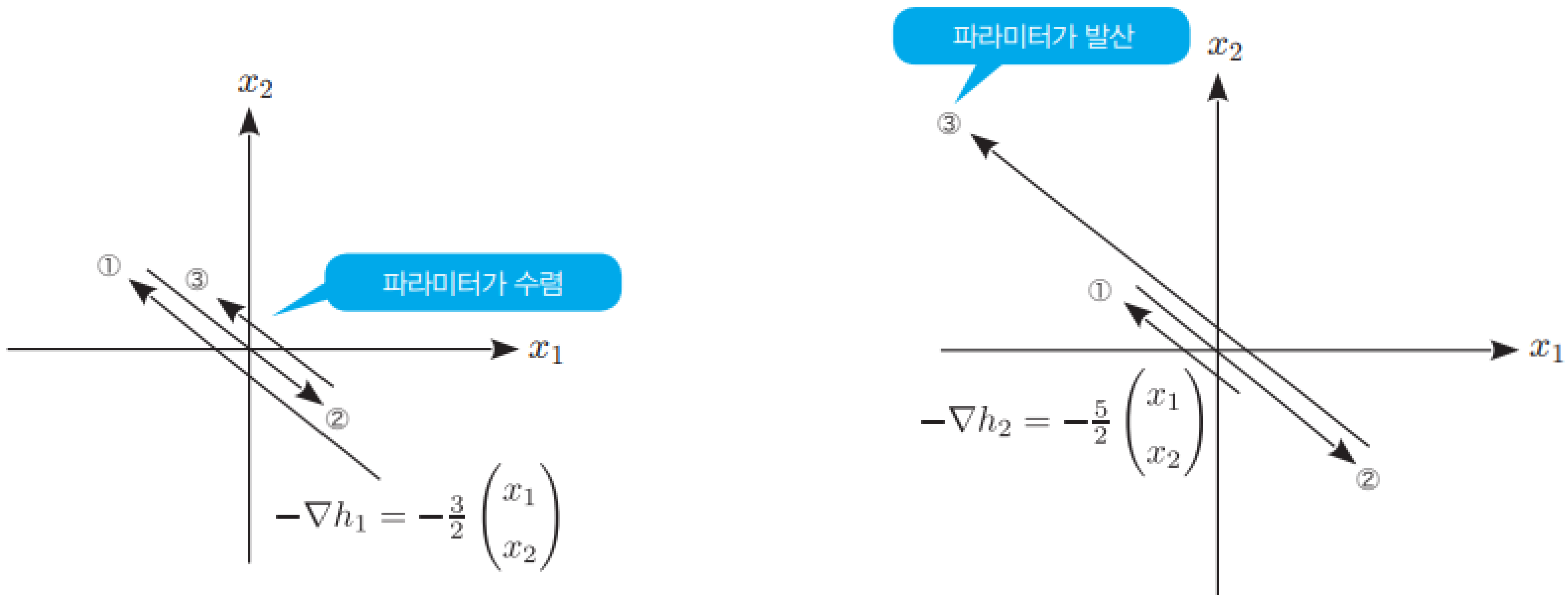


그림 1-15 경사 하강법으로 최솟값에 가까워지는 모습

텐서플로를 이용한 파라미터 최적화

- **학습률(learning rate)**
  - 경우에 따라 최솟값을 지나쳐 발산하는 경우가 발생



$$h_1(x_1, x_2) = \frac{3}{4}(x_1^2 + x_2^2)$$
$$\nabla h_2 = \frac{3}{2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$h_1(x_1, x_2) = \frac{5}{4}(x_1^2 + x_2^2)$$
$$\nabla h_2 = \frac{5}{2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

그림 1-16 경사 하강법에 의한 두 종류의 이동 예

텐서플로를 이용한 파라미터 최적화

- **학습률(learning rate)**
  - 발산을 방지하기 위해 이동량을 적당히 줄이는 파라미터
  - 클 수록 발산할 확률이 높아진다.
  - 작을수록 오차함수가 최소화되기까지 시간이 더 걸린다.

$$x^{new} = x - \epsilon \nabla h$$

- **트레이닝 세트(Training Set)**
  - 파라미터 w들을 최적화하기 위해 사용하는 데이터 집합

신경망의 필요성

- 예비 검사 데이터로 감염을 예측하는 예
  - 직선을 통해 경계를 나타내고 감염일 확률을 구한다.

