AI Programming 복습

한국항공대학교
KOREA AEROSPACE UNIVERSITY

1

목차

- 분류 (classification)
- 인공 뉴런 (Artificial Neuron)
- 인공 뉴런 (Artificial Neuron)의 연결
- 인공신경망의 분류(classification)
- 인공신경망의 학습
- 객체 지향 프로그래밍 (Object Oriented Programming)
- 상속(Inheritance)
- Numpy

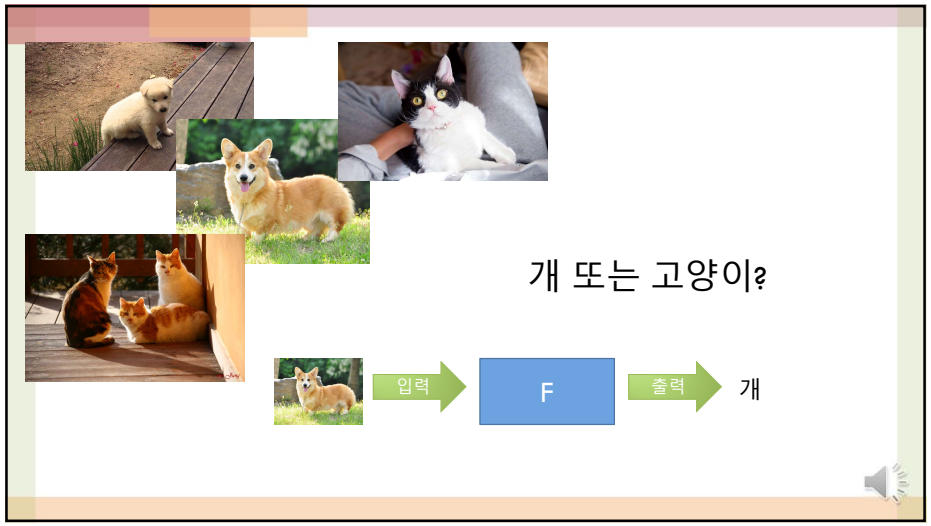
2

분류(classification)

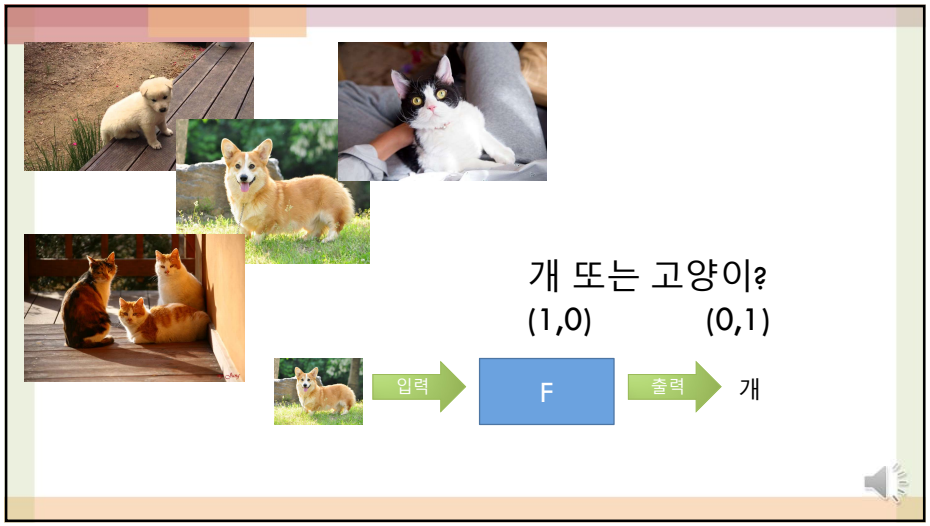
3

개 또는 고양이?

