**.weight**

获取网络层权重

embedding = nn.Embedding(10, 3)

print(embedding.weight) *#.weight得到参数*

print(type(embedding.weight))

print(embedding.weight.data)*#.weight.data得到参数对应的张量，对于需要求梯度的参数，不可以直接用.uniform\_改变其值，只能用.data.uniform\_改变*

print(type(embedding.weight.data))

embedding.weight.data.uniform\_(-0.1,0.1)

>>>

Parameter containing:

tensor([[ 0.4048, 1.1188, 0.2894],

[ 0.9620, -0.4265, 0.1693],

[-1.0076, -1.3301, 0.3185],

[ 0.3934, -1.3923, -1.2337],

[-0.7122, -0.1948, 0.2499],

[-0.0374, -1.0542, 0.3273],

[-0.7436, 1.3305, 0.4908],

[ 1.8960, -1.4166, 0.7586],

[ 0.7181, -0.2122, 0.2968],

[-0.6597, -0.2664, -0.1384]], requires\_grad=True)

<class 'torch.nn.parameter.Parameter'>

tensor([[ 0.4048, 1.1188, 0.2894],

[ 0.9620, -0.4265, 0.1693],

[-1.0076, -1.3301, 0.3185],

[ 0.3934, -1.3923, -1.2337],

[-0.7122, -0.1948, 0.2499],

[-0.0374, -1.0542, 0.3273],

[-0.7436, 1.3305, 0.4908],

[ 1.8960, -1.4166, 0.7586],

[ 0.7181, -0.2122, 0.2968],

[-0.6597, -0.2664, -0.1384]])

<class 'torch.Tensor'>