

2주차(1/3)

# 함수와 뉴론

파이썬으로 배우는 기계학습

한동대학교  
김영섭 교수

# 함수와 뉴론

---

- 학습 목표
  - 함수와 뉴론을 이해한다.
  - 인공뉴론과 인공신경망을 이해한다.
  - 첫 인공뉴론을 구현한다.
- 학습 내용
  - 함수와 뉴론
  - 인공뉴론과 인공신경망
  - 인공뉴론의 구현

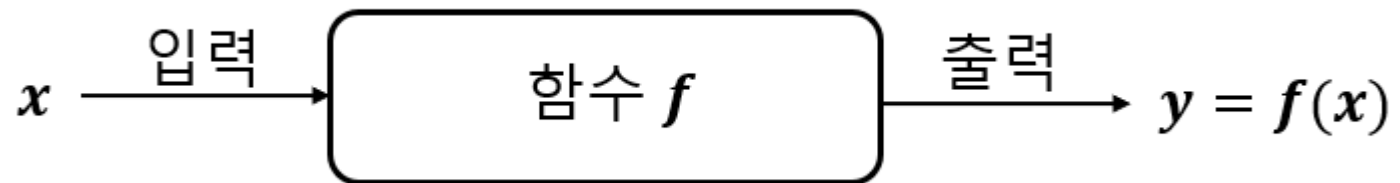
# 함수와 뉴론

---

- 함수
  - 함수  $f$

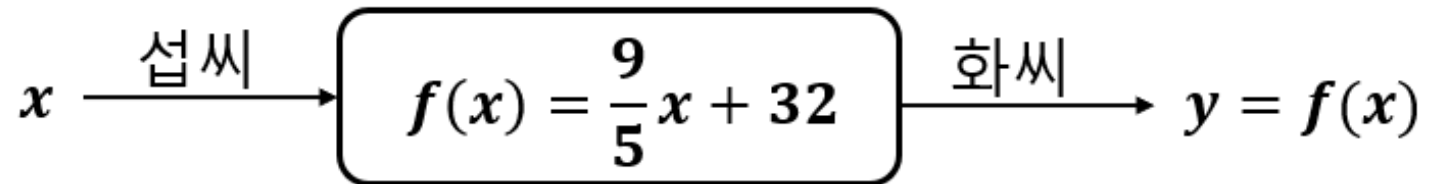
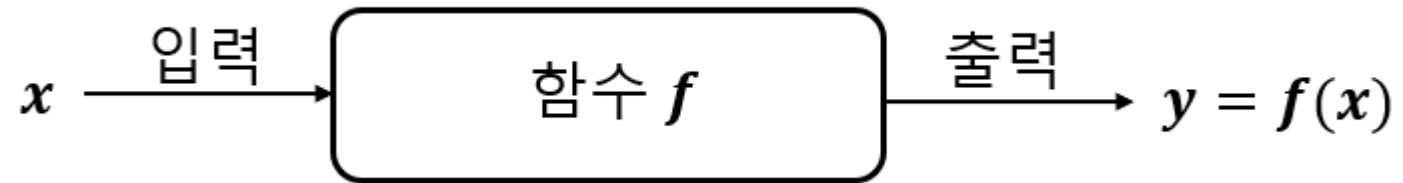
# 함수와 뉴론

- 함수
  - 함수  $f$
  - 온도변환 함수



# 함수와 뉴론

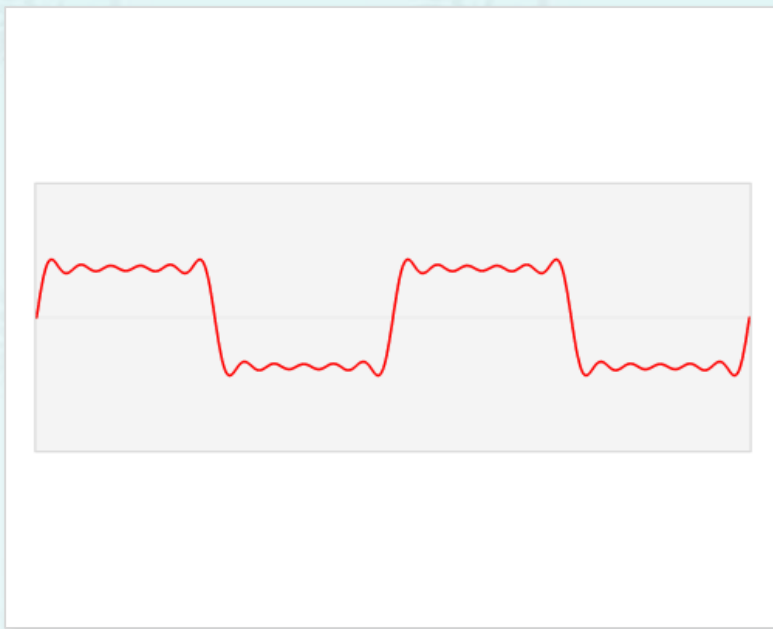
- 함수
  - 함수  $f$
  - 온도변환 함수



# 함수와 뉴론

- 함수
  - 푸리에 변환 함수

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2i\pi\omega x} dx$$

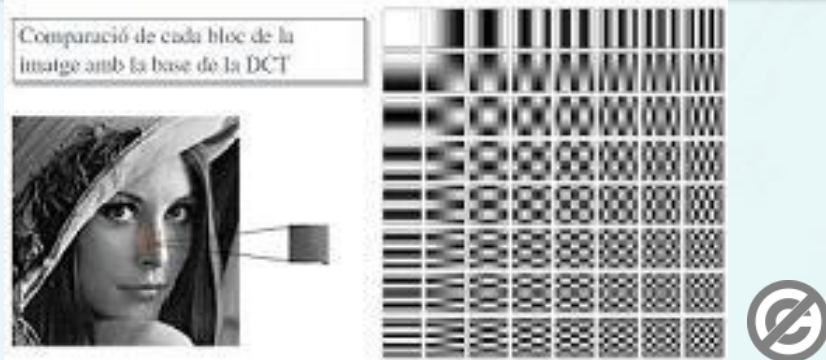


# 함수와 뉴론

- 함수

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2i\pi\omega x} dx$$

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(i, j) \cos \left[ \frac{\pi(2i+1)u}{2N} \right] \cos \left[ \frac{\pi(2j+1)v}{2M} \right]$$



# 함수와 뉴론

## ■ 함수



$$x \xrightarrow{\text{입력}} f(x) = \frac{9}{5}x + 32 \xrightarrow{\text{출력}} y = f(x)$$

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2i\pi\omega x} dx$$

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(i, j) \cos \left[ \frac{\pi(2i+1)u}{2N} \right] \cos \left[ \frac{\pi(2j+1)v}{2M} \right]$$

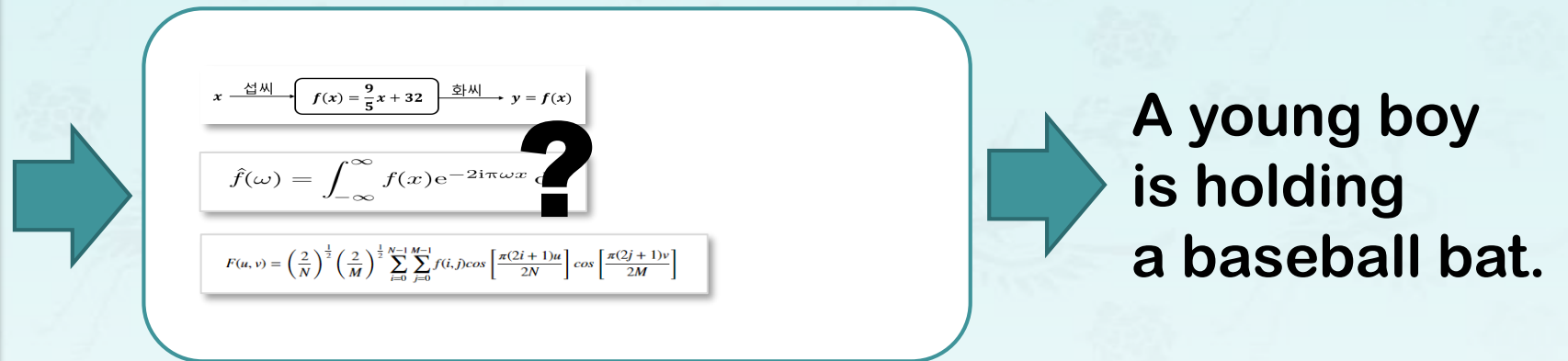
?