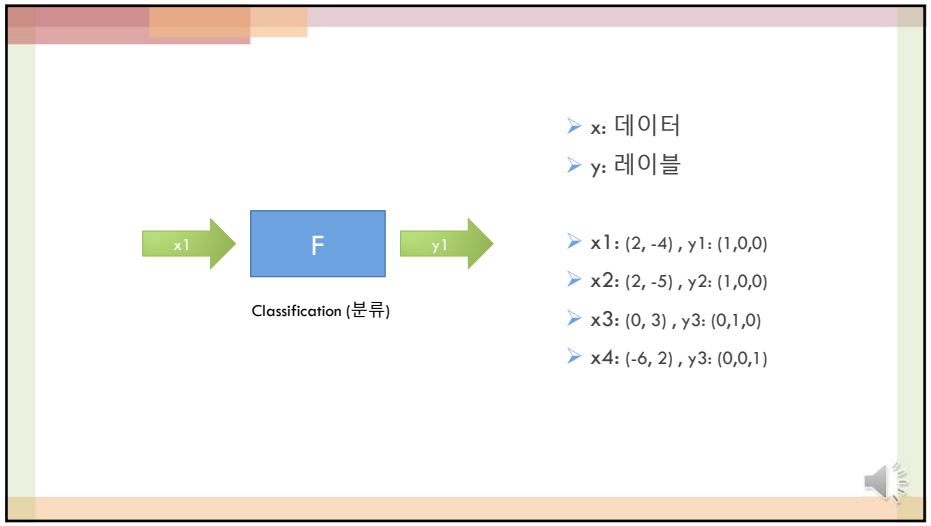
4


개 또는 고양이?
입력
F
출력
개

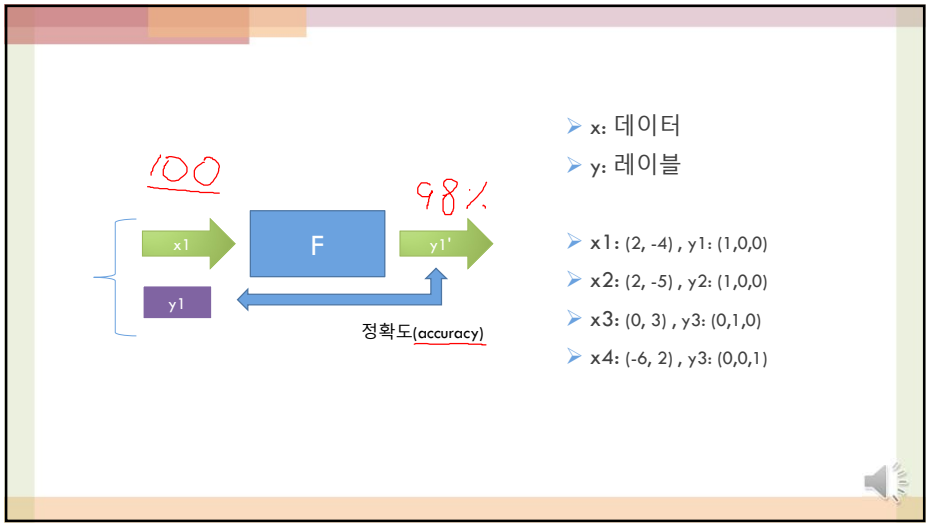
5


개 또는 고양이?
(1,0) (0,1)
입력
F
출력
개

6


x1
F
y1
Classification (분류)
x: 데이터
y: 레이블
x1: (2, -4) , y1: (1,0,0)
x2: (2, -5) , y2: (1,0,0)
x3: (0, 3) , y3: (0,1,0)
x4: (-6, 2) , y3: (0,0,1)

7


100
98%
x1
F
y1'
y1
정확도(accuracy)
x: 데이터
y: 레이블
x1: (2, -4) , y1: (1,0,0)
x2: (2, -5) , y2: (1,0,0)
x3: (0, 3) , y3: (0,1,0)
x4: (-6, 2) , y3: (0,0,1)

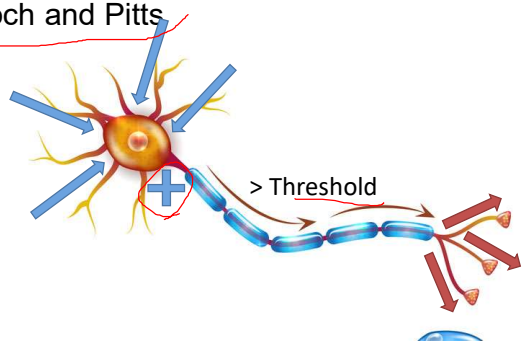
8

인공 뉴런 (Artificial Neuron)

9

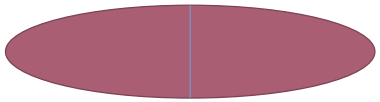
Artificial Neurons (Perceptrons)

- McCulloch and Pitts



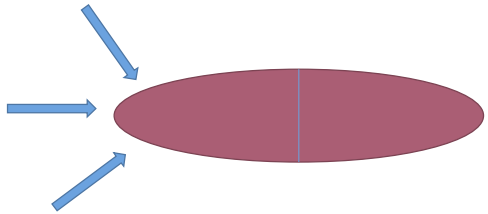
10

인공 뉴런(Artificial Neuron)



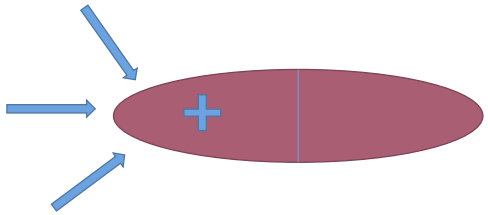
11

인공 뉴런(Artificial Neuron)



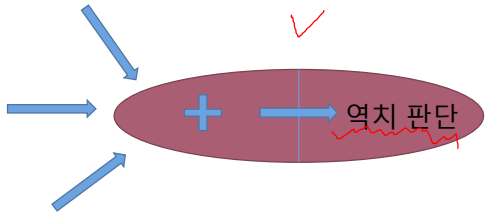
12

인공 뉴런(Artificial Neuron)