- 머신러닝 모델의 3단계 중 **1단계**
  - 경계 : $\boldsymbol{f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2 = 0}$
  - 감염 확률 : $\boldsymbol{P(x_1, x_2) = \sigma(f(x_1, x_2))}$

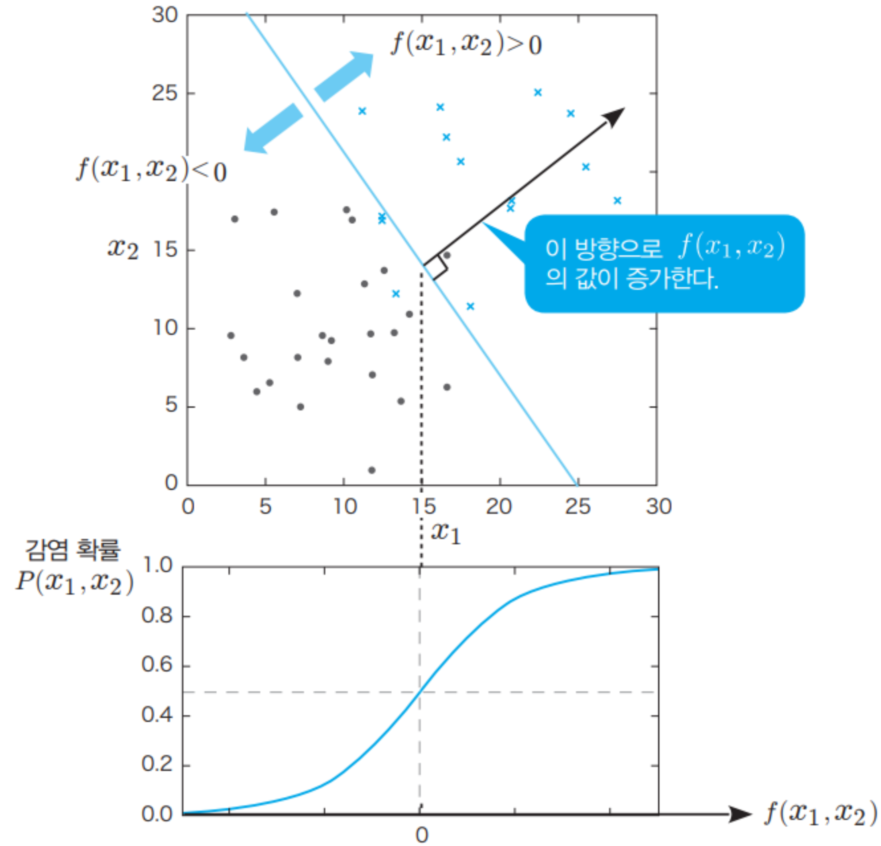


그림 1-6 직선을 이용한 분류와 감염 확률로의 변환

신경망의 필요성

- 직선의 방정식으로는 해결할 수 없는 문제
  - 데이터가 직선의 경계를 갖지 않을 때
  - 입력 데이터(x)가 여러 개일 때 (차원의 수 증가)