# 함수와 뉴론

- 함수



출처: Andrej Karpathy, Li Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", 2015

# 함수와 뉴론

- 함수



기계학습



**A young boy  
is holding  
a baseball bat.**

# 함수와 뉴론

- 함수



**Dream Function Generator**



기계학습: 만능 함수 제조기



**A young boy  
is holding  
a baseball bat.**

**Universal Function Generator**

# 뉴론과 신경망

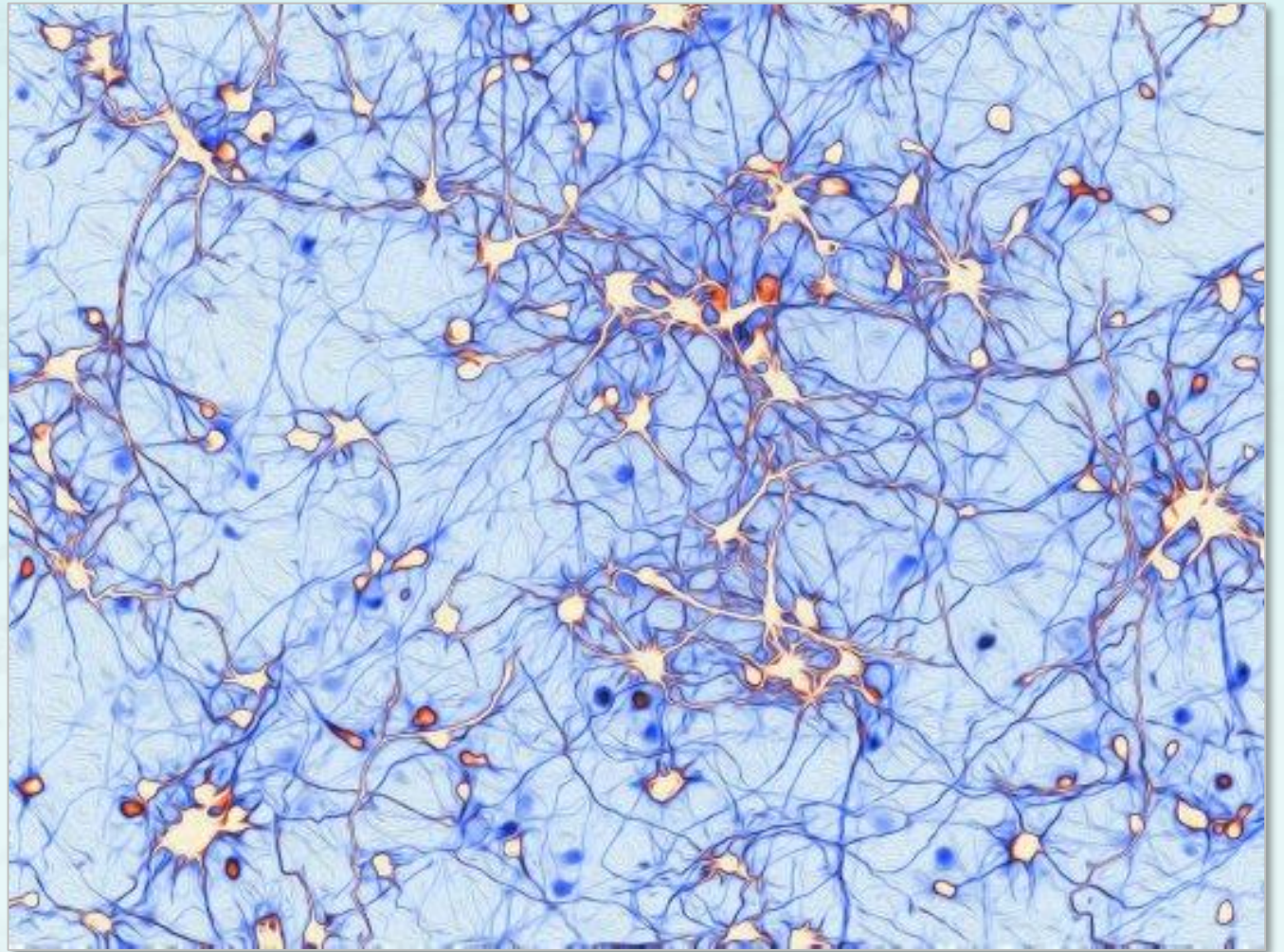
---

- 뉴론
  - 뉴론(뇌세포), 뇌의 기본 단위
  - **850억개**



# 뉴론과 신경망

- 뉴론
  - 뉴론(뇌세포), 뇌의 기본 단위
  - 850억개

