Data!

Image-to-Image Translation with Conditional Adversarial Nets

Week 2

Image as Data

So… how…?

- RGB → 3 Channels!

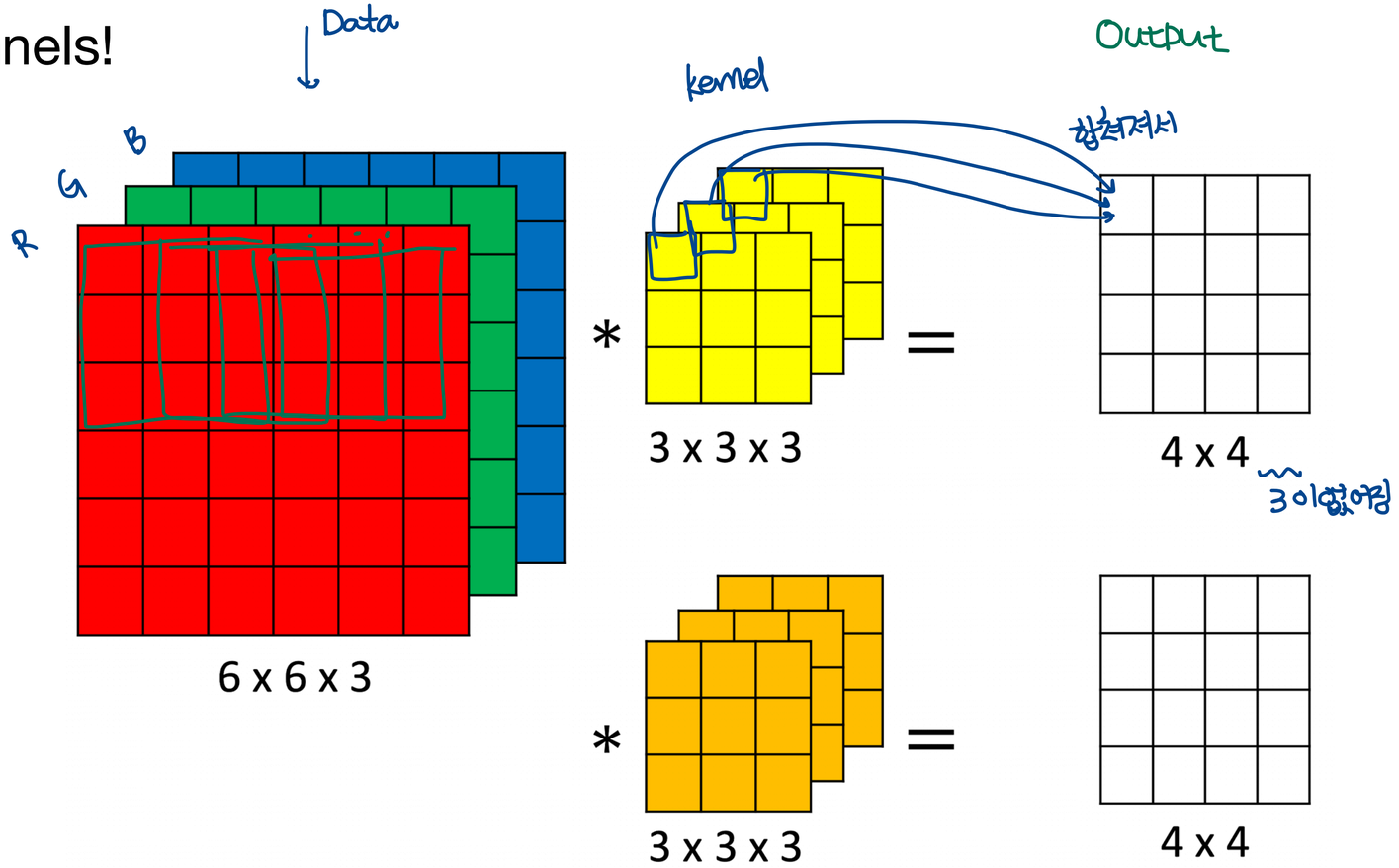


Image as Data

Max Pooling

representation

네트워크들이 깊다보니
기울기 소실 되는 문제가 있어 pooling을 씀

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

9	2
6	3

Global average pooling은 더 많이 씀

Image as Data

Average Pooling

↳ 요즘 많이 안 씀

성능이 좋아져서

의의 ① 압축

② 노이즈를 무시하긴 가겠다

conv. 통과하며 발생하는 노이즈는 무시하긴 가겠다.