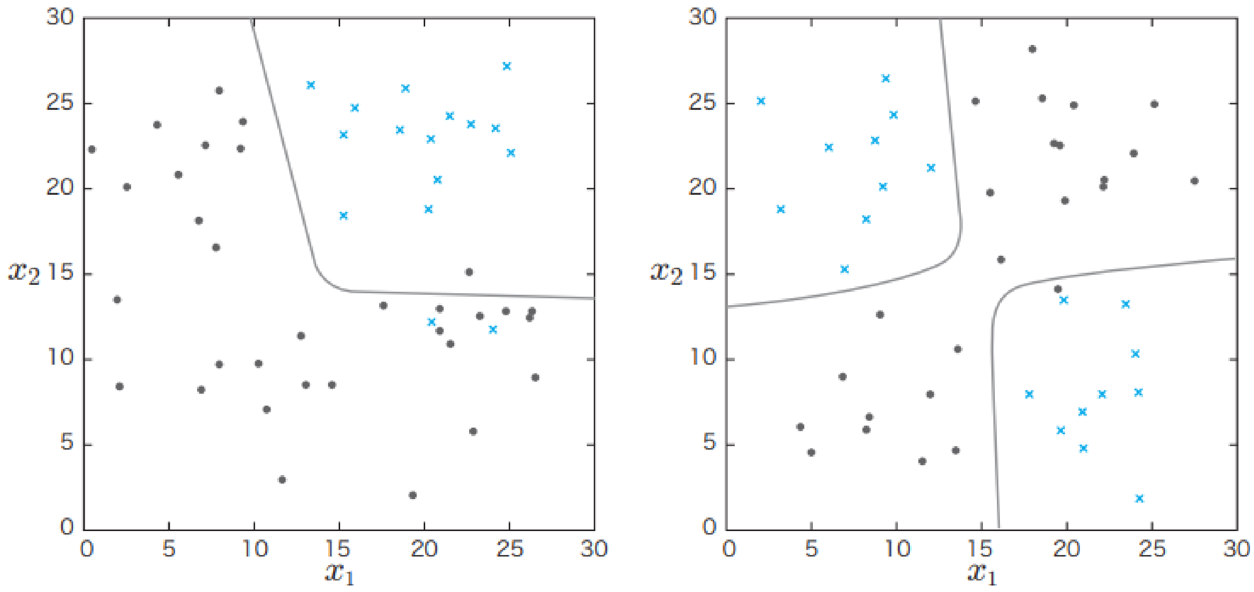


그림 1-7 보다 복잡한 데이터 배치의 예

신경망의 필요성

- **뉴런 (Neuron) 또는 노드 (Node)**
  - 신경망을 구성하는 최소 유닛

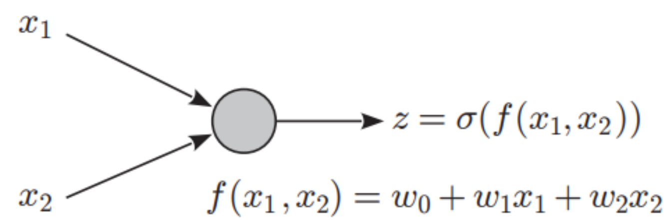
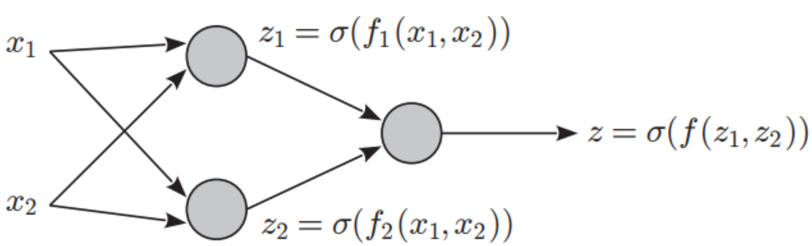


그림 1-8 단일 노드로 된 신경망

- **신경망 (Neural Network)**
  - 여러 뉴런을 다층으로 중첩한 것



$$f_1(x_1, x_2) = w_{10} + w_{11} x_1 + w_{12} x_2$$
$$f_2(x_1, x_2) = w_{20} + w_{21} x_1 + w_{22} x_2$$
$$f(z_1, z_2) = w_0 + w_1 z_1 + w_2 z_2$$