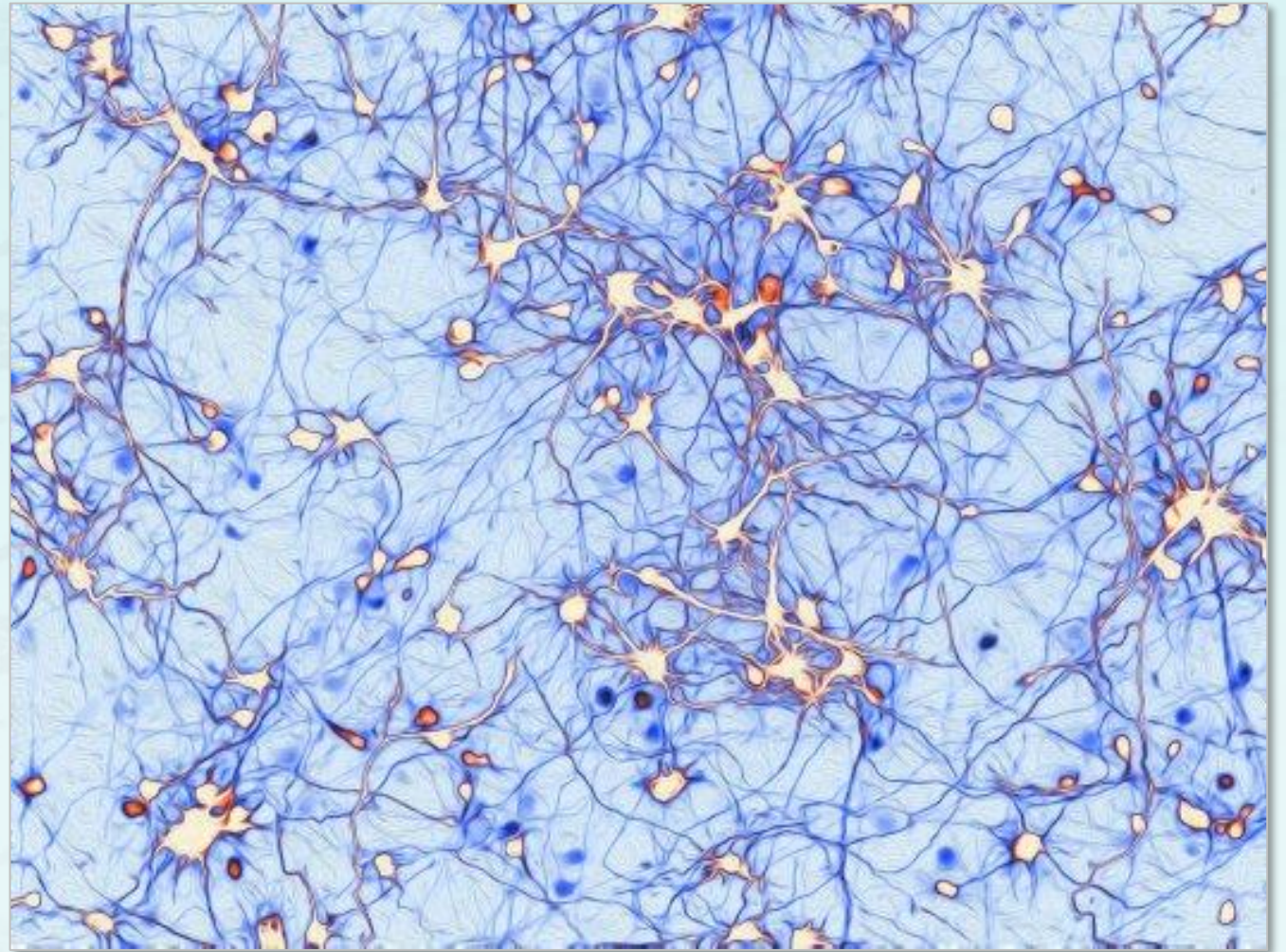


© Victor Anggono, 2015



# 뉴론과 신경망

- 뉴론
  - 뉴론(뇌세포), 뇌의 기본 단위
  - 850억개
- 뉴론 연산자
  - 임계값

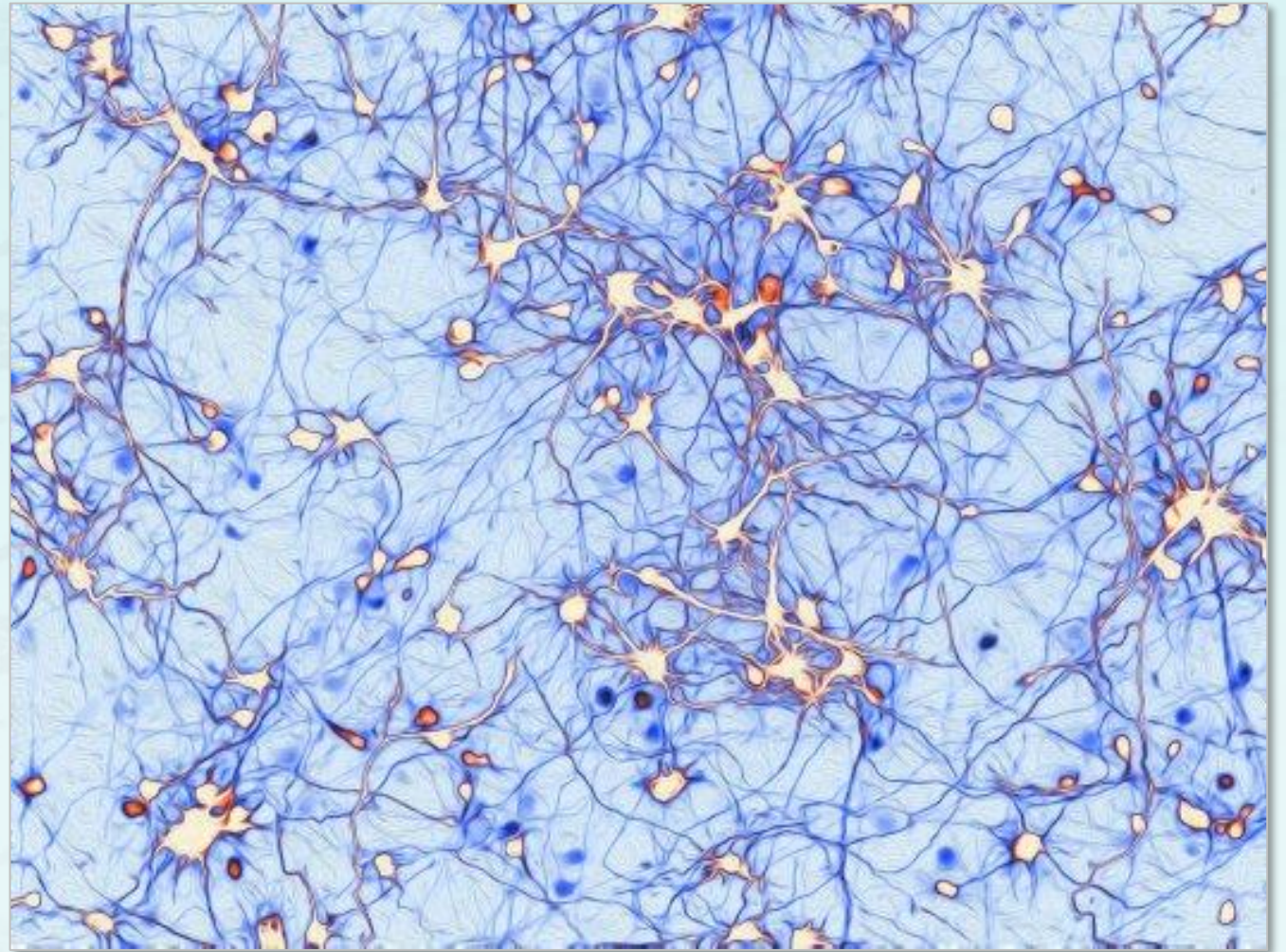


© Victor Anggono, 2015



# 뉴론과 신경망

- 뉴론
  - 뉴론(뇌세포), 뇌의 기본 단위
  - 850억개
- 뉴론 연산자
  - 임계값
  - 입력은 다수, 출력은 하나
  - 입력의 합산

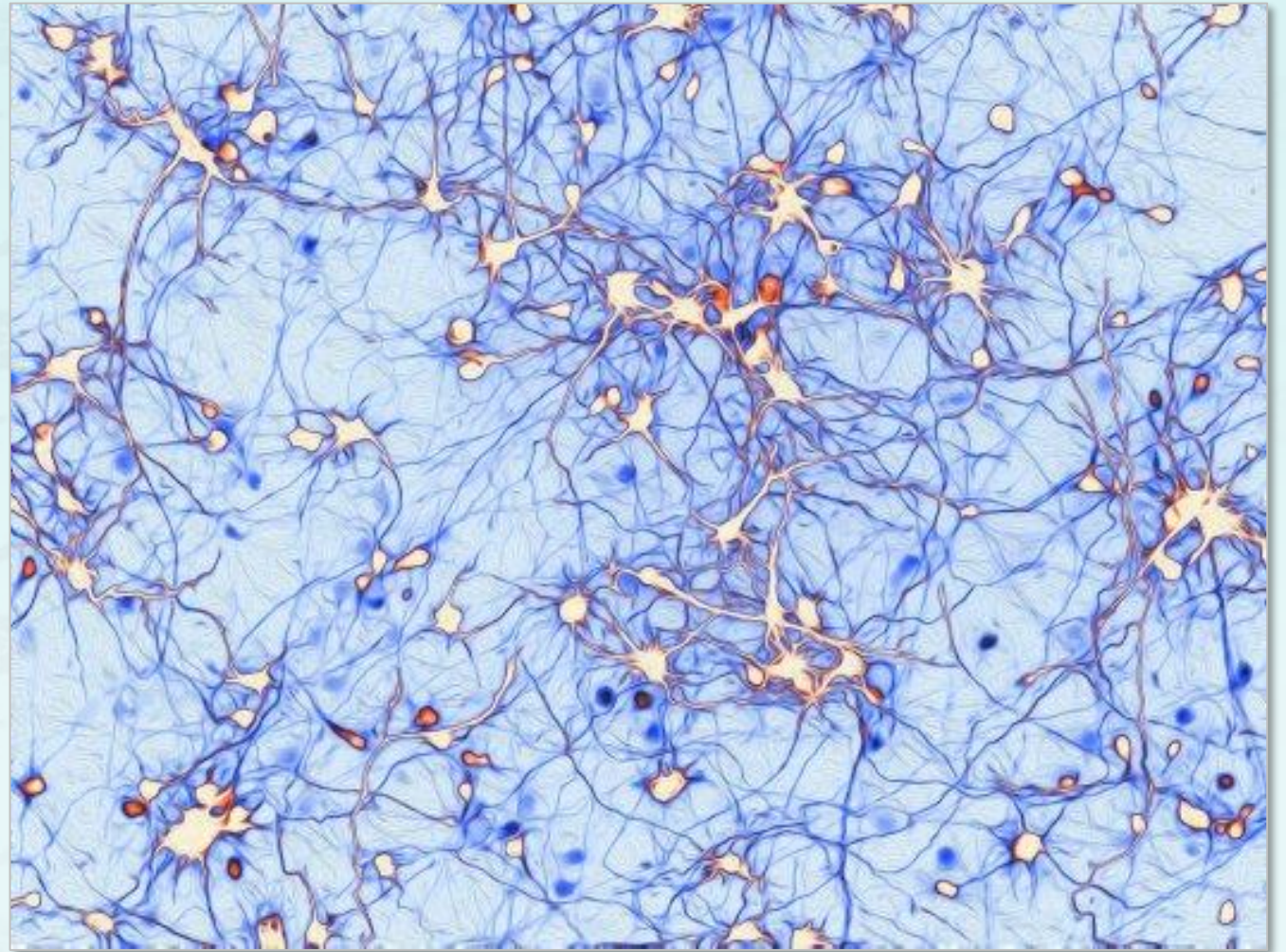


© Victor Anggono, 2015



# 뉴론과 신경망

- 뉴론
  - 뉴론(뇌세포), 뇌의 기본 단위
  - 850억개
- 뉴론 연산자
  - 임계값
  - 입력은 다수, 출력은 하나
  - 입력의 합산
  - 신경망 구성

