손에 잡히는 딥러닝

# Score Function

모두의연구소

박은수 Research Director

# 진행할 내용들

- 기본 개념 및 기초 내용 (Deep NN)
- Convolutional NN
- Recurrent NN

# 돌아보기

- 고양이 분류 룰을 만들기 어려움 ..





우리는 Data Driven 방법을 취할 겁니다

```
def train(train_images, train_labels):
  # build a model for images -> labels...
  return model

def predict(model, test_images):
  # predict test_labels using the model...
  return test_labels
```

Example training set

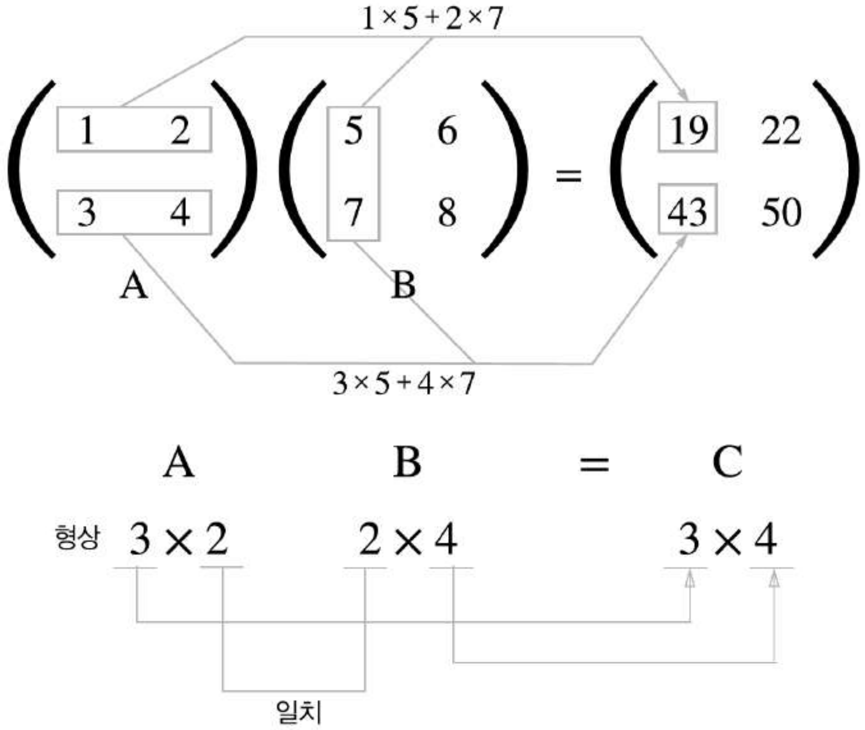
# Parametric Approach

- 분류기의 구성
  - Score function
  - Loss function
  - Optimization

# 들어가기 전에 ....

# 다차원 배열의 계산

- **다차원 배**



# Matrix 연산 리뷰 : Matrix-vector

## Example

$$\underbrace{\begin{bmatrix} 1 & 3 \\ 4 & 0 \\ 2 & 1 \end{bmatrix}}_{3\times 2} \underbrace{\begin{bmatrix} 1 \\ 5 \end{bmatrix}}_{2\times 1} = \underbrace{\begin{bmatrix} 16 \\ 4 \\ 7 \end{bmatrix}}_{3\times 1}$$

matrix

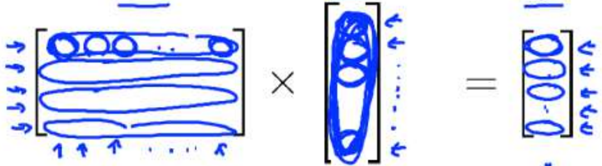
$1\times 1 + 3\times 5 = 16$

$4\times 1 + 0\times 5 = 4$

$2\times 1 + 1\times 5 = 7$

**Details:**

$$A \times x = y$$



$\boxed{m}\times\boxed{n}$ matrix (m rows, n columns) — $\boxed{n}\times 1$ matrix (n-dimensional vector) — $\boxed{m}$-dimensional vector

To get $y_i$, multiply $A$'s $i^{th}$ row with elements of vector $x$, and add them up.

