

16일차

1. 수업 내용

- 9일차~15일차 복습

2. 셀프 Q&A

xor 연산 (day 10)

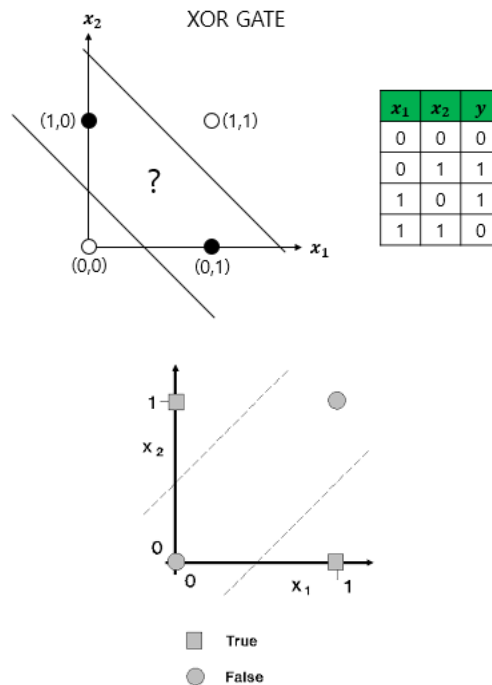
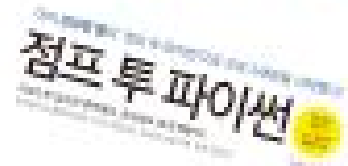
자동등록방지를 위해 보안절차를 거치고 있습니다.

<http://solarisailab.com/archives/1206>

점프 투 파이썬

점프 투 파이썬 오프라인 책(개정판) 출간 !! (2019.06) ** * [책 구입 안내](<https://wikidocs.net/4321>) 이 책은 파이썬 ...

<https://wikidocs.net/24958>



두 가지 예시.

CNN (Convolution Neural Network)

Convolutional Neural Network

Convolutional neural network(CNN 또는 ConvNet)은 수동으로 특징을 추출할 필요 없이 데이터로부터 직접 학습하는 딥러닝을 위한 신경망 아키텍처입니다. CNN은 영상에서 객체, 얼굴, 장면 인식을 위한 패턴을 찾을 때 특히 유용합니다. 오디오, 시계열, 신호 데이터와 같이 영상 이외의 데이터를 분류하는 데도 효과적입니다.