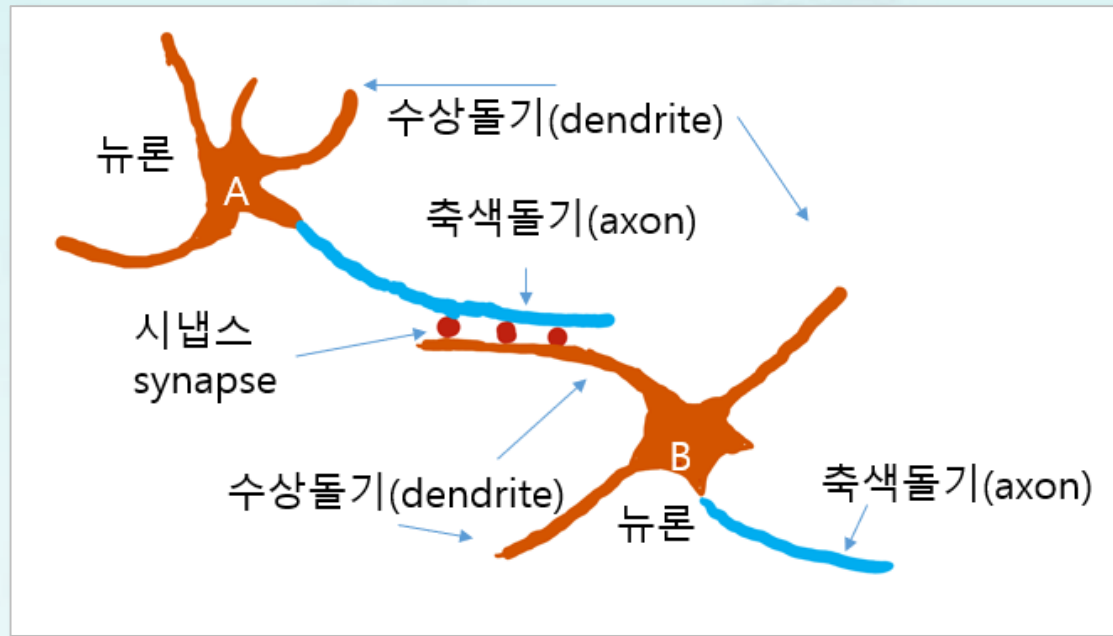


© Victor Anggono, 2015



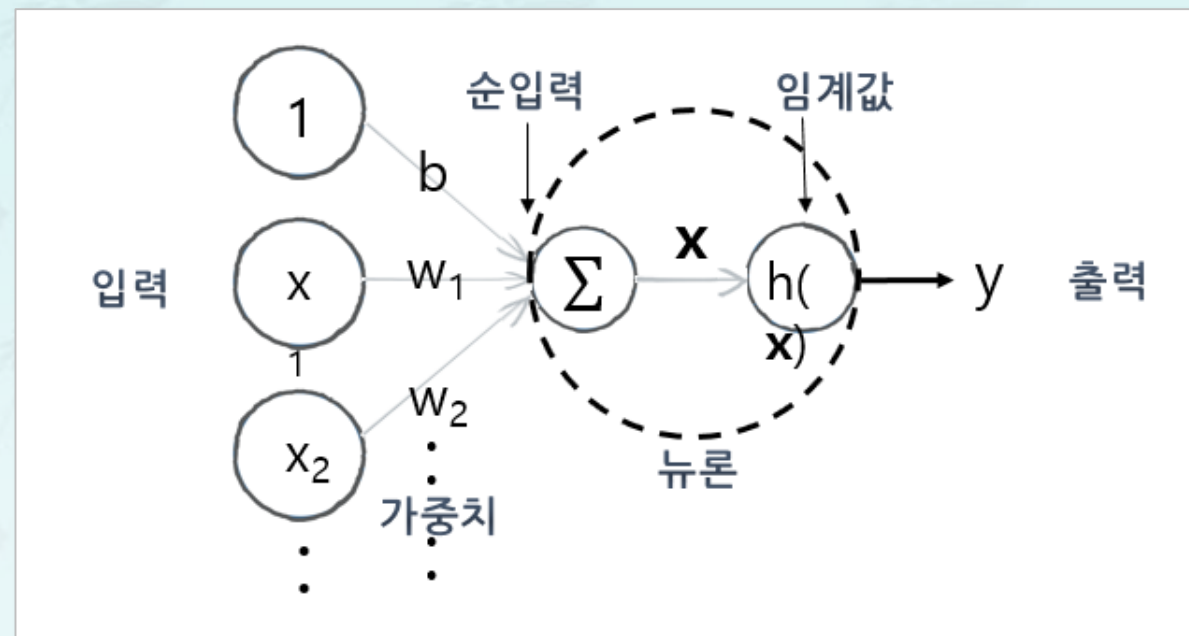
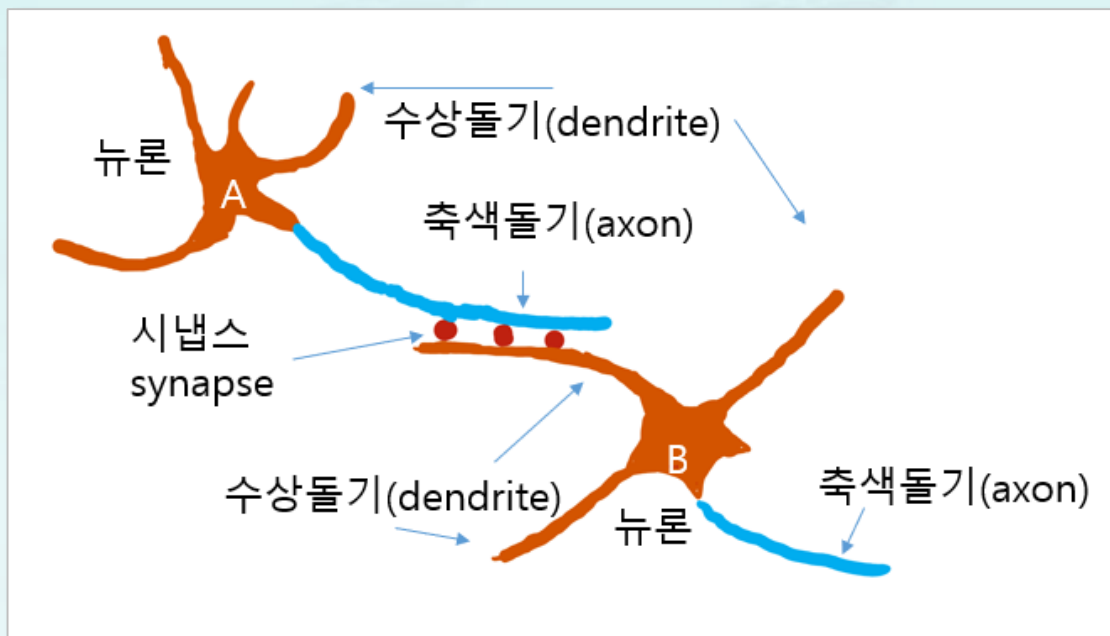
# 인공뉴론과 인공신경망