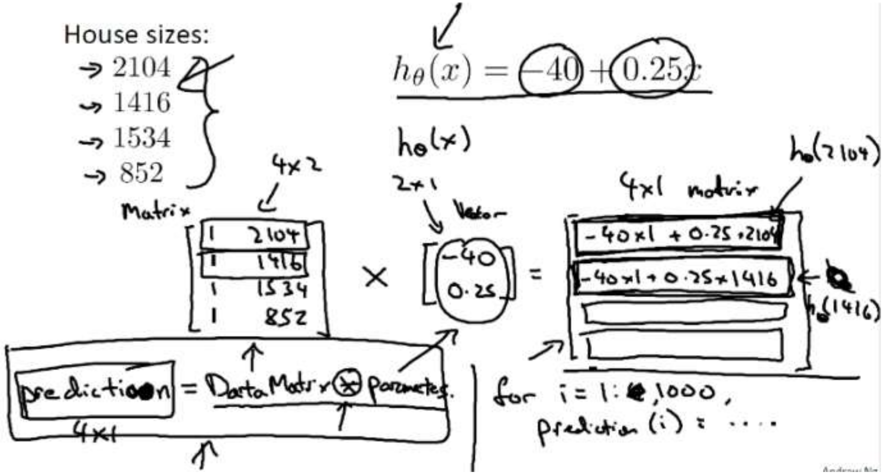
# Matrix 연산 리뷰 : Matrix-vector

모두의연구소

## Example

$$\underset{\boxed{3}\times 4}{\begin{bmatrix}1&2&1&5\\0&3&0&4\\-1&-2&0&0\end{bmatrix}}\underset{4\times 1}{\begin{bmatrix}1\\3\\2\\1\end{bmatrix}} = \underset{3\times 1}{\begin{bmatrix}\boxed{14}\\\boxed{13}\\\boxed{-7}\end{bmatrix}} = \begin{bmatrix}14\\13\\-7\end{bmatrix}$$

$$\begin{bmatrix}1\times 1 + 2\times 3 + 1\times 2 + 5\times 1 = 14\\ 0\times 1 + 3\times 3 + 0\times 2 + 4\times 1 = 13\\ -1\times 1 + (-2)\times 3 + 0\times 2 + 0\times 1 = -7\end{bmatrix}$$



# Matrix 연산 리뷰 : Matrix-matrix

모두의연구소

**Example**

$$\begin{bmatrix}1 & 3 & 2\\4 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 3\\0 & 1\\5 & 2\end{bmatrix} = \begin{bmatrix}11 & 10\\9 & 14\end{bmatrix}$$

2×3 3×2 2×2

$$\begin{bmatrix}1 & 3 & 2\\4 & 0 & 1\end{bmatrix}\times\begin{bmatrix}1\\0\\5\end{bmatrix} = \begin{bmatrix}11\\9\end{bmatrix}$$

$$\begin{bmatrix}1 & 3 & 2\\4 & 0 & 1\end{bmatrix}\times\begin{bmatrix}3\\1\\2\end{bmatrix} = \begin{bmatrix}10\\14\end{bmatrix}$$

**Details:**

$A \times B = C$

m x n matrix (m rows, n columns) × n x o matrix (n rows, o columns) = m x o matrix

n×1

The $i^{th}$ column of the matrix $C$ is obtained by multiplying $A$ with the $i^{th}$ column of $B$. (for $i$ = 1,2,...,o)

# Matrix 연산 리뷰 : Matrix-matrix

모두의연구소

House sizes:

$$\begin{Bmatrix} 2104 \\ 1416 \\ 1534 \\ 852 \end{Bmatrix}$$

Have 3 competing hypotheses:

1. $h_\theta(x) = -40 + 0.25x$
2. $h_\theta(x) = 200 + 0.1x$
3. $h_\theta(x) = -150 + 0.4x$

Matrix

Matrix

$$\begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix} \times \begin{bmatrix} \boxed{\begin{matrix} -40 \\ 0.25 \end{matrix}} & \boxed{\begin{matrix} 200 \\ 0.1 \end{matrix}} & \boxed{\begin{matrix} -150 \\ 0.4 \end{matrix}} \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} 486 \\ 314 \\ 344 \\ 173 \end{matrix}} & \boxed{\begin{matrix} 410 \\ 342 \\ 353 \\ 285 \end{matrix}} & \boxed{\begin{matrix} 692 \\ 416 \\ 464 \\ 191 \end{matrix}} \end{bmatrix}$$

Prediction of first $h_\theta$

Predictions of $2^{nd}$ $h_\theta$

From : Andrew Ng, "Machine Learning : Linear Algebra review", Lecture note; Coursera

# 선형대수 리뷰

http://cs229.stanford.edu/section/cs229-linalg.pdf

**좀 더 깊은 리뷰를 보시려면 참고하세요**

### 2.1 Vector-Vector Products

Given two vectors $x, y \in \mathbb{R}^n$, the quantity $x^T y$, sometimes called the ***inner product*** or ***dot product*** of the vectors, is a real number given by

$$x^T y \in \mathbb{R} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i.$$

Observe that inner products are really just special case of matrix multiplication. Note that it is always the case that $x^T y = y^T x$.

