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

Systems Software &
Architecture Lab.

Seoul National University

Jan. 6 – 17, 2020

Python for Data Analytics

Numpy + Pandas Lab

Pandas Lab

Install jupyter notebook

- 이번 lab은 실습환경을 jupyter notebook 으로 편할 수 있습니다.

taxi dataset – load dataset

- download-dataset.py를 실행해서 dataset을 다운받는다.
  - Urllib module이 없을 경우 pip install urllib 실행

- 홈페이지에서 다운

- nyc.2019-01.csv 파일 생성 (대략 700MB)

Lab 1. taxi dataset - trip duration

- Lab 1-1의 DataFrame을 그대로 사용한다.
- df 의 'pickup_datetime', 'dropoff_datetime' 을 사용해서 'trip_duration'(여행시간, 분) 을계산해서 column을 추가하라.

```
>>>print(df.head())
   vendor_id         …    trip_duration
0          1         …         6.666667
1          1         …        19.200000
2          2         …         4.166667
3          2         …         3.333333
```

Lab 2. taxi dataset - trip duration

- Trip duration의 평균, 최소, 최대 시간을 구해보자.
  - 최대, 최소값을 찾아보고, 오류가 있는 데이터를 제거하고 평균, 최소, 최대를 구해보자.

```
>>> min
-84280.5
>>> max
43648.01666666667
```

Lab 3. taxi dataset – pickup day (hint)

- Series containing counts of unique values in Pandas
  - value_counts() function는 시리즈에서 유일한 값들의 개수를 세는데 사용된다.

```
>>> index = pd.Index([2, 2, 5, 3, 4, np.nan])
>>> index.value_counts()
2.0 2
4.0 1
3.0 1
5.0 1
dtype: int64
```

Lab 3. taxi dataset – pickup day

- 주어진 데이터에서 **taxi**를 가장 많이 타는 요일과, 적게 타는 요일을 구해보자.

```
>>> …
Monday          1351516
Tuesday         1259695
…               1203843
…               1082795
…               1007797
…                904512
…                857634
```

Lab 4. taxi dataset – passenger number

- 주어진 데이터에서 **taxi**를 타는 손님의 수의 **count**를 구해보자.

```
>>> …
1    5456121
2    1114106
5     323842
3     314721
6     200811
4     140753
0     117381
8         29
7         19
9          9
```

Lab 5. taxi dataset – pickup_hour

- 주어진 데이터에서 **taxi**를 가장 많이 타는 시간을 구해보자.

```
>>> …
…      514036
…      474371
…      466499
…           …
…           …
…           …
…           …
…           …
…           …
…           …
…           …
```

Lab 6 vendor

- 각 Vendor 별로 fare의 평균을 구해보자.

```
VendorID
... 12.507774
... 12.016892
... 11.563717
Name: fare_amount, dtype: float64
```

- Vendor 별 하루에 번 fare_amount를 구해보자.

```
... 4650987
... 2933685
... 76822
Name: VendorID, dtype: int64
```

Lab 7 location

- Taxi data와 location id(taxi_zone_lookup.csv)에 대한 자료를 이용해서, 승객들이 가장 많이 탑승하는 장소(location id가 아닌 지역이름)는 어디인지 구해보자. (top 10을 구해보자.)

[Location name] 332310
[Location name] 322858
[Location name] 312226
[Location name] 277025
[Location name] 263463

Lab 7 location

- 같은 방법으로 이번에는 가장 많이 하차하는 top 10을 location id가 아닌 지역 이름으로 구해보자.

[Location name] 137273
[Location name] 47709
[Location name] 39903
[Location name] 19460
[Location name] 18327

Numpy Lab

Lab 8. Convolution Layer

- Convolution Neural Networ(CNN)의 핵심 원리인 Convolution Layer을 2D에서 구현해본다.

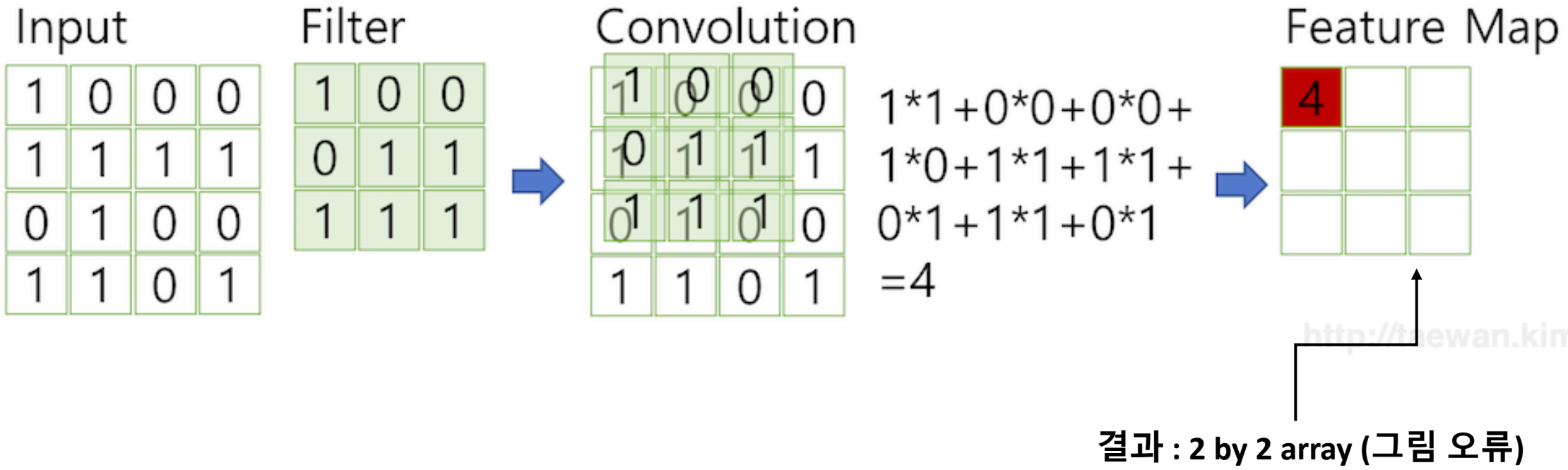
- 입력 데이터를 지정된 간격으로 순회하며 채널별로 합성곱을 하고 모든 채널의 합성곱의 합을 Feature Map로 만든다.