## ■ 인공뉴론



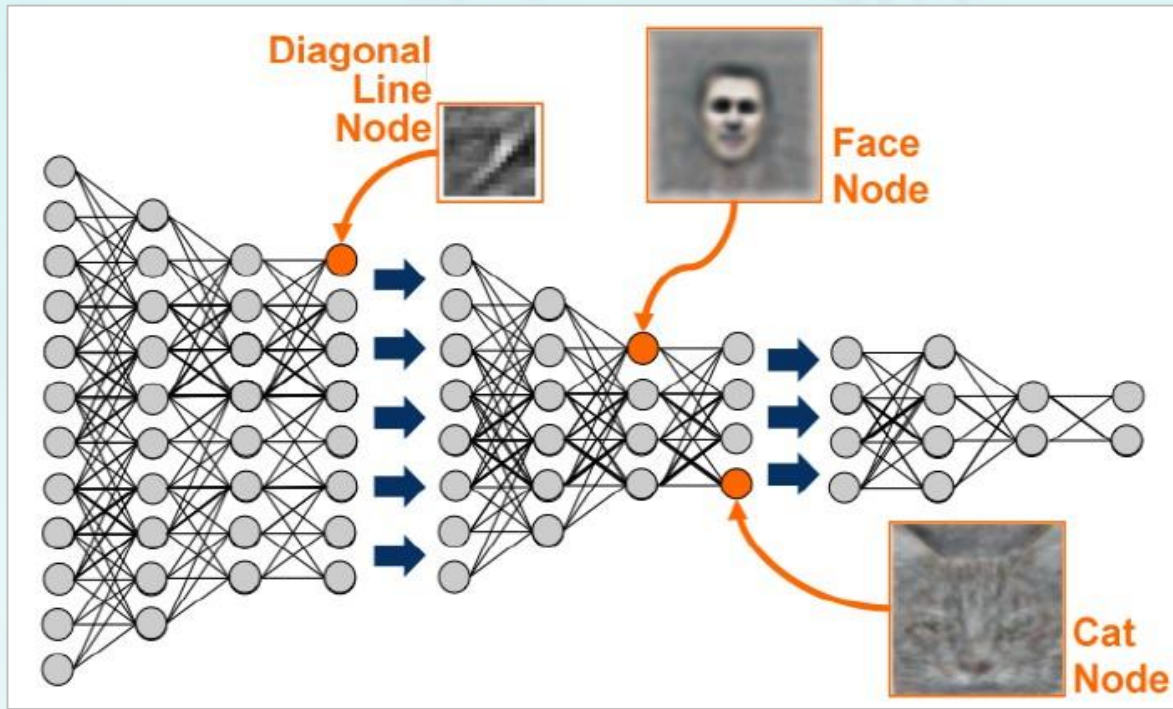
# 인공뉴론과 인공신경망

## ■ 인공뉴론



# 인공뉴론과 인공신경망

## ■ 인공신경망

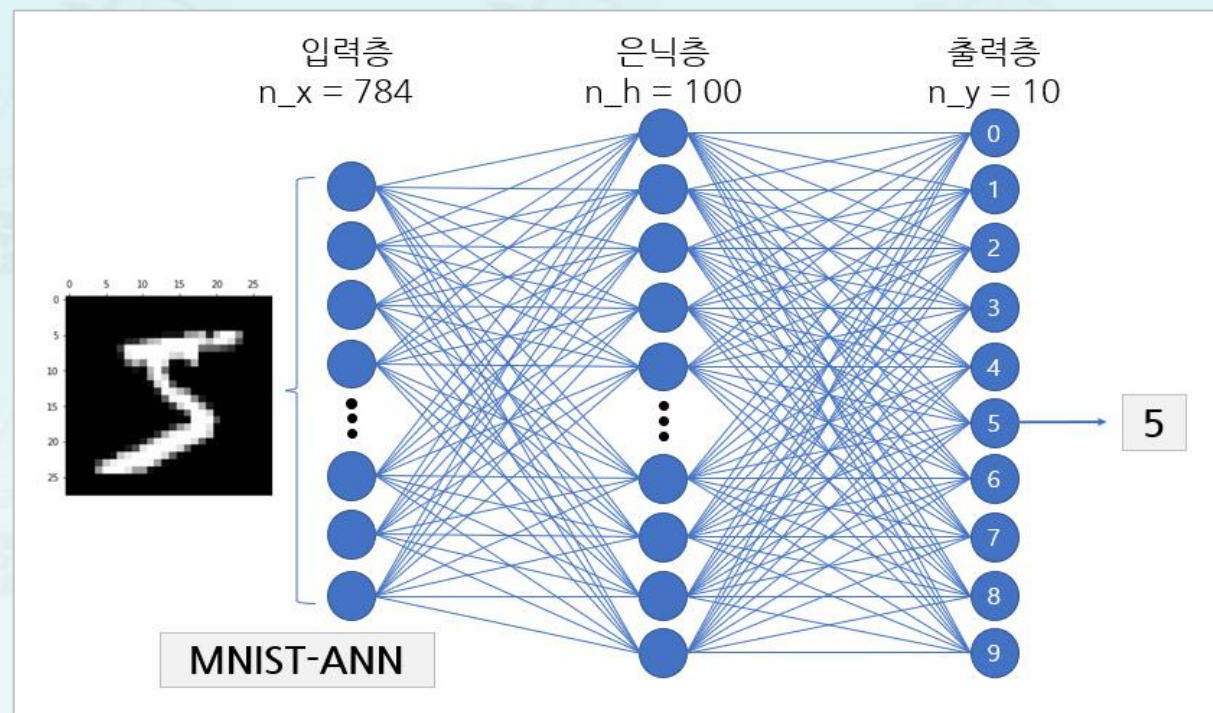
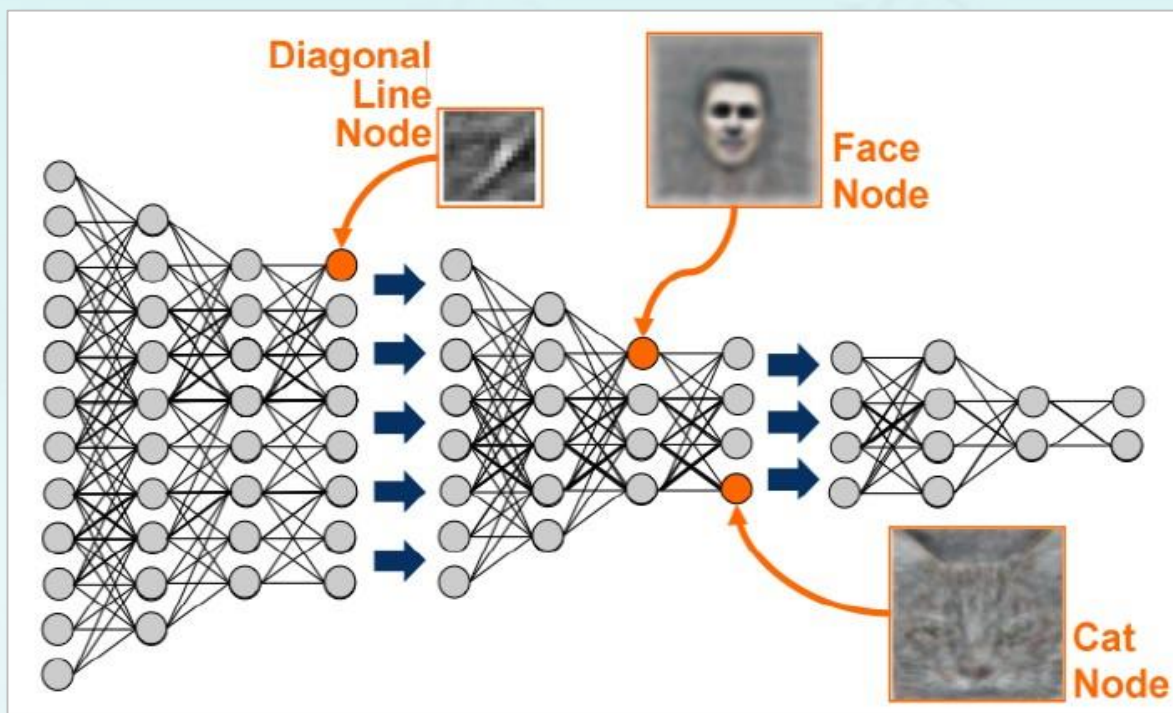


출처: Andrew Ng et al, "Building high-level features using large scale unsupervised learning", 2011

# 인공뉴론과 인공신경망

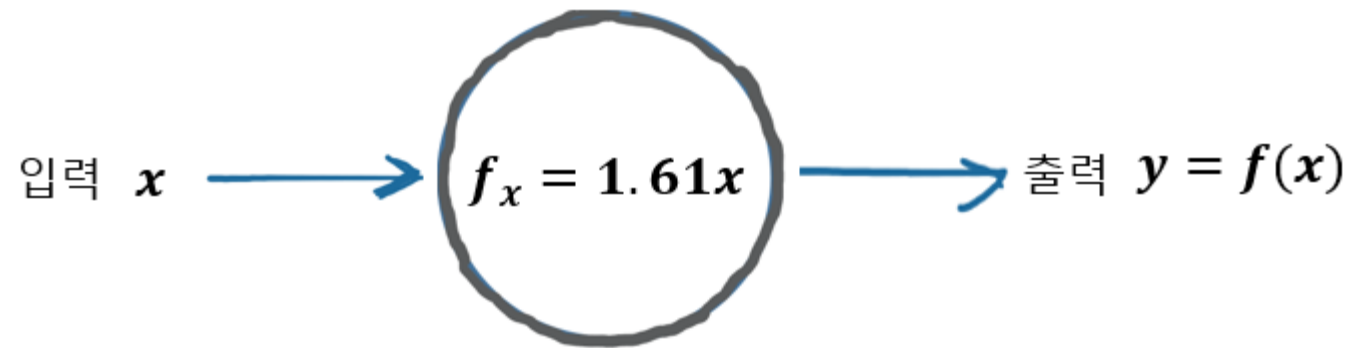
## ■ 인공신경망

## ■ 인공신경망



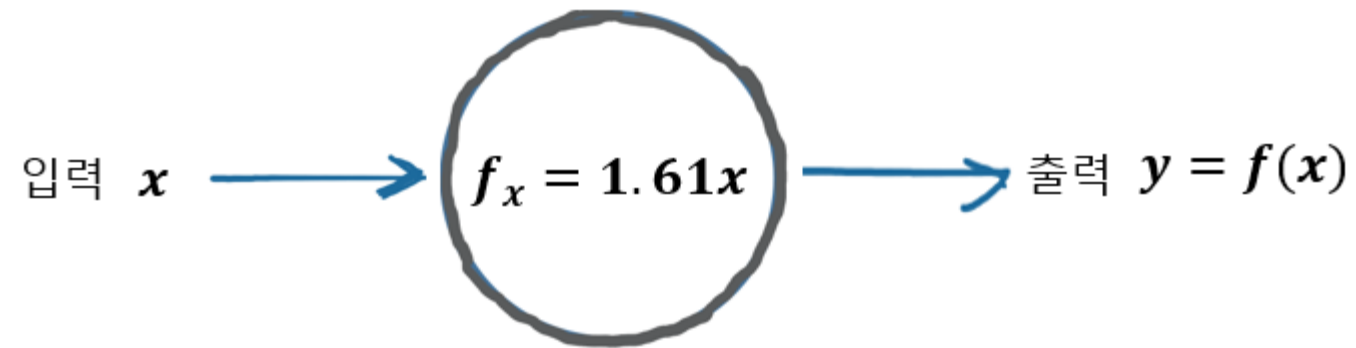
출처: Andrew Ng et al, "Building high-level features using large scale unsupervised learning", 2011