Given vectors $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$ (not necessarily of the same size), $xy^T \in \mathbb{R}^{m \times n}$ is called the ***outer product*** of the vectors. It is a matrix whose entries are given by $(xy^T)_{ij} = x_i y_j$, i.e.,

$$xy^T \in \mathbb{R}^{m \times n} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_m y_1 & x_m y_2 & \cdots & x_m y_n \end{bmatrix}.$$

As an example of how the outer product can be useful, let $\mathbf{1} \in \mathbb{R}^n$ denote an $n$-dimensional vector whose entries are all equal to 1. Furthermore, consider the matrix $A \in \mathbb{R}^{m \times n}$ whose columns are all equal to some vector $x \in \mathbb{R}^m$. Using outer products, we can represent $A$ compactly as,

$$A = \begin{bmatrix} | & | & & | \\ x & x & \cdots & x \\ | & | & & | \end{bmatrix} = \begin{bmatrix} x_1 & x_1 & \cdots & x_1 \\ x_2 & x_2 & \cdots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_m & \cdots & x_m \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} = x\mathbf{1}^T.$$

### 2.2 Matrix-Vector Products

Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $x \in \mathbb{R}^n$, their product is a vector $y = Ax \in \mathbb{R}^m$. There are a couple ways of looking at matrix-vector multiplication, and we will look at each of them in turn.

If we write $A$ by rows, then we can express $Ax$ as,

$$y = Ax = \begin{bmatrix} — & a_1^T & — \\ — & a_2^T & — \\ & \vdots & \\ — & a_m^T & — \end{bmatrix} x = \begin{bmatrix} a_1^T x \\ a_2^T x \\ \vdots \\ a_m^T x \end{bmatrix}.$$

# 다시 돌아와서 ...

강아지와 고양이 분류하기

픽셀값들을 조합하여 단순히 각 레이블에 대한 점수를 부여하면 어떻까요? Score function?

# Parametric approach

모두의연구소

- Score function

 → f(  , W) →
- 강아지 점수
- 고양이 점수

 → f(  , W) →
- 강아지 점수
- 고양이 점수

⋮ ⋮ ⋮

점수가 높은것으로 분류

# Parametric approach

- Score function : Simple Linear Classifier

$$f(\mathbf{x}, W) = W\mathbf{x} \quad 3{,}072\text{x}1$$

32x32x3 영상 =
3,072 픽셀

| | x |
|---|---|
| 1 | 155 |
| 2 | 200 |
| … | … |
| 3071 | 110 |
| 3072 | 78 |

$f(\mathbf{x}, W)$

- 강아지 점수
- 고양이 점수

# Parametric approach

- Score function : Simple Linear Classifier

$$\underset{2\times1}{f(x, W)} = Wx \quad 3{,}072\times1$$

32x32x3 영상 =
3,072 픽셀

| | |
|---|---|
| 1 | 155 |
| 2 | 200 |
| … | … |
| 3071 | 110 |
| 3072 | 78 |

$x$

$f(x, W)$

- 강아지 점수
- 고양이 점수

| |
|---|
| 0.3 |
| 0.7 |

# Parametric approach

- Score function : Simple Linear Classifier

$$\underset{2\times1}{f}(x, W) = \underset{2\times3{,}072}{W}\,\underset{3{,}072\times1}{x}$$

32x32x3 영상 =
3,072 픽셀

| | x |
|---|---|
| 1 | 155 |
| 2 | 200 |
| … | … |
| 3071 | 110 |
| 3072 | 78 |

f(x, W)

| | | | | |
|---|---|---|---|---|
| 0.2 | 0.1 | … | 0.3 | 0.7 |
| 0.7 | 0.8 | … | 0.9 | 0.4 |