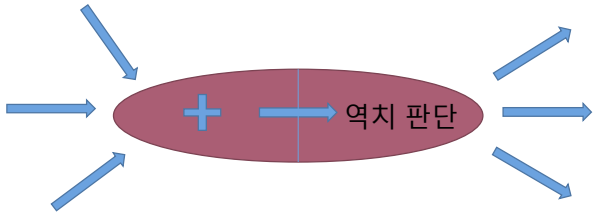
13

인공 뉴런(Artificial Neuron)



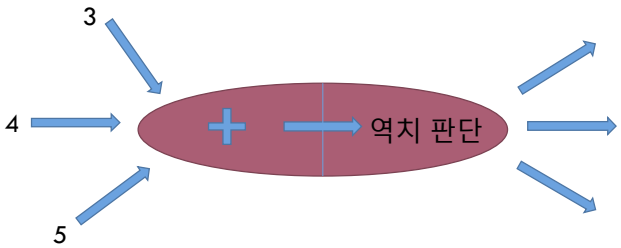
14

인공 뉴런(Artificial Neuron)



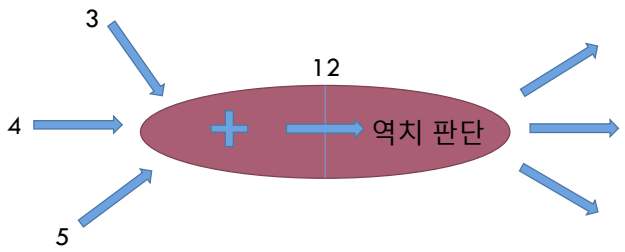
15

인공 뉴런(Artificial Neuron)



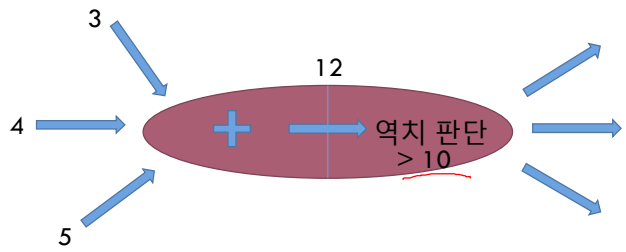
16

인공 뉴런(Artificial Neuron)



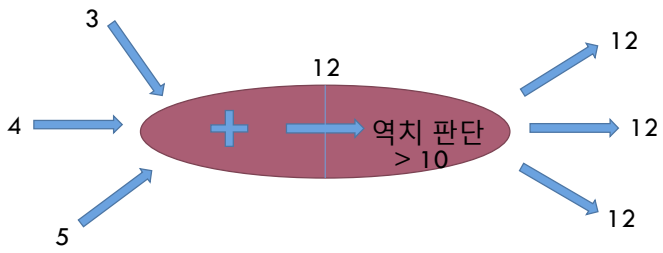
17

인공 뉴런(Artificial Neuron)



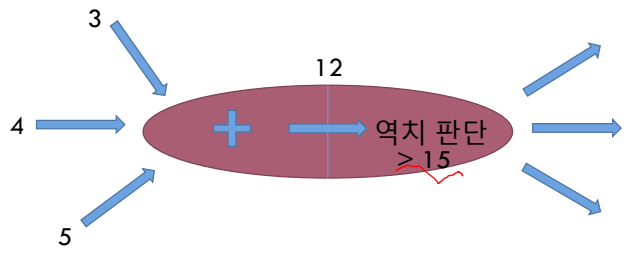
18

인공 뉴런(Artificial Neuron)



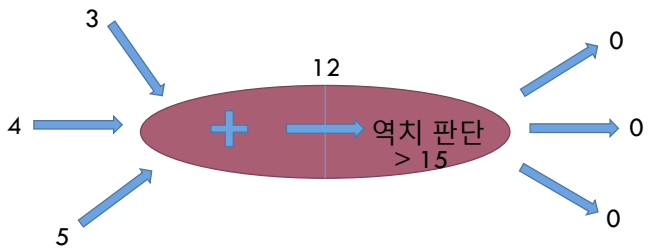
19

인공 뉴런(Artificial Neuron)



20

인공 뉴런(Artificial Neuron)

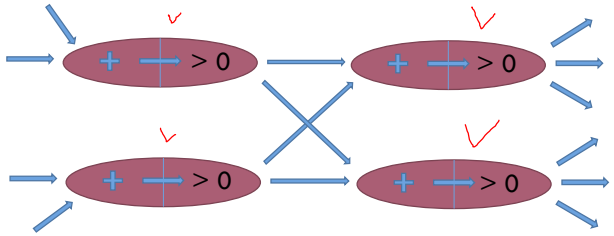


21

인공 뉴런 (Artificial Neuron)의 연결

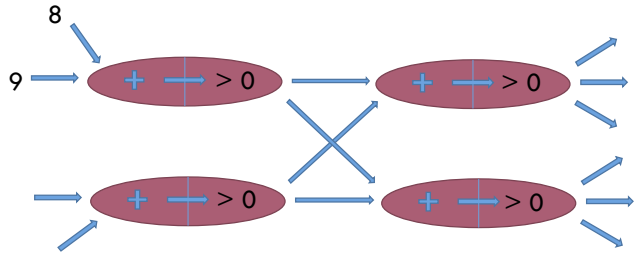
22

인공 뉴런(Artificial Neuron)