그림 1-9 2계층 노드로 된 신경망

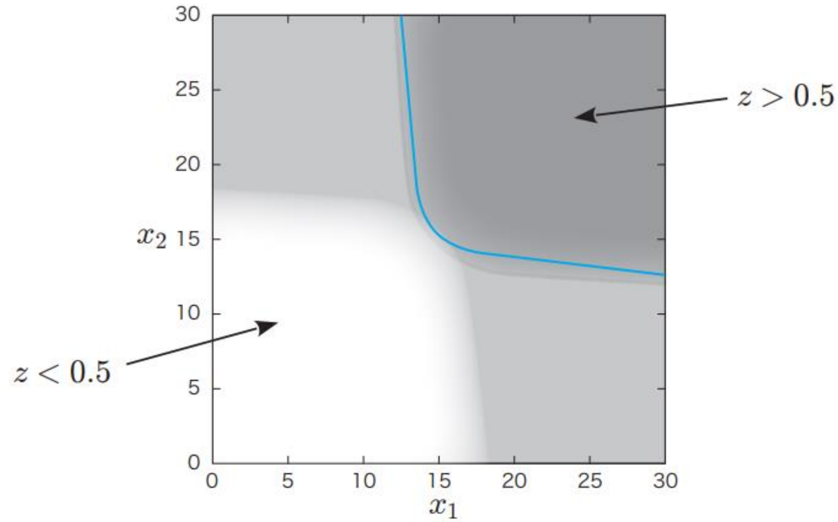


그림 1-10 2계층 신경망에 의한 분할 예

- **더욱 더 복잡한 신경망**
  - 복잡한 경계(함수)를 표현할 수 있다
  - 계산이 많이 필요해진다
  - 눈에 보이지 않는 데이터의 특성을 감으로 찾아야한다. => **딥러닝(deep learning) 등장**