- 강아지 점수 — 0.3
- 고양이 점수 — 0.7

# Parametric approach

- Score function : Simple Linear Classifier

3,072x1

$$\underset{2\times1}{f(x, W)} = Wx + \underset{2\times1}{b}$$

2x3,072

32x32x3 영상 =
3,072 픽셀

| | x |
|---|---|
| 1 | 155 |
| 2 | 200 |
| … | … |
| 3071 | 110 |
| 3072 | 78 |

x

f(x, W)

| | | | | |
|---|---|---|---|---|
| 0.2 | 0.1 | … | 0.3 | 0.7 |
| 0.7 | 0.8 | … | 0.9 | 0.4 |

- 강아지 점수
- 고양이 점수

| |
|---|
| 0.3 |
| 0.7 |

# Parametric approach

- Score function : Simple Linear Classifier

$$\underset{2\times1}{f(x, W)} = \underset{2\times3{,}072}{W}\,\overset{3{,}072\times1}{x} + \underset{2\times1}{b}$$

32x32x3 영상 =
3,072 픽셀

| | | | | | |
|---|---|---|---|---|---|
| 강아지 파라미터 | 0.2 | 0.1 | ... | 0.3 | 0.7 |
| 고양이 파라미터 | 0.7 | 0.8 | ... | 0.9 | 0.4 |

W

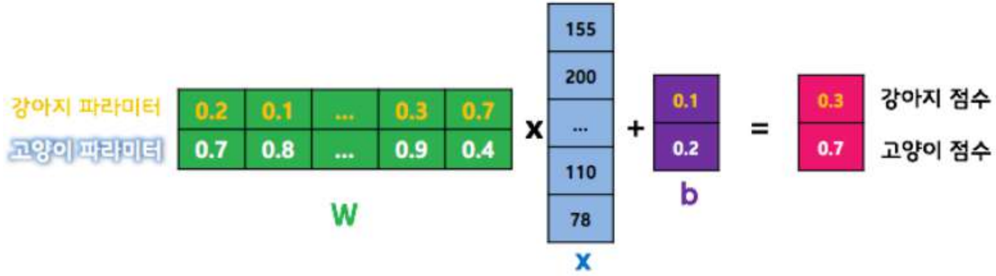
$$\begin{bmatrix}0.2 & 0.1 & \cdots & 0.3 & 0.7\\ 0.7 & 0.8 & \cdots & 0.9 & 0.4\end{bmatrix} \times \begin{bmatrix}155\\ 200\\ \cdots\\ 110\\ 78\end{bmatrix} + \begin{bmatrix}0.1\\ 0.2\end{bmatrix} = \begin{array}{cl}\begin{bmatrix}0.3\\ 0.7\end{bmatrix} & \begin{array}{l}\text{강아지 점수}\\ \text{고양이 점수}\end{array}\end{array}$$

