

# BCELoss和BCEWithLogitsLoss

## 1.BCELoss

如果3张图片分3类，会输出一个3\*3的矩阵。

```
[3]: input=torch.randn(3,3)
      input
```

```
[3]: tensor([[ -0.4089, -1.2471,  0.5907],
             [-0.4897, -0.8267, -0.7349],
             [ 0.5241, -0.1246, -0.4751]])
```

[https://blog.csdn.net/qq\\_22210253](https://blog.csdn.net/qq_22210253)

先用Sigmoid给这些值都搞到0~1之间：

```
[4]: m=nn.Sigmoid()
```

```
[5]: m(input)
```

```
[5]: tensor([[0.3992, 0.2232, 0.6435],
             [0.3800, 0.3044, 0.3241],
             [0.6281, 0.4689, 0.3834]])
```

[https://blog.csdn.net/qq\\_22210253](https://blog.csdn.net/qq_22210253)

假设Target是：

```
[30]: target=torch.FloatTensor([[0,1,1],
                                [0,0,1],
                                [1,0,1]])
```

BCELoss是 $-\frac{1}{n} \sum (y_n \times \ln x_n + (1 - y_n) \times \ln(1 - x_n))$

其中y是target，x是模型输出的值。

所以对于第一行：

第一列 $0 \times \ln 0.3992 + (1 - 0) \times \ln(1 - 0.3992) = -0.5095$

第二列 $1 \times \ln 0.2232 + (1 - 1) \times \ln(1 - 0.2232) = -1.4997$

第三列 $1 \times \ln 0.6435 + (1 - 1) \times \ln(1 - 0.6435) = -0.4408$

第二行：

第一列 $0 \times \ln 0.3800 + (1 - 0) \times \ln(1 - 0.3800) = -0.4780$

第二列 $0 \times \ln 0.3044 + (1 - 0) \times \ln(1 - 0.3044) = -0.3630$

第三列 $1 \times \ln 0.3241 + (1 - 1) \times \ln(1 - 0.3241) = -1.1267$

第三行：

第一列 $1 \times \ln 0.6281 + (1 - 1) \times \ln(1 - 0.6281) = -0.4651$

第二列 $0 \times \ln 0.4689 + (1 - 0) \times \ln(1 - 0.4689) = -0.6328$

第三列 $1 \times \ln 0.3834 + (1 - 1) \times \ln(1 - 0.3834) = -0.9587$

去掉负号求个均值：

$\frac{0.5095+1.4997+0.4408}{3} = 0.8167$

$\frac{0.4780+0.3630+1.1267}{3} = 0.6559$

$\frac{0.4651+0.6328+0.9587}{3} = 0.6855$

再取个平均：

$\frac{0.8167+0.6559+0.6855}{3} = 0.7194$

下面我们用BCELoss来验证一下Loss是不是0.7194！

```
[25]: loss=nn.BCELoss()
```

```
[31]: loss(input,target)
```

```
[31]: tensor(0.7193)
```

[https://blog.csdn.net/qq\\_22210253](https://blog.csdn.net/qq_22210253)

## BCEWithLogitsLoss

BCEWithLogitsLoss就是把Sigmoid-BCELoss合成一步。我们直接用刚刚的input验证一下是不是0.7193：

```
[46]: input=torch.FloatTensor([[-0.4089, -1.2471, 0.5907],
                                [-0.4897, -0.8267, -0.7349],
                                [ 0.5241, -0.1246, -0.4751]])
```

```
[47]: target=torch.FloatTensor([[0,1,1],
                                [0,0,1],
                                [1,0,1]])
```

```
[48]: loss=nn.BCEWithLogitsLoss()
```

```
[49]: loss(input,target)
```

```
[49]: tensor(0.7193)
```

[https://blog.csdn.net/qq\\_22210253](https://blog.csdn.net/qq_22210253)

