3. BCE is a loss function that computes a distance between an output distribution and a ground truth distribution for optimization. We backpropogate over this loss to train our neural network to output distributions close to the ground truth distribution. The function for BCE is as follows, with x_i being the input data, $f(x_i)$ being the output & y_i being the desired probability,

$$BCE(\vec{x}, \vec{y}) = -\sum_{i=1}^{n} \left(y_i \log(f(x_i)) + (1-y_i)\log(1-f(x_i))\right)$$

Given $o_{min} = \text{argmin}_o\, BCE(o, g)$, the best approximation to g our network $f: \mathbb{R}^N \to [0,1]$ can produce, we need to find g_{max} s.t.

$$g_{max} = \underset{g}{\text{argmax}}\; BCE(o_{min}, g)$$

We first find O_{min}. For i^{th} term in o_{min} & g, we maximize the expression

$$I = g_i \log(o_{min,i}) + (1-g_i)\log(1-o_{min,i})$$

We do this by differentiating

$$\frac{d}{do_{min,i}} I = \frac{g_i}{o_{min,i}} - \frac{1-g_i}{1-o_{min,i}}$$, and we find maxima at $\frac{d}{do_{min,i}} = 0$

$$\Rightarrow \underline{\underline{O_{min,i} = g_i}} \Rightarrow O_{min} = g$$

For finding g_{max},

$\Rightarrow$ For every term in $o_{min,i}\log(g_{max,i}) + (1-O_{min})\log(1-g_{max,i})$, this

This term needs to be minimized. It is easy to see this happens when

$$g_{max,i} = \begin{cases} 1 & o_{min} < 0.5 \\ 0 & o_{min} \geq 0.5 \end{cases}$$

It's intuitively the furthest distribution from o_{min} as well

$$\therefore g_{max,i} = \begin{cases} 1 & g < 0.5 \\ 0 & g \geq 0.5 \end{cases}$$