

Type something...

01) Tensor manipulation

01_Pytorch learning

01) Tensor manipulation

01_Pytorch learning

View (Reshape)

```
t = np.array([[[0, 1, 2],
               [3, 4, 5]],
              [[6, 7, 8],
               [9, 10, 11]]])
ft = torch.FloatTensor(t)
print(ft.shape)

torch.Size([2, 2, 3])

print(ft.view([-1, 3]))
print(ft.view([-1, 3]).shape)

tensor([[ 0.,  1.,  2.],
        [ 3.,  4.,  5.],
        [ 6.,  7.,  8.],
        [ 9., 10., 11.]])
torch.Size([4, 3])

print(ft.view([-1, 1, 3]))
print(ft.view([-1, 1, 3]).shape)

tensor([[[ 0.,  1.,  2.],
          [ 3.,  4.,  5.],
          [ 6.,  7.,  8.],
          [ 9., 10., 11.]])
torch.Size([4, 1, 3])
```

- `ft.view([-1, 3])` column 개수를 3개로 고정하고 row개수를 남은 개수로 조정해달라는 의미이다. 정말 많이 사용하는 method로 잘 익혀주자.

Squeeze

```
ft = torch.FloatTensor([[[0], [1], [2]])
print(ft)
print(ft.shape)

tensor([[0.],
        [1.],
        [2.]])
torch.Size([3, 1])

print(ft.squeeze())
print(ft.squeeze().shape)

tensor([0., 1., 2.])
torch.Size([3])
```

- `ft.squeeze()` 는 torch를 불러준다 생각해서 2차원 → 1차원으로 줄여주는 method이다.

Unsqueeze

```
ft = torch.Tensor([0, 1, 2])
print(ft.shape)

torch.Size([3])

print(ft.unsqueeze(0))
print(ft.unsqueeze(0).shape)

tensor([[0., 1., 2.]])
torch.Size([1, 3])

print(ft.view(1, -1))
print(ft.view(1, -1).shape)

tensor([[0., 1., 2.]])
torch.Size([1, 3])

print(ft.unsqueeze(1))
print(ft.unsqueeze(1).shape)

tensor([[0.],
        [1.],
        [2.]])
torch.Size([3, 1])

print(ft.unsqueeze(-1))
print(ft.unsqueeze(-1).shape)

tensor([[0.],
        [1.],
        [2.]])
torch.Size([3, 1])
```

- 반대로 `ft.unsqueeze()` 는 torch를 팽창한다고 생각해서 1차원 → 2차원으로 늘려주는 method이다.
- `ft.unsqueeze(0)` 을 하게 되면 column 방향으로 팽창하고 `ft.unsqueeze(1)` 을 하게 되면 row 방향으로 팽창한다.