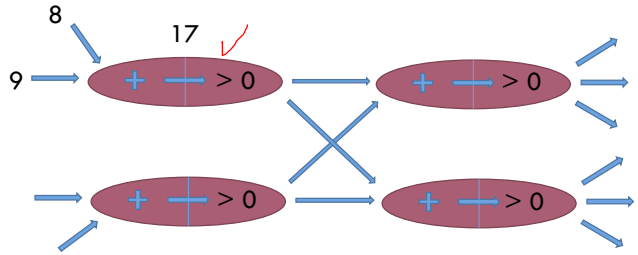
23

인공 뉴런(Artificial Neuron)



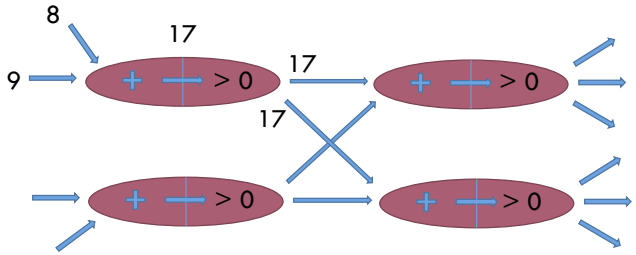
24

인공 뉴런(Artificial Neuron)



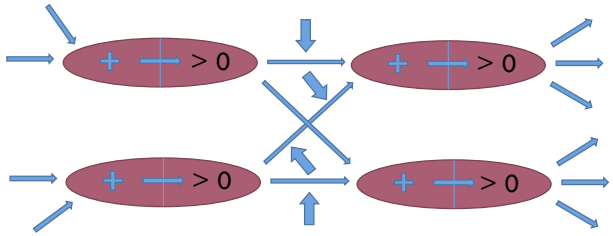
25

인공 뉴런(Artificial Neuron)



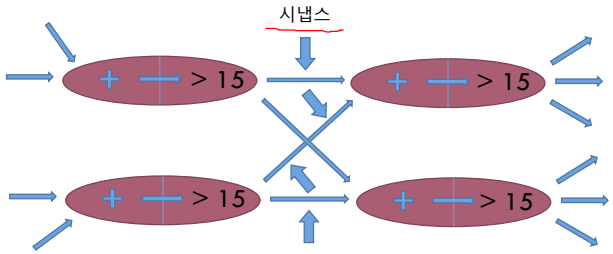
26

인공 뉴런(Artificial Neuron)



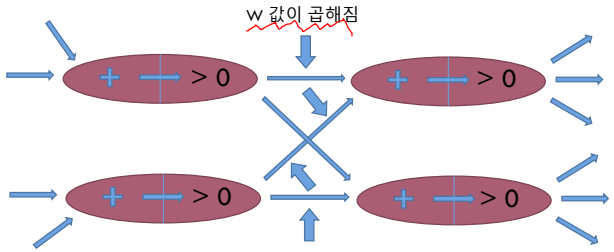
27

인공 뉴런(Artificial Neuron)



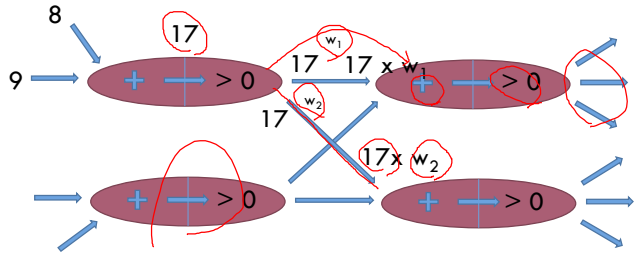
28

인공 뉴런(Artificial Neuron)



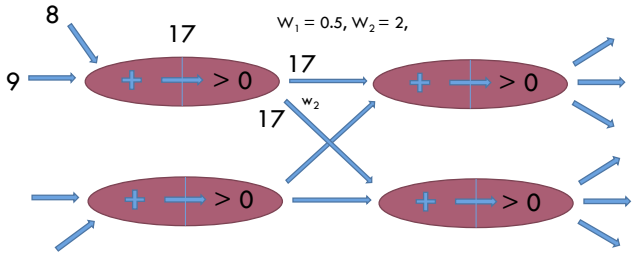
29

인공 뉴런(Artificial Neuron)



