Logits and BCE with Logits Loss - Explained

February 17, 2025

1 Logits, BCE Loss, and BCE with Logits Loss

Binary classification is a fundamental task in machine learning, often using the Binary Cross-Entropy (BCE) loss function combined with a Sigmoid activation. However, an alternative, BCEWithLogitsLoss, provides better numerical stability and is preferred in many cases.

To make this topic easier to understand, I will explain it in two ways:

1. **A simple, non-technical manner**
2. **A technical and mathematically reasoned approach**

1.1 Logits and BCE Loss: Simple Explanation

1.1.1 What are Logits?

Logits are the raw outputs of a neural network before applying an activation function like Sigmoid or Softmax. Unlike probabilities (which range from 0 to 1), logits can take any real value $(-\infty,+\infty)$.

1.1.2 How Logits Relate to BCE Loss?

We normally have multiple parameters inside an NN architecture with multiple different types of layers (e.g., weights & bias) connceted to one another. While we train our models to learn to adjust the parameters, we send the parameters with a forward pass function where we use the activation function like sigmoid. The activation function is nothing but a normalizing function that represents a raw linear combination of inputs and learned weights (**logits**) into a probabilistic range (0 to 1).

So, when we use the BCE loss function, we technically apply the sigmoid function first to the final layer before calculating the loss in backpropagation. This means, when using BCE loss for a binary classification task, we must include a sigmoid activation function in the final layer to represent the logits into a range of probability.

1.1.3 Why do we apply Sigmoid?

But what will happen if we do not use an activation function? Well, the model's network could output values outside the normal range [0, 1] of probability and lead to negative logs while calculating loss while causing instability with gradients.

1.1.4 Why use Logits?

To simply put, logits are more expressive than a restricted range of probability [0, 1]. Logits provide an unrestricted range $(-\infty,+\infty)$, meaning a more expressive range of a parameter's learning pattern

rather than a restricted range of (0, 1). Eventually, this avoids saturation later when we use sigmoid activation.

1.1.5 How are Logits useful?

We already covered the first usefulness of logits above; now let's see how it helps future sigmoid activation later on.

If we apply sigmoid activation before training (what we do in BCE Loss), we might face gradient vanishing issues (for more details on vanishing gradients, click here) when the network produces very high or very low values. For example:

If the logits range is very large, let's say (-100, 100), then the sigmoid function saturates to 0 or 1, leading to a near-zero gradient. Which by all means is a bad thing for our model. This leads to the computational instability with gradients. Because computing BCE loss directly with probabilities can lead to an undefined value or undefined results beyond the probability range. So, we could say logits are important for ***three things*** *mainly:*

- Logits are more expressive.
- Logits help avoid vanishing gradients.
- Logits are computationally stable.