Jin-Soo Kim
(jinsoo.kim@snu.ac.kr)

Systems Software &
Architecture Lab.

Seoul National University

Jan. 6 – 17, 2020

Python for Data Analytics

NumPy Lab

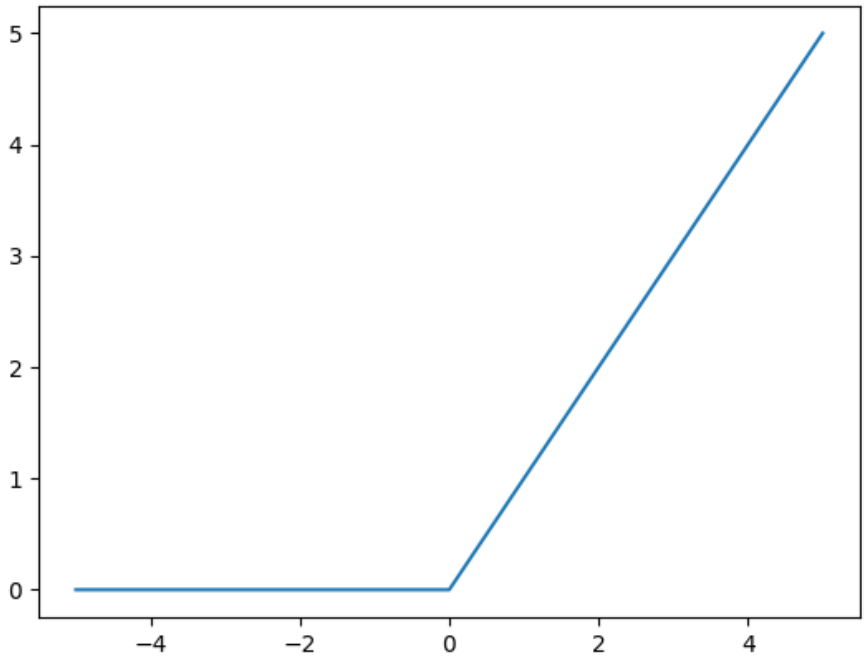
Basic Lab

Lab Numpy-1. Activation Function

- Relu 함수는 최근 가장 많이 사용되는 활성화 함수이다. 함수는 다음과 같이 정의된다.
- $f(x) = \max(0, x)$
- Relu함수는 들어가는 값과, 0 중에 큰 값을 반환하는 함수이다.