- 필터는 지정된 간격으로 이동하면서 전체 입력데이터와 합성곱하여 Feature Map을 만든다. 다음 장의 그림은 입력 데이터를 (3, 3) 크기의 필터(kernel)로 합성곱하는 과정이다.

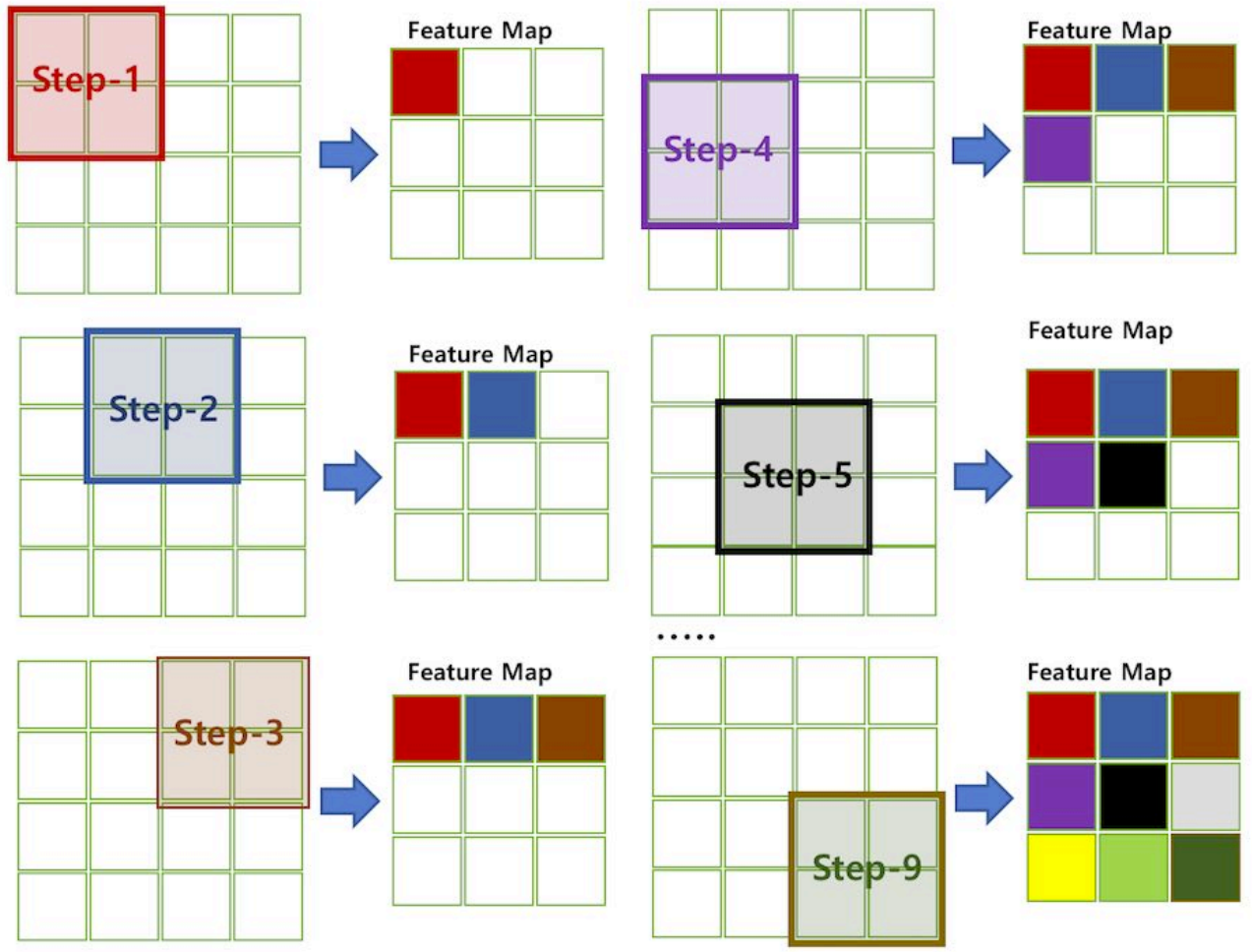
Lab 8. Convolution Layer

- def convolution2d(x, kernel, stride = 1)
  - x 2차원 배열로, input으로 들어가는 값이다.
  - Kernel은 2차원 배열로 합성곱을 할 때 쓰이는 필터이다.
  - Stride는 한번에 이동하는 걸음의 수이다.
    - 예를 들어 stride가 2인 경우에는 filter(or kernel)이 2걸음씩 움직이면서 합성곱을 하여 Feature Map을 만든다.
  - Output size 는 (int((height – kernel_size) / stride) + 1 , int((width – kernel_size)/stride) + 1)

Lab 8. Convolution Layer