x

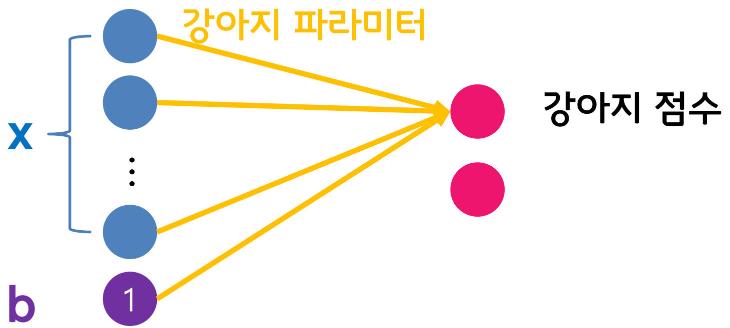
b

# Parametric approach

- Score function : Simple Linear Classifier

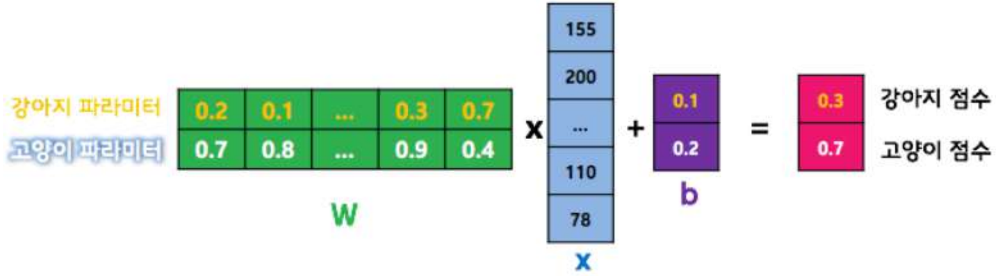



선형분류기의
뉴럴 네트워크 표현

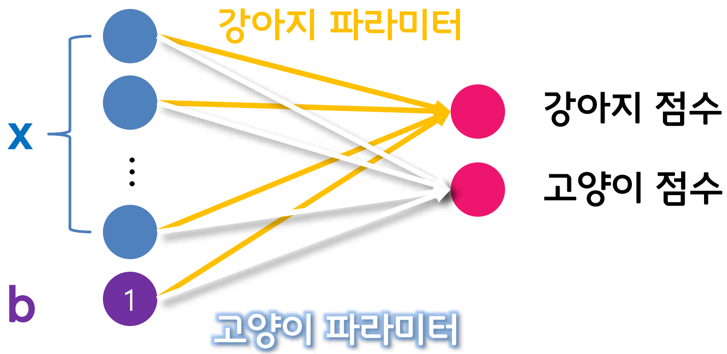


# Parametric approach

- Score function : Simple Linear Classifier

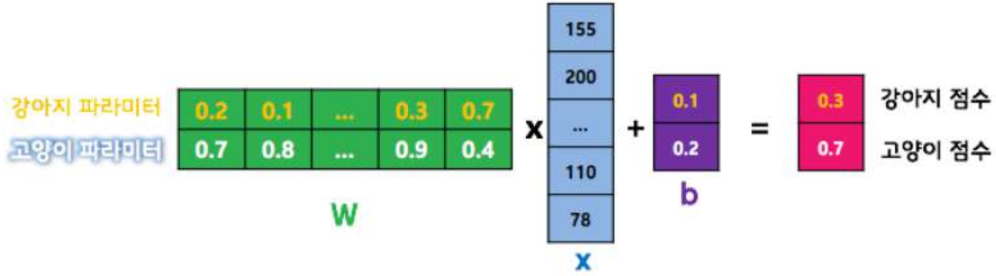



선형분류기의
뉴럴 네트워크 표현

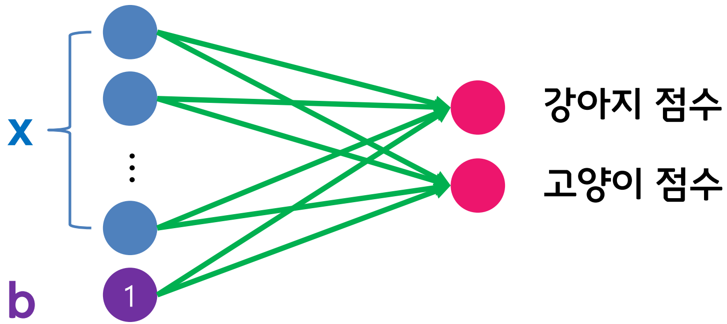


# Parametric approach

- Score function : Simple Linear Classifier




선형분류기의
뉴럴 네트워크 표현



# 이제 해야 할 것은

**Score**

| | | | |
|---|---|---|---|
| airplane | -3.45 | -0.51 | 3.42 |
| automobile | -8.87 | **6.04** | 4.64 |
| bird | 0.09 | 5.31 | 2.65 |
| cat | **2.9** | -4.22 | 5.1 |
| deer | 4.48 | -4.19 | 2.64 |
| dog | 8.02 | 3.58 | 5.55 |
| frog | 3.78 | 4.49 | **-4.34** |
| horse | 1.06 | -4.37 | -1.5 |
| ship | -0.36 | -2.09 | -4.79 |
| truck | -0.72 | -2.93 | 6.14 |

1. 계산된 Score가 얼마나 안좋은지 측정할 수 있는 함수를 만들어야 합니다

   ➡ Loss function

2. Loss function을 최소화 할수 있는 방법을 만들어야 합니다

   (Optimization)

Coming up:
- Loss function
- Optimization

$$f(x, W) = Wx$$

- Loss Function : 현재 내가 구성한 W가 좋은지 안좋은지 측정할 수 있게 해줍니다
- Optimization : random W로 시작해서 Loss를 최소로 갖게 W를 변경해 줍니다

모두의연구소


박 은 수  Research Director

E-mail : es.park@modulabs.co.kr