torch.nn.LayerNorm

```
CLASS torch.nn.LayerNorm(normalized_shape : Union[int, List[int],torch.Size],
                          eps : float = le-0.5, elementwise_affine : bool = True)
```

Layer Normalization over a mini-batch of inputs as described in the paper Layer Normalization.

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} * \gamma + \beta$$

Unlike Batch Normalization and Instance Normalization, which applies scalar scale and bias for each entire channel/plane with the affine option, Layer Normalization applies per-element scale and bias with elementwise_affine.

파라미터

- normalized_shape (int or list or torch.Size) :
  input shape from an expected input of size
  if a single is used, it is treated as a singleton list, and this module will normalize over the last dimension which is expected to be of that specific size.
- eps :
  a value added to the denominator fordenominator for numerical stability.
  Default: 1e-5
- elementwise_affine :
  a boolean value that when set to True, this module has learnable per-element affine parameters initialized to ones (for weights) and zeros (for biases).
  Default: True

Examples

```
input = torch.randn(20, 5, 10, 10)

# with Learnable Parameters
m = nn.LayerNorm(input.size()[1:]) #torch.Size([5, 10, 10])

# without Learnable Parameters
m = nn.LayerNorm(input.size()[1:],elementwise_affine=False)

# Normalize over last two dimensions
m = nn.LayerNorm(input.size()[-2:])

# Normalize over last dimension of the size 10
m = nn.LayerNorm(input.size()[-1])

# activating the module
output = m(input)
```

출처

https://pytorch.org/docs/master/generated/torch.nn.LayerNorm.html