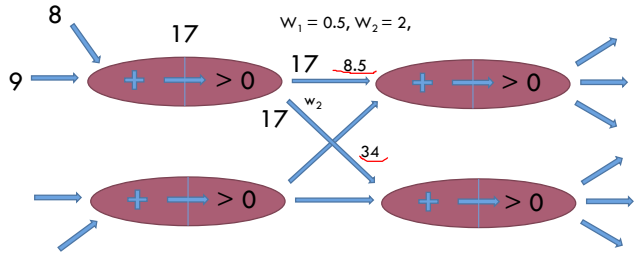
30

인공 뉴런(Artificial Neuron)



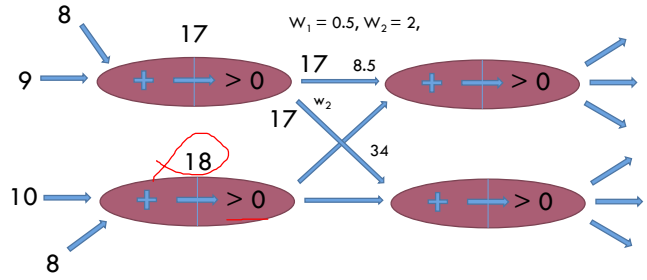
31

인공 뉴런(Artificial Neuron)



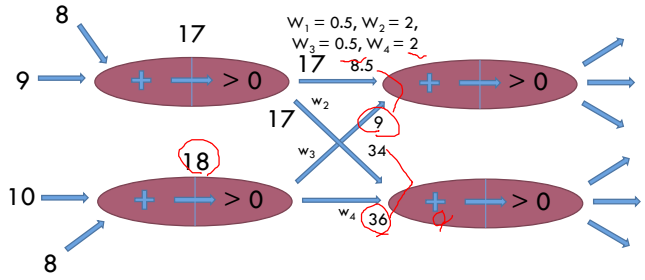
32

인공 뉴런(Artificial Neuron)



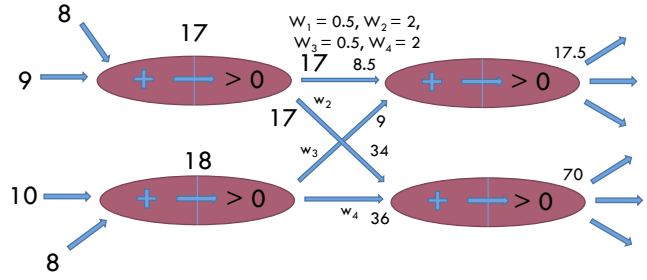
33

인공 뉴런(Artificial Neuron)



34

인공 뉴런(Artificial Neuron)



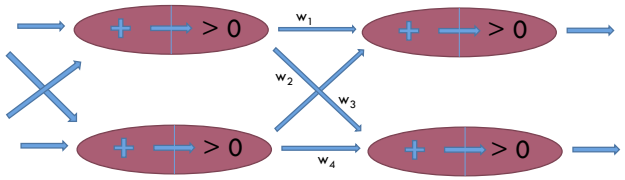
35

인공신경망의 분류(classification)

36

인공신경망을 이용한 분류

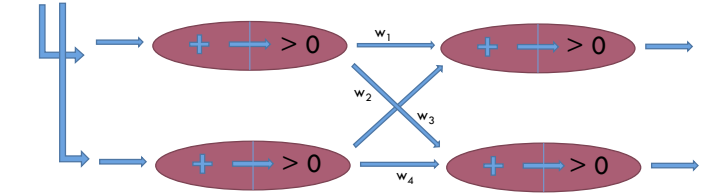
x1: (2, -4) , y1: (1,0)



37

인공신경망을 이용한 분류

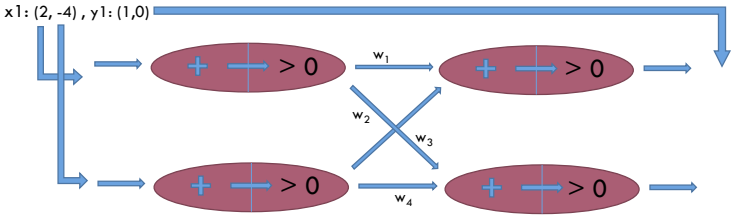
x1: (2, -4) , y1: (1,0)



38

인공신경망을 이용한 분류

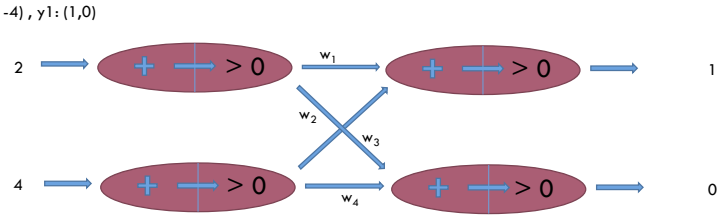
x1: (2, -4) , y1: (1,0)



39

인공신경망을 이용한 분류

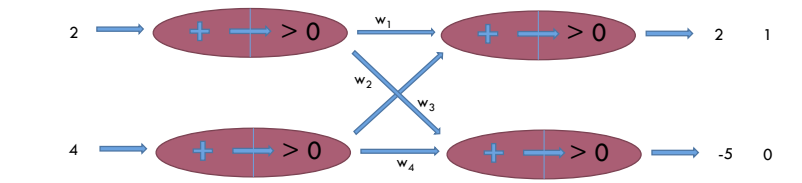
x1: (2, -4) , y1: (1,0)



