

# Very Deep Learning

## Exercise Sheet 5 Task 9

1. Can Cross Entropy Loss be used instead of BCELoss?

If no, why not? If yes, how?

Yes it is possible because BCELoss (binary classification) is just a special case of Cross Entropy Loss (multiclass classification). First, we have to compute the onehot encoded vectors, then we can use Cross Entropy Loss. Since, we only have two classes, we can use Cross Entropy Loss directly and without any further adjustments.

2. Is NLLLoss() and last layer as LogSoftmax() is same as using CrossEntropyLoss() with a Softmax final layer? Can you get the mathematical intuition behind it?

NLLLoss() is the negative log likelihood loss. The input includes the log probabilities of each class, weights are optional. The output is not logarithmized, therefore the LogSoftmax layer is added afterwards. LogSoftmax only changes the value, hence input and output are from the same shape.

The CrossEntropyLoss() is a combination of NLLLoss() and LogSoftmax(). Like the NLLLoss() class it expect the scores for each class as an input and weights are optional. Then it does the same like NLLLoss(). However the output is already logarithmized in comparison to NLLLoss(), hence LogSoftmax() is not needed when using CrossEntropyLoss().

$$\text{NLLLoss: } \text{loss}(x, y) = \sum_{n=1}^N \frac{1}{w_{y_n}} l_n$$

$$\text{LogSoftmax}(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right)$$

$$\text{CrossEntropyLoss: } \text{loss}(x, y) = -\log\left(\frac{\exp(x[y])}{\sum_j \exp(x[j])}\right),$$

$$\text{or with weight } \text{weight}[class](-\log\left(\frac{\exp(x[y])}{\sum_j \exp(x[j])}\right))$$

$y = \text{class}$  ,  $N = \text{batch size}$

The only difference between the two options is that with NLLLoss() the extra layer LogSoftmax() is needed, while CrossEntropyLoss does it in one layer.

Source (Task 2):

<https://pytorch.org/docs/stable/nn.html?highlight=crossentropy#torch.nn.CrossEntropyLoss>