# 인공뉴런의 구현

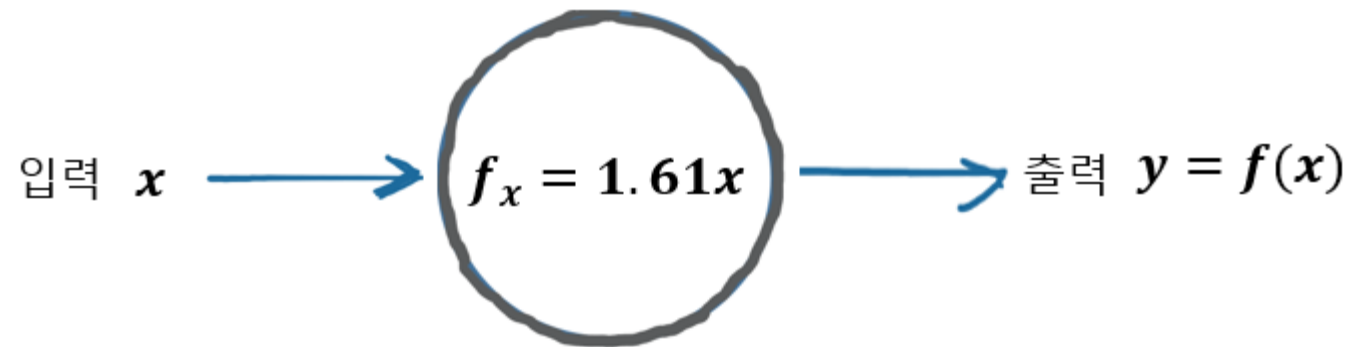




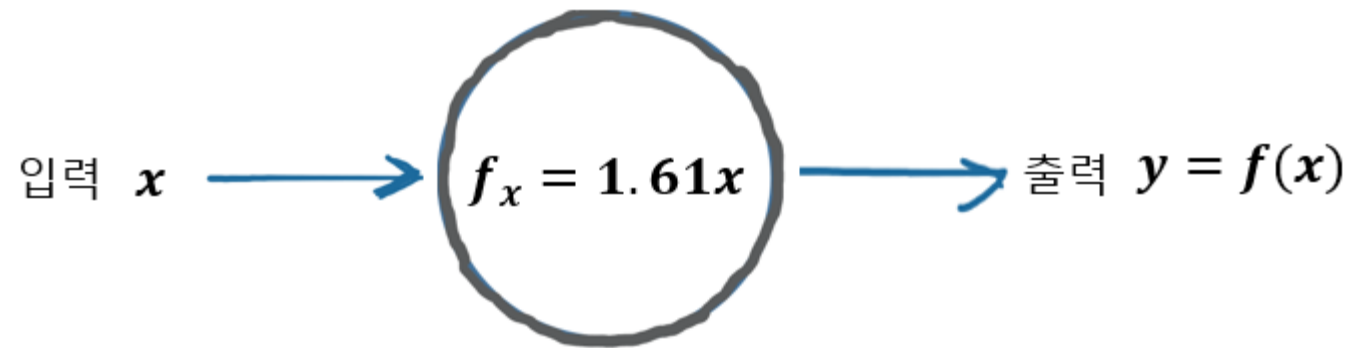
# 인공뉴런의 구현



# 인공뉴런의 구현



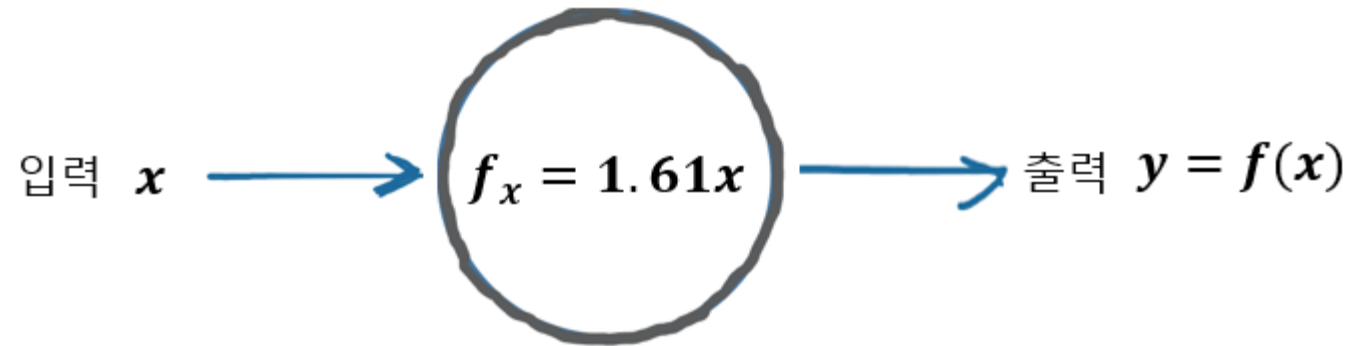
# 인공뉴런의 구현





## 인공뉴런의 구현

- `mileToKm()`와 `plotMileToKm()`



# 인공뉴론의 구현

- mileToKm()와 plotMileToKm()

```
1  import matplotlib.pyplot as plt
2  %matplotlib inline
3
4  def mileToKm(x):
5      """ x 마일을 Km로 변환하여 반환 """
6      return 1.61 * x
7
8  def plotMileToKm(x, y):
9      """ x, y의 값들을 그래프로 출력 """
10     plt.figure()
11     plt.plot(x, y)
12     plt.title('Mile to Km')
13     plt.xlabel('Mile')
14     plt.ylabel('Km')
15     plt.show()
```

# 인공뉴론의 구현

## ■ For 루프를 사용한 함수 호출

```
for mile in range(0, 5):  
    print('{}mi:{}'.format(mile, mileToKm(mile)))
```

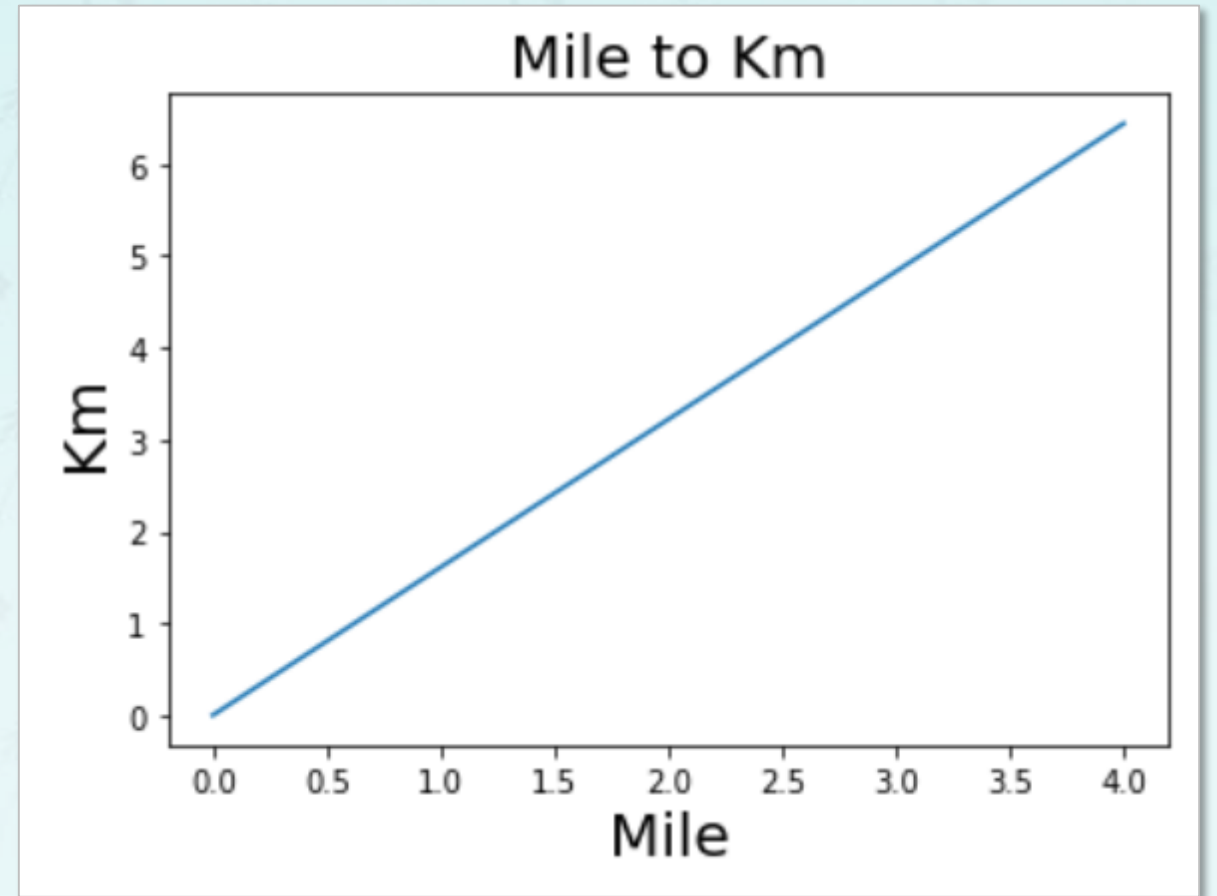
```
0mi:0.0km  
1mi:1.61km  
2mi:3.22km  
3mi:4.83km  
4mi:6.44km
```

```
1 import matplotlib.pyplot as plt  
2 %matplotlib inline  
3  
4 def mileToKm(x):  
5     """ x 마일을 Km로 변환하여 반환 """  
6     return 1.61 * x  
7  
8 def plotMileToKm(x, y):  
9     """ x, y의 값들을 그래프로 출력 """  
10    plt.figure()  
11    plt.plot(x, y)  
12    plt.title('Mile to Km')  
13    plt.xlabel('Mile')  
14    plt.ylabel('Km')  
15    plt.show()
```