Data → conv. → Conv → ..... → output

0.9 → 0.11

0.8 → 0.004

gap ≒ 노이즈

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

15/4	5/4
15/4	8/4

Input Data

~~Average Pooling~~

- Image Files
  - jpg, png, RAW…
- Numpy Files
  - .npy
- h5 Files
  - .h5

특징

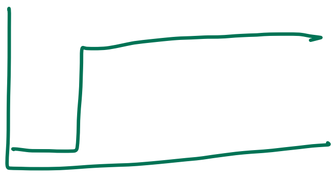
- 컴터 GPU 등 processing power가 좋아야 함 but 요즘 컴터는 ok
- old fashioned way 컴터가 안 따라줘서
- 일반적이지 X
  pytorch 보단 tensorflow → 강점
  DataLoader가 있어서 잘 안씀.

RGB 값 → list

database의 형식 중 하나

GPU utilization

GPU 성능이 안좋으면 delay 있을수있기때문



중간에 delay X

DataLoader

이미지를 deeplearning에 넣을 때 이터레이터를 넘겨주는 친구 → 도우미

What is DataLoader and why do we use it?

- 수만장의 데이터를 직접 for loop 돌면서 넣는다고 생각해 보자.
- GPU에 하나씩 집어넣고… 빼고….
- 한번에 다 부르면 메모리는…?

DataLoader

Parameters of DataLoader

Documentation

Pytorch: 라이브러리

CLASS `torch.utils.data.DataLoader`(*dataset*, *batch_size=1*, *shuffle=None*, *sampler=None*, *batch_sampler=None*, *num_workers=0*, *collate_fn=None*, *pin_memory=False*, *drop_last=False*, *timeout=0*, *worker_init_fn=None*, *multiprocessing_context=None*, *generator=None*, ***, *prefetch_factor=2*, *persistent_workers=False*, *pin_memory_device=''*) [SOURCE]

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `DataLoader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See `torch.utils.data` documentation page for more details.

DataLoader

Parameters of DataLoader

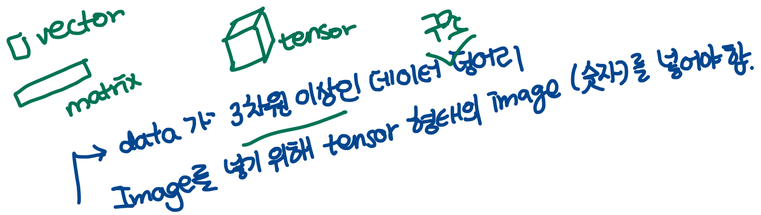
(parameter) image (tensor 형태의)

```
CLASS torch.utils.data.DataLoader(dataset, batch_size=1, shuffle=None, sampler=None,
      batch_sampler=None, num_workers=0, collate_fn=None, pin_memory=False, drop_last=False,
      timeout=0, worker_init_fn=None, multiprocessing_context=None, generator=None, *,
      prefetch_factor=2, persistent_workers=False, pin_memory_device='')  [SOURCE]
```

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `DataLoader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See `torch.utils.data` documentation page for more details.



- dataset
  - 실제 넣을 데이터! → tensor

DataLoader

Parameters of DataLoader

```
CLASS  torch.utils.data.DataLoader(dataset, batch_size=1, shuffle=None, sampler=None,
       batch_sampler=None, num_workers=0, collate_fn=None, pin_memory=False, drop_last=False,
       timeout=0, worker_init_fn=None, multiprocessing_context=None, generator=None, *,
       prefetch_factor=2, persistent_workers=False, pin_memory_device='')  [SOURCE]
```

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `DataLoader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See `torch.utils.data` documentation page for more details.