🔗 <https://kr.mathworks.com/discovery/convolutional-neural-network-matlab.html>

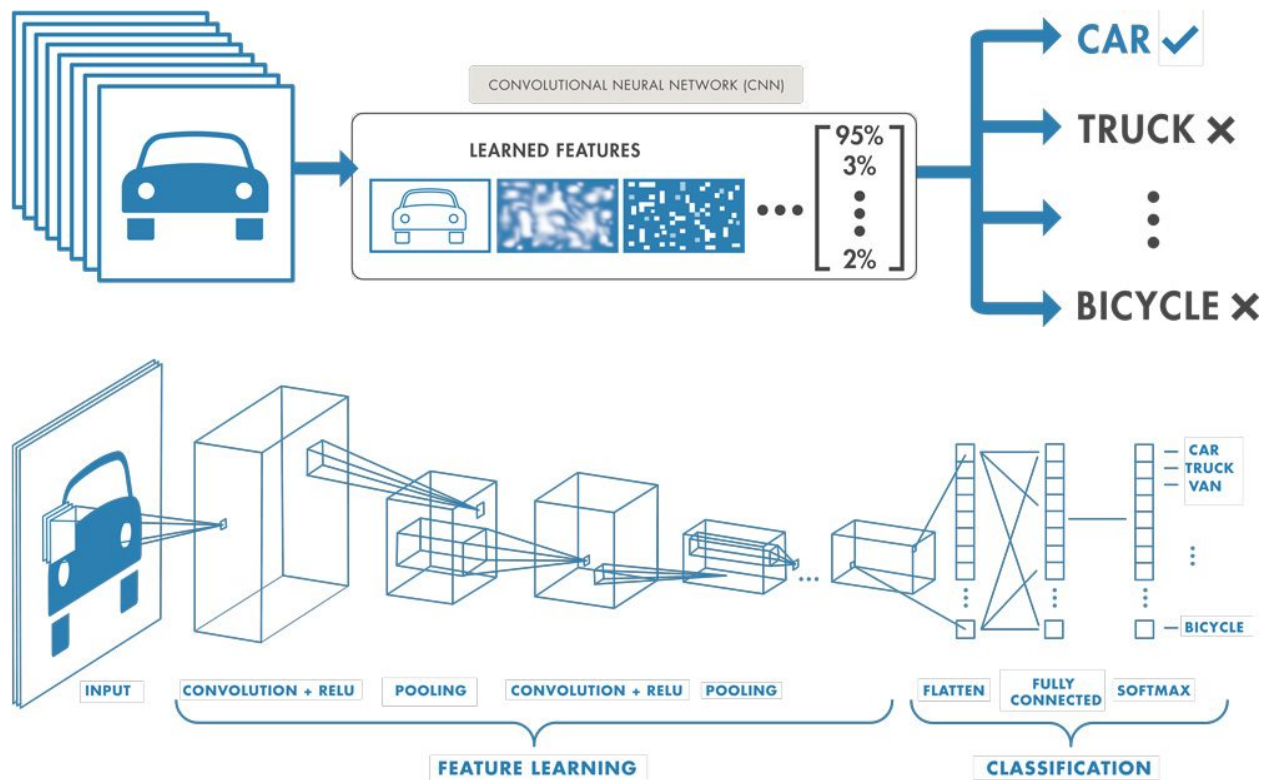


⇒ 자동으로 데이터로부터 직접 학습하는, 딥러닝을 위한 신경망 아키텍처.

⇒ 영상에서 객체, 얼굴, 장면 인식을 위한 패턴을 찾을 때 유용.

⇒ 오디오, 시계열, 신호 데이터 등 영상 외의 데이터 분류하는 데도 효과적.

⇒ 자율 주행 차량, 얼굴 인식 응용 분야에서도 쓰임



np.where : 조건 만족 인덱스

[Numpy] np.where 사용법 : 조건 만족하는 위치 인덱스 찾기

파이썬 넘파이 np.where : 조건 만족 인덱스 반환 파이썬에서 넘파이를 사용하다보면, np.where 함수를 자주 만나게 됩니다. 만일때 마다 복잡해보이지만, 사실 알고보면 매우 간단한 np.where 함수의 사용 방법을..

🔗 <https://jimmy-ai.tistory.com/46>



```
import numpy as np

a = np.arange(5, 15) # array([ 5,  6,  7,  8,  9, 10, 11, 12, 13, 14])

np.where(a > 10) # 6 ~ 9번 위치가 만족 -> (array([6, 7, 8, 9]),)

np.where(a % 3 == 0) # 1, 4, 7번 위치가 6, 9, 12로 만족 -> (array([1, 4, 7]),)
```

```
# 2차원 이상이면 axis를 기준으로 인덱스 번호를 가져옴

a = np.array([[15, 8, 12], [11, 7, 3]])
np.where(a > 10) # (array([0, 0, 1]), array([0, 2, 0]))
```

```
a = np.array([[15, 8, 12], [11, 7, 3]])
np.where(a > 10, a, 10) # a가 10보다 크면 그대로, 그렇지 않으면 10으로 설정
# array([[15, 10, 12],
#        [11, 10, 10]])

b = np.array([[0, -1, 2], [-2, -1, -3]])
np.where(a > 10, a * 2, b) # a가 10보다 크면 2배, 그렇지 않으면 b의 해당 위치 값으로 설정
# array([[30, -1, 24],
#        [22, -1, -3]])
```

np.predict

<https://blog.naver.com/teorw272/22207348911>

```
heights = [[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71],[72]]
weights = [[115],[117],[120],[123],[126],[129],[132],[135],[139],[142],[146],[150],[154],[159],[164]]

from sklearn.linear_model import LinearRegression

model = LinearRegression() #model 생성
model.fit(X = heights, y = weights)


eights_model = model.predict([[64]])[0][0]
round(weights_model,2)

weights_model = model.predict([[78]])[0][0]
round(weights_model,2)
```

argmax, argmin

[Tensorflow][기초] argmax , argmin 함수 소개 및 실습

Tensorflow [Tensorflow][기초] argmax , argmin 함수 소개 및 실습 원핫인코딩때문에 자주 사용하게 되는 argmax 함수에 대해서 알아보자. one-hot(원핫)인코딩이란? 단 하나의 값만 True이고 나머지는 모두 False인 인코딩을 말한다. 즉, 1개만 Hot(True)이고 나머지는 Cold(False)이다. 예를들면 [0, 0, 0, 0, 1]이다. 5번째(Zero-based 인덱스이므로 4)만 1이

 <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=wideeyed&logNo=221164668953>

```
# 1번째(0 인덱스) 자원 제거하고 2번째(1 인덱스) 자원에 가장 큰 값의 인덱스를 표현
# max 인덱스(2, 9), max 인덱스(4, 9), max 인덱스(6, 1) => [1, 0, 0]
tf.argmax(x, axis=0).eval(session=sess)
```

```
array([1, 0, 0], dtype=int64)
```

```
# 기본 axis는 0이다
```

```
tf.argmax(x).eval(session=sess)
```

```
array([1, 0, 0], dtype=int64)
```

```
# 2번째(1 인덱스) 자원 제거하고 1번째(0 인덱스) 자원에 가장 큰 값의 인덱스를 표현
# max 인덱스(2, 3, 4), max 인덱스(9, 3, 1) => [2, 0]
```

⇒ 가장 큰 값의 인덱스 반환 0, 1, ...