# 인공뉴론의 구현

- 함수 결과의 시각화

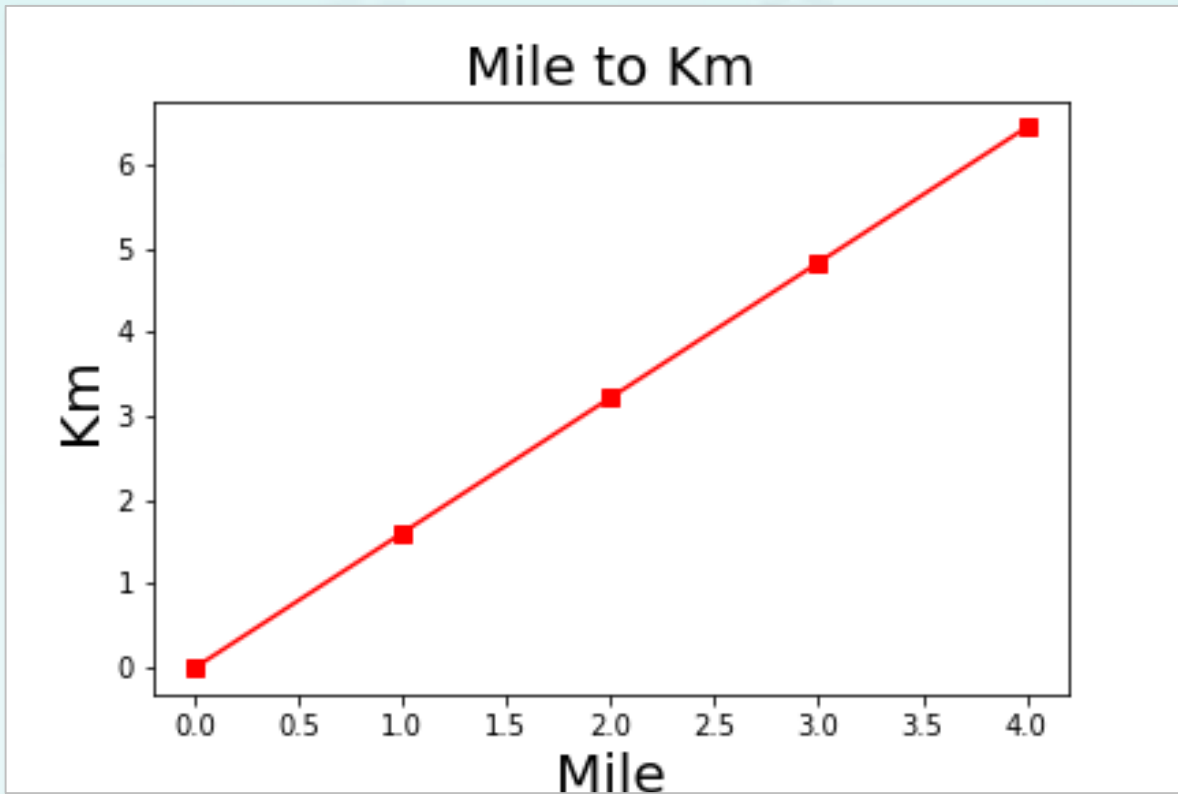
```
x = [0, 1, 2, 3, 4]
y = [0, 1.61, 3.22, 4.83, 6.44]
plotMileToKm(x, y)
```