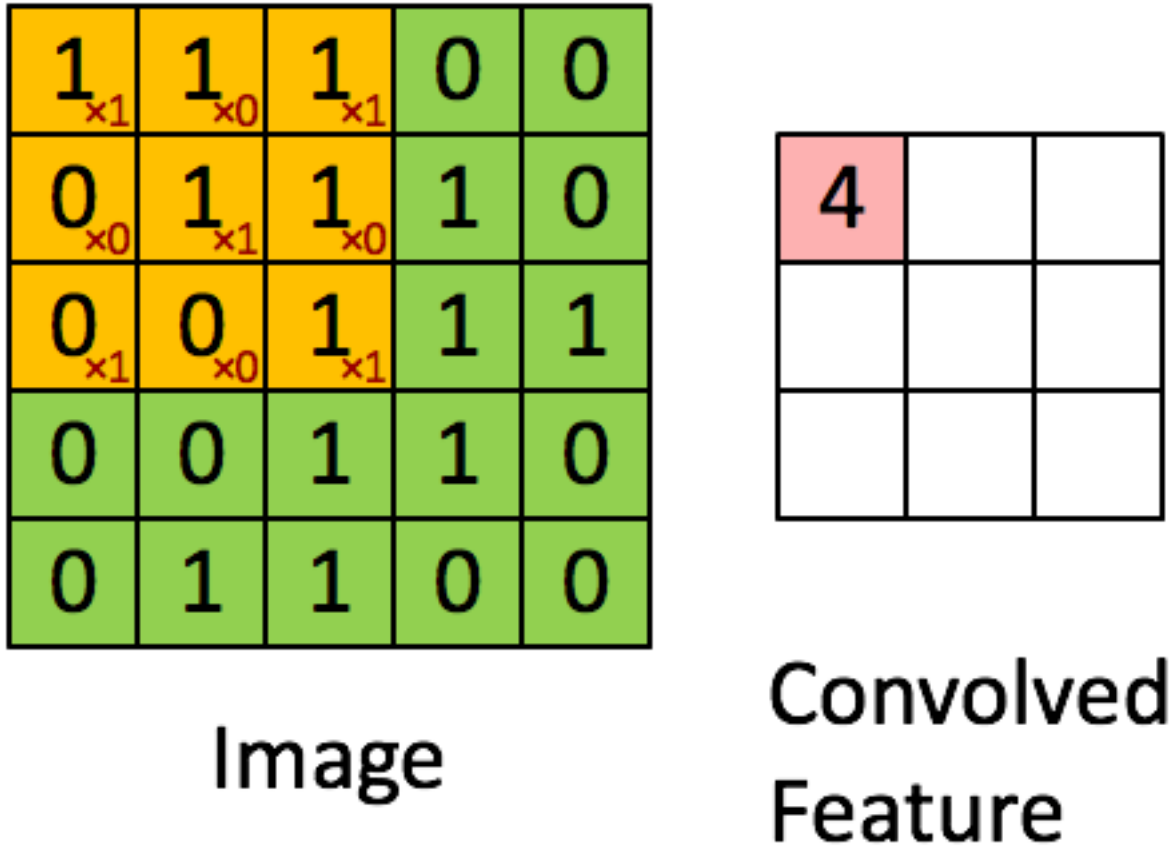


Lab 8. Convolution Layer



Lab 8. Convolution Layer

- 작동 예시



Lab 8. Convolution Layer

- 확인 예시

```
>>> x = np.array([[1,0,0,0], [1,1,1,1], [0,1,0,0], [1,1,0,1]])
>>> kernel = np.array([[1,0,0], [0,1,1], [1,1,1]])
>>> convolution2d(x,kernel)
array([[4., 3.],
       [4., 3.]])
```

Lab 8. Convolution Layer

- 확인 예시

```
>>> x_ = np.array([[1,1,0,1,1], [0,1,0,0,1], [1,1,0,1,0], [0,0,1,1,1], [1,1,1,1,1]])
>>> kernel_ = np.array([[1,0,1], [1,1,1], [0,1,0]])
>>> convolution2d(x,kernel,stride=2)
array([[3. 3.]
 [3. 4.]])
```