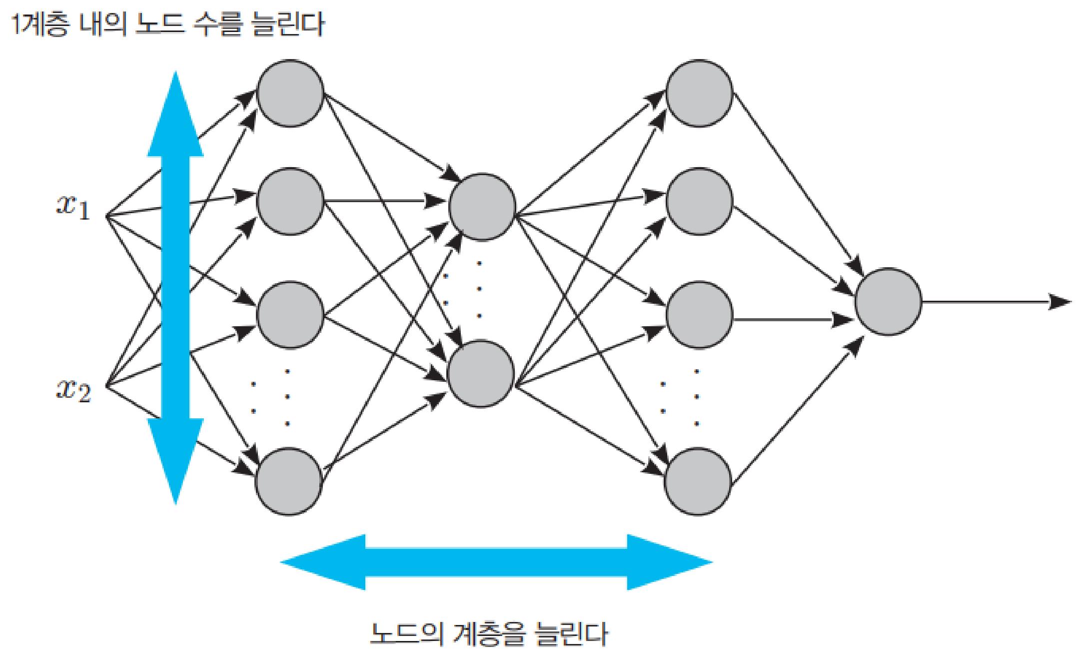


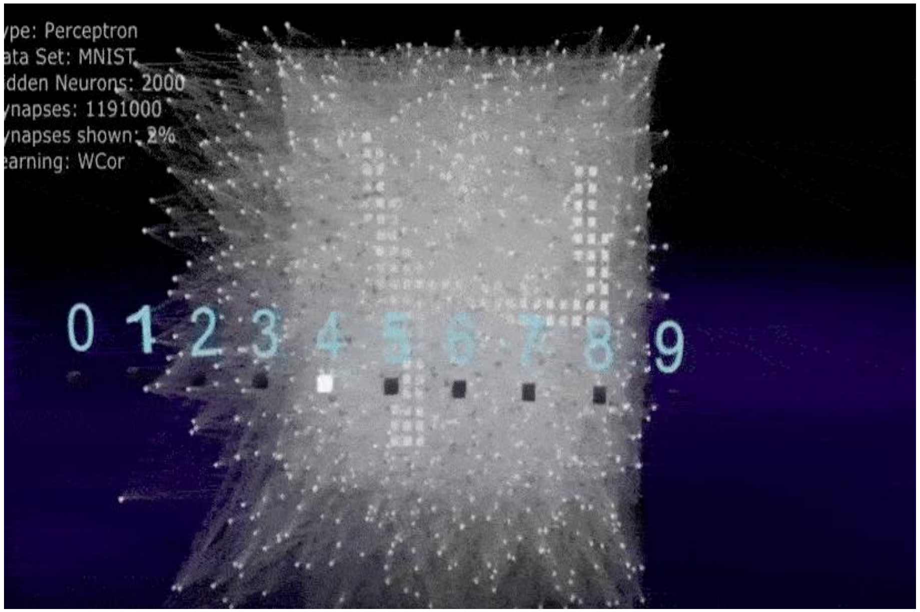
그림 1-11 보다 복잡한 다층 신경망의 예

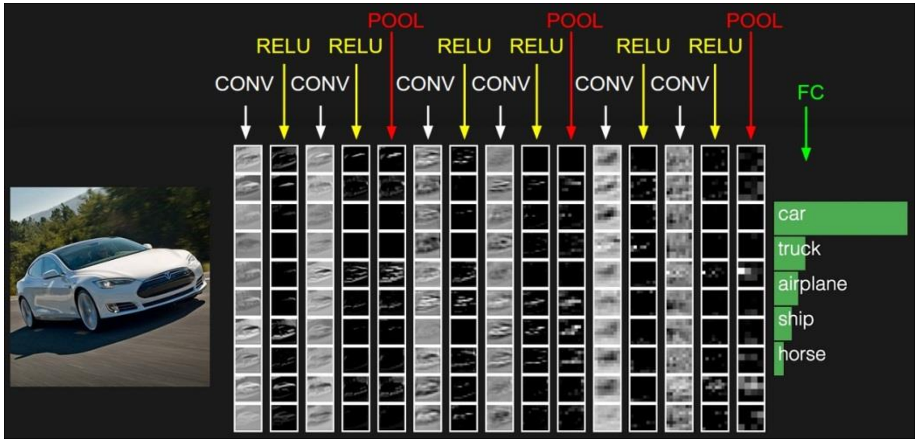
딥러닝의 특징

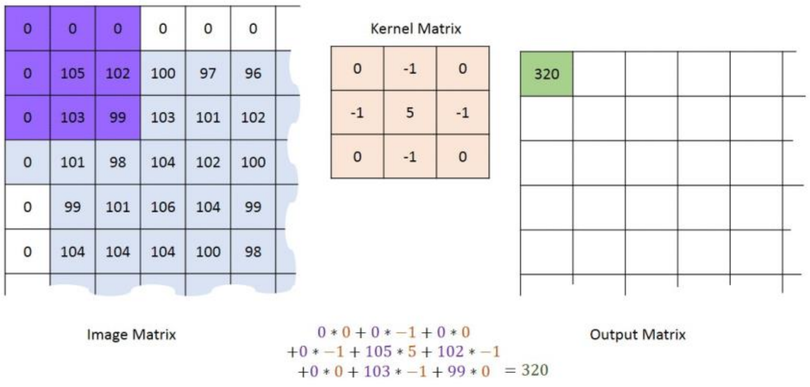
- **다층 신경망을 이용한 머신러닝**
  - 노드에 특별한 역할이나 노드 간의 연결방식을 다양하게 연구한 것

- **예시**
  - 합성곱 신경망(Convolutional Neural Network)
    - 합성곱 필터(Convolutional filter)
    - 풀링 계층(pooling layer)
  - 순환 신경망(RNN, Recurrent Neural Network)

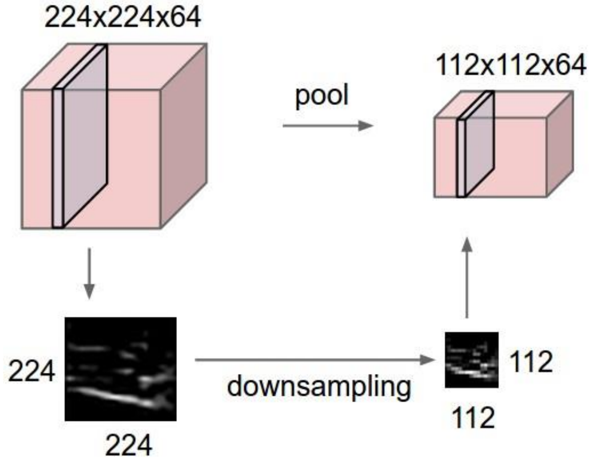
- **합성곱 신경망(Convolutional Neural Network)**