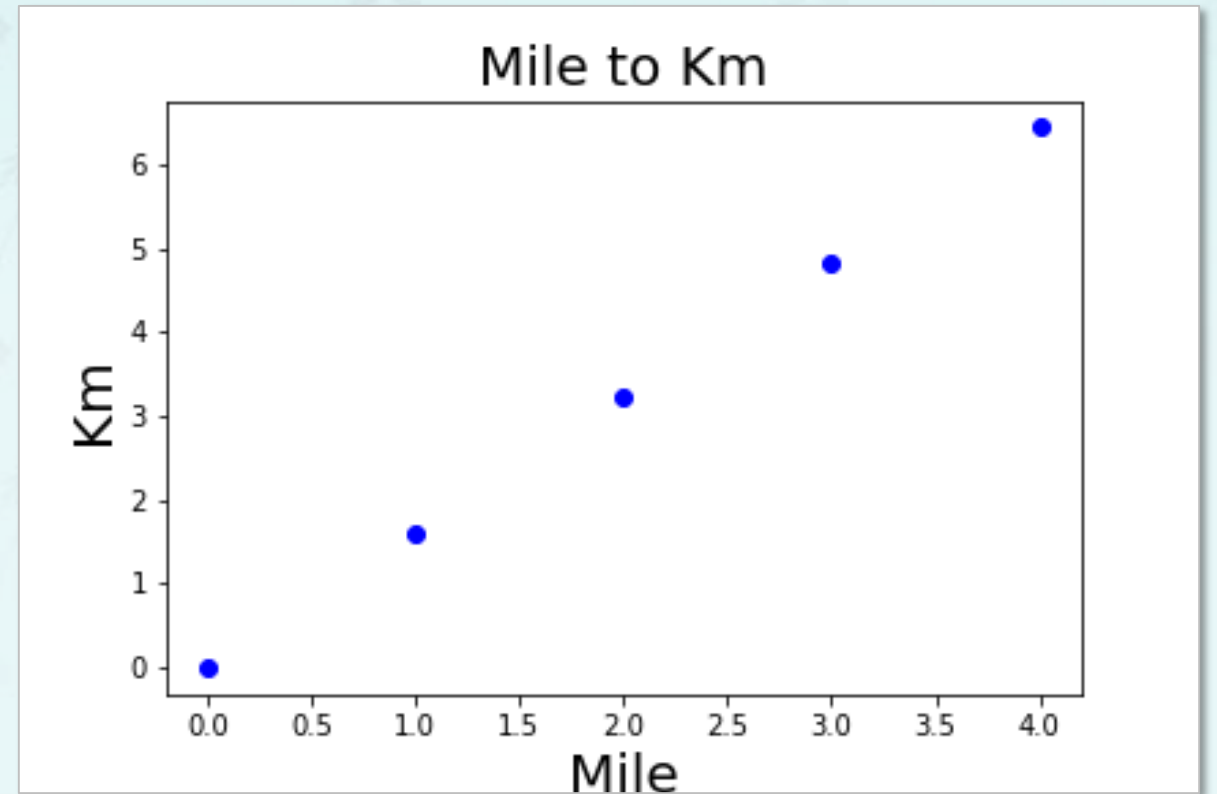
# 인공뉴론의 구현

- Marker의 모양: v, ^, <, >, ., o, s, p, x
- Marker의 색: b, c, m, r, y, k

`plt.plot(x, y, '-sr')`

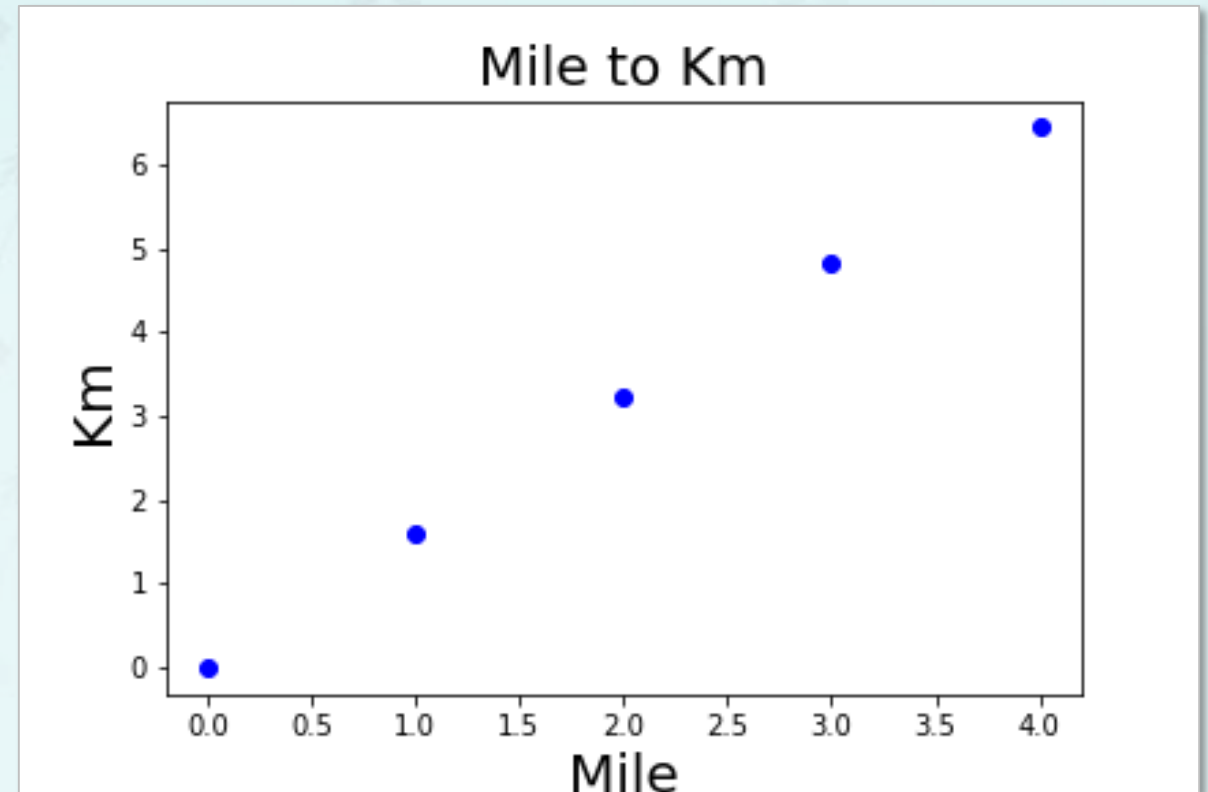


`plt.plot(x, y, 'ob')`



# 인공뉴론의 구현

- 손 코딩한  $x, y$  리스트 데이터  
 $x = [0, 1, 2, 3, 4]$   
 $y = [0, 1.61, 3.22, 4.83, 6.44]$



# 인공뉴론의 구현

---

- 손 코딩한 **x, y** 리스트 데이터  
**x = [0, 1, 2, 3, 4]**  
**y = [0, 1.61, 3.22, 4.83, 6.44]**
- **arange(start, end, step)**
  - 예: **arange(0,5) → 0, 1, 2, 3, 4**

# 인공뉴론의 구현

- **for** 문으로 리스트 **y** 만들기

```
1 y = []  
2 for mile in range(0, 5):  
3     y.append(mileToKm(mile))  
4 print(y)
```

[0.0, 1.61, 3.22, 4.83, 6.44]

- **y**값 초기화
- **arange(0,5)** → 0,1,2,3,4
- **append()** → 리스트 클래스 메소드



# 인공뉴론의 구현

- list comprehension으로 리스트 y 만들기

```
1 y = []  
2 for mile in range(0, 5):  
3     y.append(mileToKm(mile))  
4 print(y)
```

```
[0.0, 1.61, 3.22, 4.83, 6.44]
```

```
y = [ mileToKm(mile) for mile in range(0, 5) ]  
print(y)
```

```
[0.0, 1.61, 3.22, 4.83, 6.44]
```

# 인공뉴론의 구현

- 넘파이 사용 이점

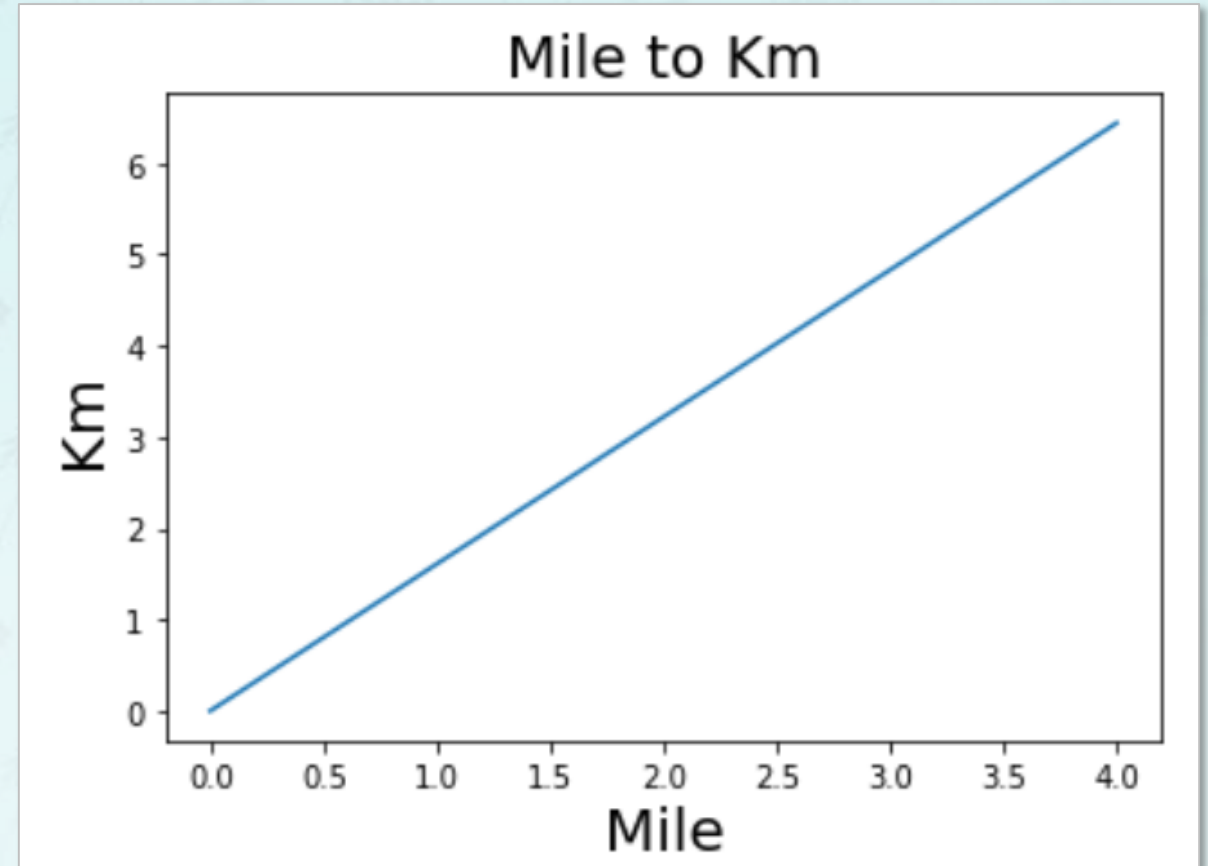
```
x = np.arange(0, 5)
y = mileToKm(x)
print(x)
print(y)
plotMileToKm(x, y)
```

# 인공뉴론의 구현

- 넘파이 사용 이점

```
x = np.arange(0, 5)  
y = mileToKm(x)  
print(x)  
print(y)  
plotMileToKm(x, y)
```

```
[0 1 2 3 4]  
[0.  1.61 3.22 4.83 6.44]
```



# 함수와 뉴론

---

- 학습 목표
  - 함수와 뉴론을 이해한다.
  - 인공뉴론과 인공신경망을 이해한다.
  - 첫 인공뉴론을 구현한다.
- 학습 내용
  - 함수와 뉴론
  - 인공뉴론과 인공신경망
  - 인공뉴론의 구현
- 차시 예고
  - **2-2** 넘파이 튜토리얼

2주차(1/3)

# 함수와 뉴론

파이썬으로 배우는 기계학습

한동대학교  
김영섭 교수

여러분 곁에 항상 열려 있는 K-MOOC 강의실에서 만나 뵙기를 바랍니다.