⇒ 가장 작은 값의 인덱스 반환

from sklearn.metrics import confusion_matrix

데이터 사이언스 스쿨

분류문제는 회귀 분석과 달리 다양한 성능평가 기준이 필요하다. 이 절에서는 분류문제에 사용되는 다양한 성능평가 기준에 대해 알아본다. 사이킷런 패키지는 metrics 서브패키지에서 다음처럼 다양한 분류용 성능평가 명령을 제공한다. 분류결과표(Confusion Matrix)는 타겟의 원래 클래스와 모형이 예측한 클래스가 일치하는지는 갯수로 센 결과를 표나 나타낸 것이다. 정답 클래스는 행(row)으로 예측한 클래스는 열(column)로 나타낸다.

<https://datascienceschool.net/03%20machine%20learning/09.04%20%EB%B6%84%EB%A5%98%20%EC%84%B1%EB%8A%A5%ED%8F%89%EA%B0%80.html>

파이썬) 혼동행렬 그리고 정확도, 정밀도, 민감도, f1 score 등 계산하기 (+heatmap, confusion_matrix, classification_report)

혼동행렬은 분류 모델의 성능을 평가하는 지표로서 영어로는 confusion matrix 라고 하며 혼동행렬, 정오행렬, 오분류표 등으로도 불린다. 지도 학습을 통해 모델링한 "분류 모델이 예측한 값"과 레이블되어 있는 "원래의 값" 간의 관계를 표로 나타낸다. 이 표를 통해 해당 모델의 정확도 (accuracy), 정밀도(precision), 민감도(sensitivity), f1 score 등을 파악할 수 있으며, 이 중에서도 특히 정확도를 통해 해당 모델이 정확하게 분

💻 <https://lovelydiary.tistory.com/363>

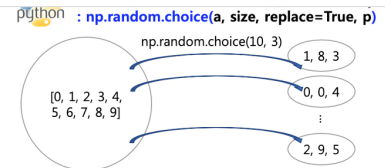
1558	1	82.5999
2188	1	82.5999
1611	1	79.3000
...
1022	1	81.5999

np.random.choice

[Python numpy] np.random.choice() 메소드로 임의표본 추출하기 (무작위, 확률 샘플링)

이번 포스팅에서는 Python numpy 모듈의 random.choice() 메소드를 사용하여 임의(무작위, 확률) 추출 (random sampling) 하는 방법을 소개하겠습니다. numpy.random.choice() 메소드의 기본 Syntax는 아래와 같습니다. 각 parameter별로 예를 들어서 설명을 해보겠습니다. numpy.random.choice(a, size=None, replace=True, p=None) 표

💻 <https://rfriend.tistory.com/548>



numpy.random.choice() 함수 사용하기

numpy.random.choice(a, size=None, replace=True, p=None) a : 1차원 배열 또는 정수 (정수인 경우, np.arange(a) 와 같은 배열 생성) size : 정수 또는 튜플(튜플인 경우, 행렬로 리턴됨. (m, n, k) -> m * n * k), optional

replace : 중복 허용 여부, boolean, optional

p : 1차원 배열, 각 데이터가 선택될 확률, optional

💻 <https://sweepover.tistory.com/11>



```
numpy.random.choice(a, size=None, replace=True, p=None)
```

a : 1차원 배열 또는 정수 (정수인 경우, np.arange(a) 와 같은 배열 생성)

size : 정수 또는 튜플(튜플인 경우, 행렬로 리턴됨. (m, n, k) -> m * n * k), optional

replace : 중복 허용 여부, boolean, optional

p : 1차원 배열, 각 데이터가 선택될 확률, optional

```
numpy.random.choice(5, 3, True)
```

- 0 이상 5 미만인 정수 중 3개를 출력한다. (중복 허용)

```
numpy.random.choice(5, 3, False)
```

- 0 이상 5 미만인 정수 중 3개를 출력한다. (중복 비허용)