- batch_size
  - 배치 크기를 정해준다! GPU에 한번에 몇 개의 이미지를 넣어 줄 것이냐!

DataLoader

Parameters of DataLoader

순서에 따른 bias 방지 random으로 batch를 가져오게함.

```
CLASS  torch.utils.data.DataLoader(dataset, batch_size=1, shuffle=None, sampler=None,
       batch_sampler=None, num_workers=0, collate_fn=None, pin_memory=False, drop_last=False,
       timeout=0, worker_init_fn=None, multiprocessing_context=None, generator=None, *,
       prefetch_factor=2, persistent_workers=False, pin_memory_device='')  [SOURCE]
```

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `DataLoader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See `torch.utils.data` documentation page for more details.

- shuffle

  99%의 확률로 일반적으로 해야 함

  - True일 시, 매 epoch마다 데이터가 임의로 섞인다!

DataLoader

Parameters of DataLoader

```
CLASS  torch.utils.data.DataLoader(dataset, batch_size=1, shuffle=None, sampler=None,
       batch_sampler=None, num_workers=0, collate_fn=None, pin_memory=False, drop_last=False,
       timeout=0, worker_init_fn=None, multiprocessing_context=None, generator=None, *,
       prefetch_factor=2, persistent_workers=False, pin_memory_device='')  [SOURCE]
```

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `DataLoader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See `torch.utils.data` documentation page for more details.

- num_workers
  - 멀티 프로세싱! 보통은 “GPU 개수 X 2” 혹은 “GPU 개수 X 4”

DataLoader

Parameters of DataLoader

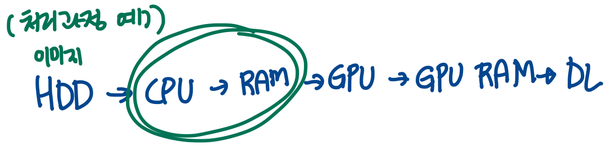
```
CLASS  torch.utils.data.DataLoader(dataset, batch_size=1, shuffle=None, sampler=None,
       batch_sampler=None, num_workers=0, collate_fn=None, pin_memory=False, drop_last=False,
       timeout=0, worker_init_fn=None, multiprocessing_context=None, generator=None, *,
       prefetch_factor=2, persistent_workers=False, pin_memory_device='')  [SOURCE]
```

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `DataLoader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See `torch.utils.data` documentation page for more details.

- pin_memory (기본 FALSE) :: 식당 예약제



  - 예약해놓고 씁시다! 빠르게 빠르게 갑시다!

DataLoader

Parameters of DataLoader

CLASS torch.utils.data.DataLoader(*dataset*, *batch_size=1*, *shuffle=None*, *sampler=None*, *batch_sampler=None*, *num_workers=0*, *collate_fn=None*, *pin_memory=False*, *drop_last=False*, *timeout=0*, *worker_init_fn=None*, *multiprocessing_context=None*, *generator=None*, ***, *prefetch_factor=2*, *persistent_workers=False*, *pin_memory_device=''*) [SOURCE]

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `DataLoader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See `torch.utils.data` documentation page for more details.

- drop_last
  - 배치로 나누다 남는 녀석들은…?

0 1 2 3 4 5 6 7 8 9

batch_size = 3

data가 많으면 : True

but 웬만하면 False

DataLoader

Parameters of DataLoader

CLASS `torch.utils.data.DataLoader(`*`dataset`*`, `*`batch_size=1`*`, `*`shuffle=None`*`, `*`sampler=None`*`, `*`batch_sampler=None`*`, `*`num_workers=0`*`, `*`collate_fn=None`*`, `*`pin_memory=False`*`, `*`drop_last=False`*`, `*`timeout=0`*`, `*`worker_init_fn=None`*`, `*`multiprocessing_context=None`*`, `*`generator=None`*`, *, `*`prefetch_factor=2`*`, `*`persistent_workers=False`*`, `*`pin_memory_device=''`*`)` [SOURCE]

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `DataLoader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See `torch.utils.data` documentation page for more details.

- drop_last
  - 배치로 나누다 남는 녀석들은…?