40

인공신경망을 이용한 분류

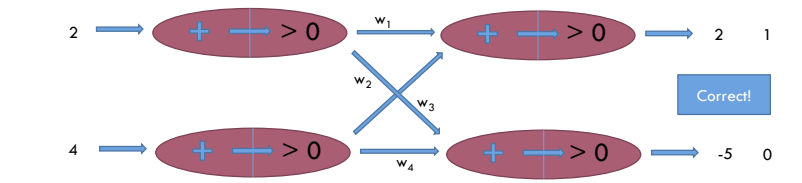
x1: (2, -4) , y1: (1,0)



41

인공신경망을 이용한 분류

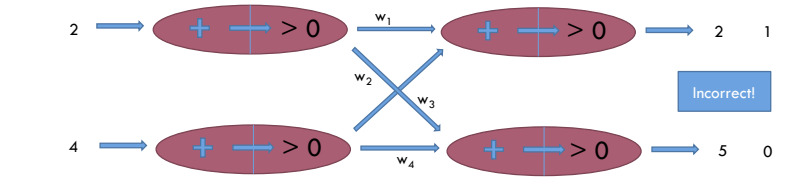
x1: (2, -4) , y1: (1,0)



42

인공신경망을 이용한 분류

x1: (2, -4) , y1: (1,0)

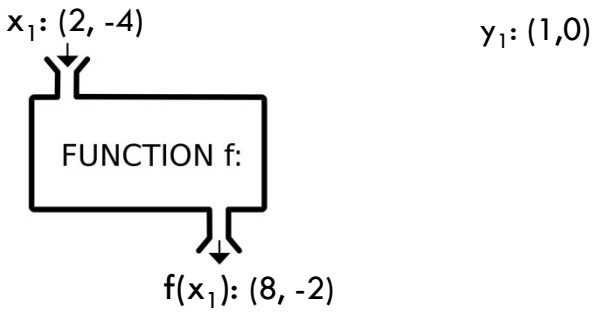


43

인공신경망의 학습

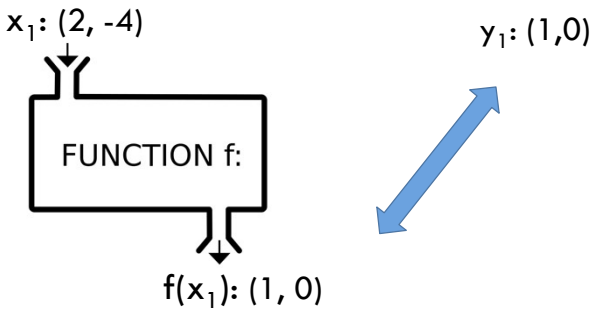
44

인공신경망을 이용한 분류



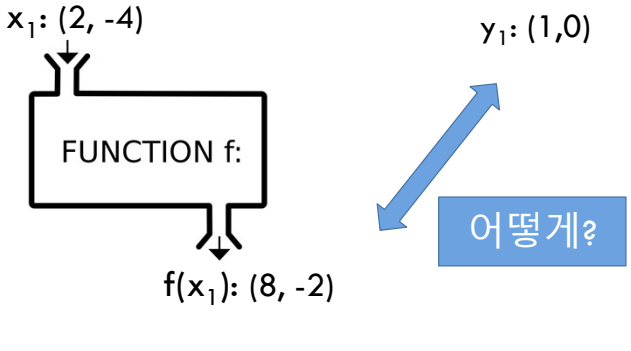
45

인공신경망을 이용한 분류



46

인공신경망을 이용한 분류



47

Loss function

f(x) ~ y 와 얼마나 가까운지 판단!

48

Loss function

f(x) ~ y 와 얼마나 가까운지 판단!

$f(x_1)$: (8, -2) ~ y_1: (1,0)

49

Loss function

f(x) ~ y 와 얼마나 가까운지 판단!

$f(x_1)$: (8, -2) ~ y_1: (1,0)

문제 1) SCALE 이 안 맞다 (양/음수)

50

Loss function

$$\text{softmax}(\mathbf{a}) = \frac{e^{a_i}}{\sum_j e^{a_j}}$$

$\mathbf{a}$ = [-0.21,0.47,1.72]

$\sum_j e^{a_j} = e^{-0.21} + e^{0.47} + e^{1.72} = 8$

$\text{softmax}(\mathbf{a}) = [\frac{e^{-0.21}}{8}, \frac{e^{0.47}}{8}, \frac{e^{1.72}}{8}]$

y = [0.1,0.2,0.7]

51

Loss function

f(x) ~ y 와 얼마나 가까운지 판단!

$f(x_1)$: (8, -2) ~ y_1: (1,0)

문제 2) 어떻게 가까운지 판단하지?

