

```
In [1]: # Google Colab에서 노트북을 실행하실 때에는
# https://tutorials.pytorch.kr/beginner/colab 를 참고하세요.
%matplotlib inline
```

파이토치(PyTorch) 기본 익히기 || 빠른 시작 || 텐서(Tensor) || Dataset과 Dataloader || 변형(Transform) || 신경망 모델 구성하기 || Autograd || 최적화(Optimization) || 모델 저장하고 불러오기

## 변형(Transform)

데이터가 항상 머신러닝 알고리즘 학습에 필요한 최종 처리가 된 형태로 제공되지는 않습니다. **변형(transform)** 을 해서 데이터를 조작하고 학습에 적합하게 만듭니다.

모든 torchvision 데이터셋들은 변형 로직을 갖는, 호출 가능한 객체(callable)를 받는 매개변수 두 개 ( 특징(feature)을 변경하기 위한 `transform` 과 정답(label)을 변경하기 위한 `target_transform` )를 갖습니다 `torchvision.transforms` 모듈은 주로 사용하는 몇가지 변형(transform)을 제공합니다.

FashionMNIST 특징(feature)은 PIL Image 형식이며, 정답(label)은 정수(integer)입니다. 학습을 하려면 정규화(normalize)된 텐서 형태의 특징(feature)과 원-핫(one-hot)으로 부호화(encode)된 텐서 형태의 정답(label)이 필요합니다. 이러한 변형(transformation)을 하기 위해 `ToTensor` 와 `Lambda` 를 사용합니다.

```
In [2]: import torch
from torchvision import datasets
from torchvision.transforms import ToTensor, Lambda

ds = datasets.FashionMNIST(
    root="data",
    train=True,
    download=True,
    transform=ToTensor(), #label 값은 0-9까지 숫자
    target_transform=Lambda(lambda y: torch.zeros(10, dtype=torch.float).
)

# 텐서에 0으로 가득찬 텐서 제작 -> 텐서에 해당되는 값만 1로 고정 -> onehot encoding
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz to data/FashionMNIST/raw/train-images-idx3-ubyte.gz
```

```
100%|██████████| 26421880/26421880 [00:01<00:00, 16975066.17it/s]
```

```
Extracting data/FashionMNIST/raw/train-images-idx3-ubyte.gz to data/FashionMNIST/raw
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz to data/FashionMNIST/raw/train-labels-idx1-ubyte.gz
```

```
100%|██████████| 29515/29515 [00:00<00:00, 330584.27it/s]
```

Extracting data/FashionMNIST/raw/train-labels-idx1-ubyte.gz to data/FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz to data/FashionMNIST/raw/t10k-images-idx3-ubyte.gz

100%|██████████| 4422102/4422102 [00:00<00:00, 6141921.98it/s]

Extracting data/FashionMNIST/raw/t10k-images-idx3-ubyte.gz to data/FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz to data/FashionMNIST/raw/t10k-labels-idx1-ubyte.gz

100%|██████████| 5148/5148 [00:00<00:00, 6158664.29it/s]

Extracting data/FashionMNIST/raw/t10k-labels-idx1-ubyte.gz to data/FashionMNIST/raw

## ToTensor()

`ToTensor` 는 PIL Image나 NumPy `ndarray` 를 `FloatTensor` 로 변환하고, 이미지의 픽셀의 크기(intensity) 값을 `[0., 1.]` 범위로 비례하여 조정(scale)합니다.

```
In [5]: torch.tensor([1,2,3,4]).float() # 이곳 변형 # tensor 형태로 가공 변형
```

```
Out[5]: tensor([1., 2., 3., 4.])
```

## Lambda 변형(Transform)

Lambda 변형은 사용자 정의 람다(lambda) 함수를 적용합니다. 여기에서는 정수를 원-핫으로 부호화된 텐서로 바꾸는 함수를 정의합니다. 이 함수는 먼저 (데이터셋 정답의 개수인) 크기 10짜리 영 텐서(zero tensor)를 만들고, `scatter_` 를 호출하여 주어진 정답 `y` 에 해당하는 인덱스에 `value=1` 을 할당합니다.

```
In [ ]:
```

```
In [6]: target_transform = Lambda(lambda y: torch.zeros(
    10, dtype=torch.float).scatter_(dim=0, index=torch.tensor(y), value=1
```

```
In [8]: for i in range(1,8): # 이곳 변형
    print(i,target_transform(i))
```

```
1 tensor([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.])
2 tensor([0., 0., 1., 0., 0., 0., 0., 0., 0., 0.])
3 tensor([0., 0., 0., 1., 0., 0., 0., 0., 0., 0.])
4 tensor([0., 0., 0., 0., 1., 0., 0., 0., 0., 0.])
5 tensor([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.])
6 tensor([0., 0., 0., 0., 0., 0., 1., 0., 0., 0.])
7 tensor([0., 0., 0., 0., 0., 0., 0., 1., 0., 0.])
```

## 이것을 해주는 이유 :

- 로스값을 계산해줄때 loss 함수를 변형해주어야 한다.
  - loss 계산 -> softmax값으로 나올때, onehot encoding형태일때 계산이 편리하다
- 

## 더 알아보기

- [torchvision.transforms API](#)