Lab Numpy-1. Activation Function

- Relu함수의 개형
- 결과로 출력되는 그래프

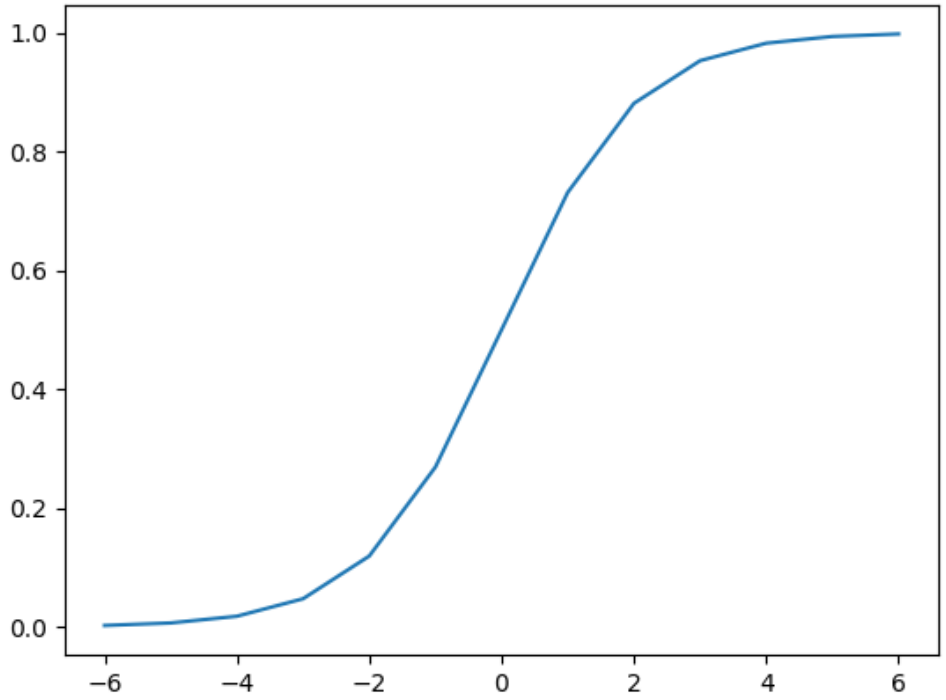


Lab Numpy-1. Activation Function

- Simoid 함수는 활성화 함수이다. 함수는 다음과 같이 정의된다.
- $sigmoid(x) = \frac{1}{1+e^{-x}}$

Lab Numpy-1. Activation Function

- Sigmoid 함수의 개형
- 결과로 출력되는 그래프

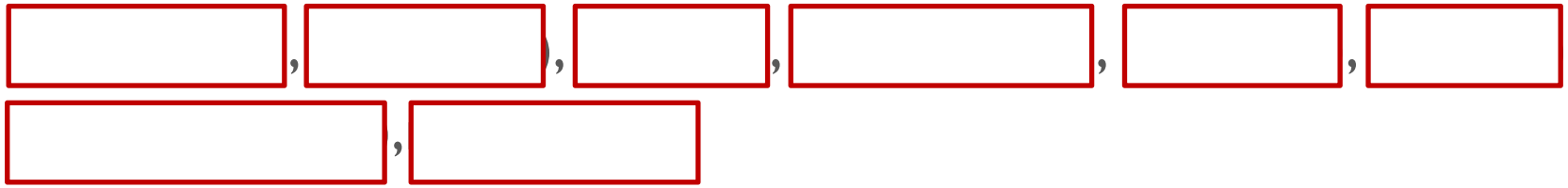


Lab Numpy-2. 카드게임

- 1부터 25까지의 숫자가 그려져 있는 카드가 있다.
- 5명의 사람들에게 5장의 카드를 나눠준다.
- 일정한 규칙을 통해서 합을 얻고, 이 합이 가장 큰 사람이 winner가 된다.

- 합은 아래와 같은 방식으로 구한다.
- 1번째 사람은 1번째 카드에서 나머지 카드의 합을 뺀 값
- 2번째 사람은 2번째 카드에서 나머지 카드의 합을 뺀 값
- 3번째 사람은 3번째 …

Lab Numpy-2. 카드게임

- 예시
  - Hint : 

  - Numpy의 함수들을 잘 이용하면 보다 쉽게 구현할 수 있음, 위의 힌트들은 문제를 풀 때 사용할 수 있는 함수들이다.

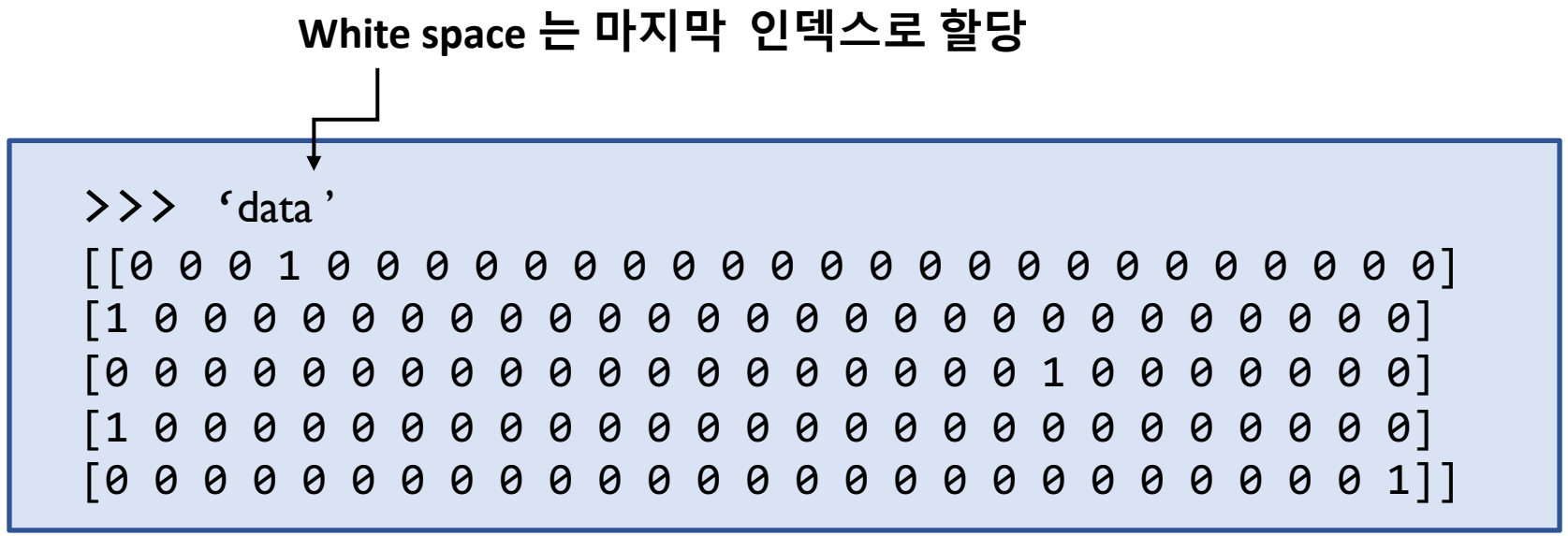
```
1 : [8 18 14 6 3] -> +8 -18 -14 -6 -3 = -33
2 : [ 9 21 25 2 7]  -> -9 +21 -25 -2 -7 = -22
3 : [ 1 19 23 12 16]  -> -1 -19 +23 -12 -16 = -25
4 : [24 13 4 10 22] -> -24 -13 -4 +10 +22 = -53
5 : [20 5 15 17 11] -> -20 -5 -15 -17 + 11 = -46
Winner : 2
```