Root Mean Square Deviation?

52

Loss function

Cross Entropy:

$$H(p,q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

p, q 모두 확률 이어야함

p → label

q → f(x)

53

Loss function

54

Loss function

Cross Entropy:

$$H(p,q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

언제 가장 큰 값을 가질까?

p→ [0,0,1]

q → [0,0,1]

55

Loss function

Cross Entropy:

$$H(p,q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

p→ [0,0,1]

q → [0,0,1]

-1*(0 * log (0) + 0*log(0) + 1 * log(1)) ➔ 0

56

Loss function

Cross Entropy:

$$H(p,q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

p→ [0,1,0]
q → [0,0,1]
-1*(0 * log (0) + 1*log(0) + 0 * log(1)) ➔ inf

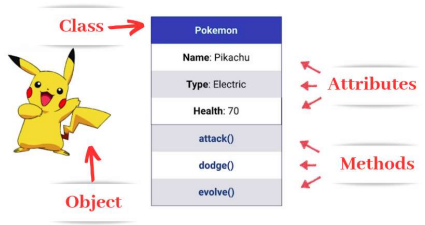
57

객체 지향 프로그래밍
(Object Oriented Programming)

58

객체 (Object) 의 정의

- 객체 (Object):
  데이터(실체)와 그 데이터에 관련되는 동작(절차, 방법, 기능)을 모두 포함한 개념



59

객체 (Object) 의 정의

- 객체 (Object):
  어떻게 정의할까?

60

객체 (Object) 의 정의

- 객체 (Object):

어떻게 정의할까?

CLASS

필드 (field)를 미리 결정

객체를 형성

사용!

61

클래스(CLASS) 정의

- 형식 (Syntex)

class (클래스 이름):

클래스를 사용하겠다

62

초기화 메서드(METHOD)

```
class Person:
    def __init__(self, name):
        self.name = name
    def say_hi(self):
        print 'Hello, my name is', self.name

p = Person('Swaroop')
p.say_hi()
```

63

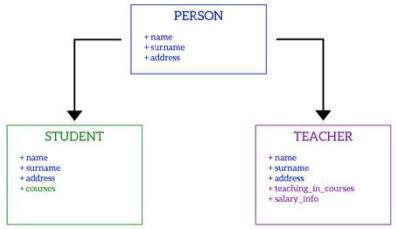
상속(Inheritance)

64

상속(Inheritance)

상속(Inheritance)

- 상속 (Inheritance):
  2. 뒤를 이음.



65

상속하는 법

- 형식 (Syntex)

class (클래스 이름)(상속 클래스):

66

상속하는 법

- 형식 (Syntex)

class student(상속 클래스):

67

Numpy

AI Programming

한국항공대학교 KOREA AEROSPACE UNIVERSITY

68

Numpy

Numpy

- 프로그램 라이브러리
- 수학의 함수 등 여러 계산을 편리하게 하기 위한 라이브러리

```
>>> import numpy as np
```

69

np.array([ ])

```
>>> a = np.array([1,2,3,4,5])
>>> type(a)
<class 'numpy.ndarray'>
```

- class → ndarray
- 객체
- Attribute 과 Methods

< 사랑에 빠진 합리주의자(NT)들 >

INTP

Error 404 : 객체 love 가 정의되지 않았습니다

ENTP

사랑? 처음보는건데 일단 다 뜯어봐서 어떤건지 확인해야 겠당

70

List vs ndarray

```
>>> a = [1,2,3,4,5]
>>> b = [1,2,3,4,5]
>>> a+b
[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
>>> result = []
... for i,j in zip(a,b):
...     result.append(i+j)
...
>>> result
[2, 4, 6, 8, 10]
```

```
>>> a = np.array([1,2,3,4,5])
>>> b = np.array([1,2,3,4,5])
>>> a + b
array([ 2,  4,  6,  8, 10])
```

71

List vs ndarray

```
>>> a = [1,2,3,4,5]
>>> b = [1,2,3,4,5]
>>> 10*a
[1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
>>> a/10
Traceback (most recent call last):
  File "<input>", line 1, in <module>
TypeError: unsupported operand type(s) for /: 'list' and 'int'
>>> a+1
Traceback (most recent call last):
  File "<input>", line 1, in <module>
```

```
>>> import numpy as np
>>> a = np.array(a)
>>> b= np.array(b)
>>> a
array([1, 2, 3, 4, 5])
>>> b
array([1, 2, 3, 4, 5])
>>> a+b
array([ 2,  4,  6,  8, 10])
>>> a*10
array([10, 20, 30, 40, 50])
>>> a/10
array([0.1, 0.2, 0.3, 0.4, 0.5])
>>>
```