Convolution with horizontal and vertical strides = 1



ESL Embedded System Laboratory Since 1995

- **순환 신경망(Recurrent Neural Network)**

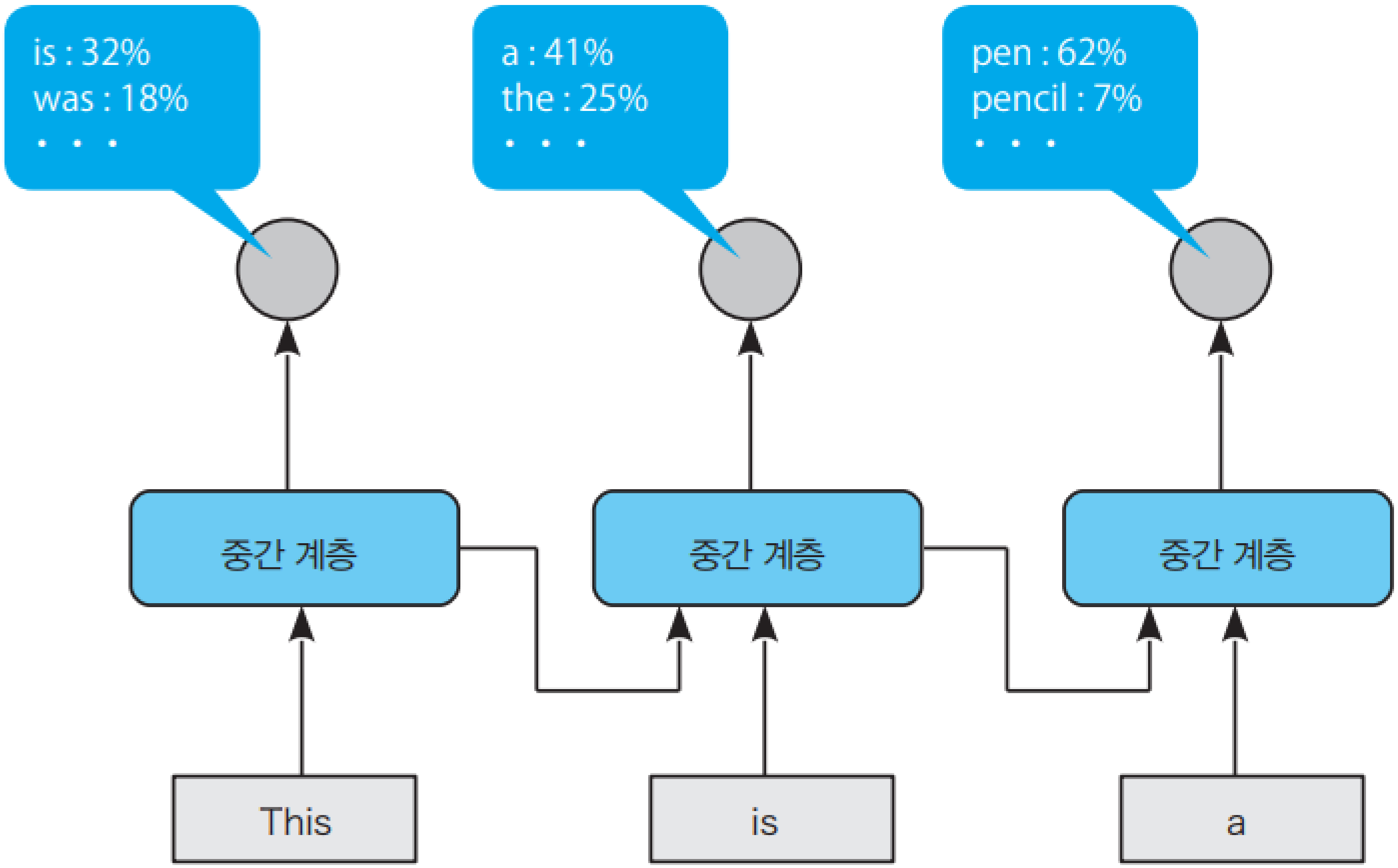


그림 1-12 RNN으로 다음 단어를 예측하는 예