Lab Numpy-2. 카드게임

- 주의사항
  - For loop 은 쓰지 않고, numpy함수들을 이용해서 구현한다.
  - Matrix 연산을 통해서 구한다.

Lab Numpy-3. one hot encoding

- One hot encoding은 단 하나의 값만 True이고 나머지 값은 모두 False로 encoding하는 것이다.
- 'data'를 one hot encoding하면 아래와 같다.
  - 각각의 alphabet 'a' : 0, 'b' : 1, 'c' : 2, 'd' : 3, …, 'z' : 25 ,' ' : 26 의 index의 값만 True 로 인코딩



Lab Numpy-3. one hot encoding

- Decoding 예시

```
>>> [[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]]

'data '
```

- Numpy를 이용해서 encoding, decoding을 구현해보자.
  - Hint : [ ], [ ], [ ]

Advanced Lab

Lab Numpy-4. 기계학습의 성능 평가 지표

- 기계학습에서 모델이나 패턴의 분류 성능 평가에 사용되는 지표인 Accuracy, Precision, Recall, F-measure(F1 score)를 구현해보기
- lab_numpy_1_Answer.py 에 있는 함수의 EDIT HERE 부분을 구현
- 각 함수에서 넘겨주는 변수 label은 정답이고, pred는 예측값
- Label, pred에서 1은 positive이고, 0은 negative이다.

- numpy를 이용해서 함수를 구현해본다.

Lab Numpy-4. 기계학습의 성능 평가 지표

- Accuracy = $\frac{True\ Nagative + True\ Positive}{Total}$
  - Accuracy(정확도)는 전체에서 실제 positive를 positive라 예측한 것과 실제 negative를 negative라 예측한 것의 비율
  - Hint :

Predicted(pred)

Actual (label)	**Negative(0)**	**Positive(1)**
Negative(0)	True Negative	False Positive
Positive(1)	False Negative	True Positive

Lab Numpy-4. 기계학습의 성능 평가 지표

- Precision = $\frac{True\ Positive}{True\ Positive + False\ Positive}$

  $= \frac{True\ Positive}{Total\ Predicted\ Positive}$

  - Precision은 예측이 positive라고 했을 때, 실제 positive의 비율
  - Hint : [ ] && [ ]

Predicted(pred)

Actual (label)	**Negative(0)**	**Positive(1)**
Negative(0)	True Negative	False Positive
Positive(1)	False Negative	True Positive

Lab Numpy-4. 기계학습의 성능 평가 지표

- Recall $= \frac{True\ Positive}{True\ Positive + False\ Negative}$

$= \frac{True\ Positive}{Total\ Actual\ Positive}$

  - Recall은 실제 positive 중에 모델이 positive라 예측한 것의 비율

Predicted(pred)

Actual (label)	**Negative(0)**	**Positive(1)**
Negative(0)	True Negative	False Positive
Positive(1)	False Negative	True Positive

Lab Numpy-4. 기계학습의 성능 평가 지표

- F_measure = $\frac{2 * precision * recall}{precision + recall}$
  - 앞서 구현한 Precision, Recall의 조화평균
  - 앞서 구현한 precision과 recall을 이용해서 구현하면 편리하게 구현할 수 있음

Predicted(pred)

Actual (label)	**Negative(0)**	**Positive(1)**
Negative(0)	True Negative	False Positive
Positive(1)	False Negative	True Positive