72

Indexing

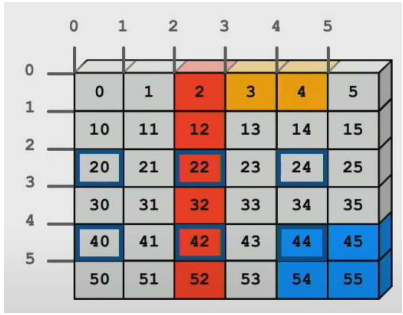
var[lower:upper:step]

```
#                -5 -4 -3 -2 -1
# indices:        0  1  2  3  4
>>> a = np.array([10,11,12,13,14])
```

```
# [10, 11, 12, 13, 14]
>>> a[1:3]
array([11, 12])

# negative indices work also
>>> a[1:-2]
array([11, 12])
>>> a[-4:3]
array([11, 12])
```

73

Indexing

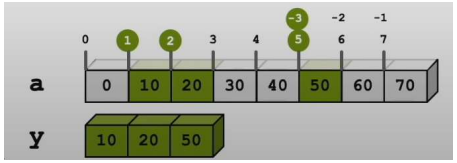



```
>>> a[0, 3:5]
array([3, 4])

>>> a[4:, 4:]
array([[44, 45],
       [54, 55]])

>>> a[:, 2]
array([2, 12, 22, 32, 42, 52])
```

```
>>> a[2::2, ::2]
array([[20, 22, 24],
       [40, 42, 44]])
```

74

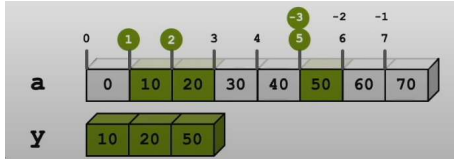
Advanced Indexing




```
>>> a = np.arange(0, 80, 10)

# fancy indexing
>>> indices = [1, 2, -3]
>>> y = a[indices]
>>> print(y)
[10 20 50]

# this also works with setting
>>> a[indices] = 99
>>> a
array([ 0, 99, 99, 30, 40, 99,
60, 70])
```

75

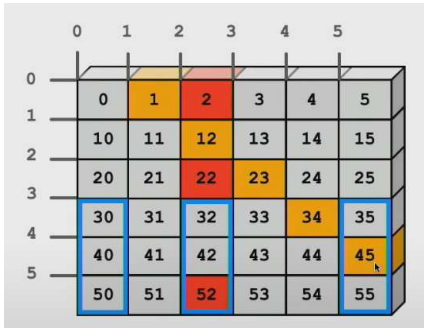
Advanced Indexing




```
>>> a = np.arange(0,80,10)
>>> indices = np.array([0,1,1,0,0,1,0,0],dtype=bool)
>>> y = a[indices]
>>> y
array([10, 20, 50])
```

76

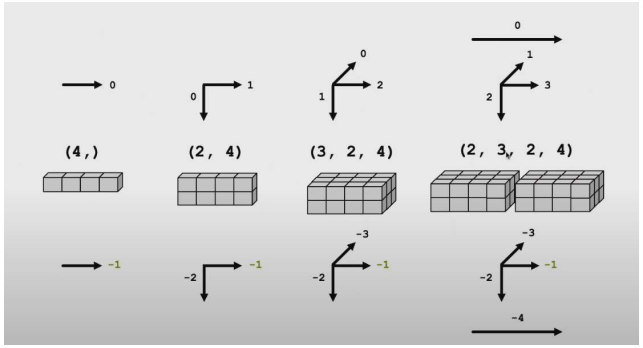
Advanced Indexing




```
>>> a[[0, 1, 2, 3, 4],
...   [1, 2, 3, 4, 5]]
array([ 1, 12, 23, 34, 45])

>>> a[3:, [0, 2, 5]]
array([[30, 32, 35],
       [40, 42, 45],
       [50, 52, 55]])

>>> mask = np.array(
...     [1, 0, 1, 0, 0, 1],
...     dtype=bool)
>>> a[mask, 2]
array([2, 22, 52])
```

77

ndarray multidimension



78

연산 규칙

1. 고차원 array들의 연산은 shape 이 맞는 지 확인
2. 기본적인 연산 (+, -, *, /, exp, …)는 원소 마다 계산
3. 차원이 줄어드는 연산 (mean, std, sum …)는 array 전체에 적용된다. 다만, Axis를 줄 수도 있다.
4. nan의 계산 결과는 nan이다.

79

where

Mathematical functions
- sum, prod
- min, max, argmin, argmax
- ptp (max – min)

Statistics
- mean, std, var

Truth value testing
- any, all

```
>>> a = np.arange(0,40)
>>> a
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39])
>>> np.ptp(a)
39
>>> a.ptp()
39
>>> a.mean()
19.5
>>> a.std()
11.543396380615196
>>> a.var()
133.25
```

80

목차

- 분류 (classification)
- 인공 뉴런 (Artificial Neuron)
- 인공 뉴런 (Artificial Neuron)의 연결
- 인공신경망의 분류(classification)
- 인공신경망의 학습
- 객체 지향 프로그래밍 (Object Oriented Programming)
- 상속(Inheritance)
- Numpy

81