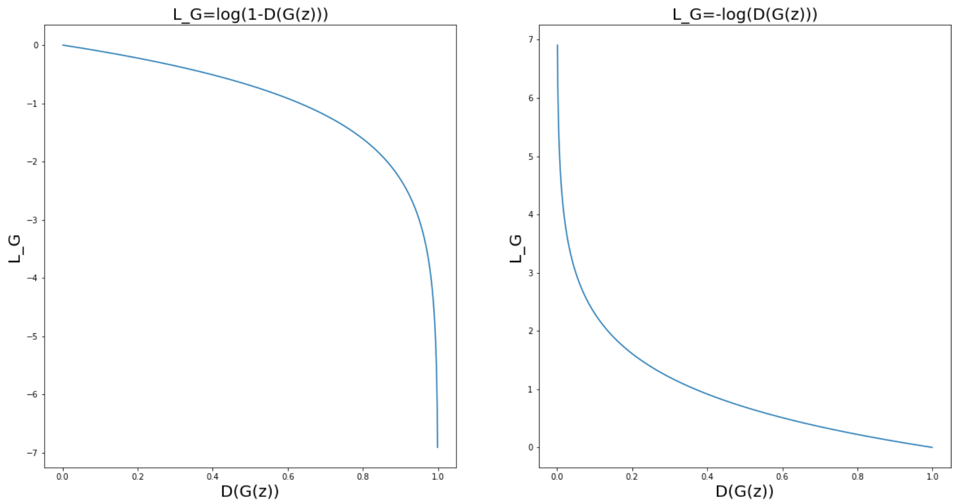
Reference Answer to the Comparison of Two Loss Functions for Generator in Vanilla GANs



Above is the curve of the two loss functions:

$L_G = \log(1 - D(G(z)))$ and $L_G = -\log D(G(z))$, w.r.t. $\mathrm{D(G(z))}$.

As described in the original GAN paper "Generative Adversarial Networks":

> In practice, equation 1 may not provide sufficient gradient for G to learn well. Early in learning, when G is poor, D can reject samples with high confidence because they are clearly different from the training data. In this case, $\log(1 - D(G(\boldsymbol{z})))$ saturates. Rather than training G to minimize $\log(1 - D(G(\boldsymbol{z})))$ we can train G to maximize $\log D(G(\boldsymbol{z}))$. This objective function results in the same fixed point of the dynamics of G and D but provides much stronger gradients early in learning.

The loss function $L_G = \log(1 - D(G(z)))$ saturates when G is poor in the early learning while another loss function $L_G = -\log D(G(z))$ does not.

We further explain it a bit.

Since the activation function of D right before its output is a sigmoid function, the output of D, including $\mathrm{D(G(z))}$, is always in $(0, 1)$. As shown in the above curve, both two loss functions are monotonically decreasing over $(0, 1)$ hence both ideally being equal optimization objectives.

However, they are different in gradient calculations.

Assuming a weight vector w_G which is in the generator G and takes part in the generation of $\mathrm{G(z)}$, we calculate the gradient of L_G w.r.t w_G by the chain rule:

$$\nabla_{w_G} L_G = \left(\frac{\partial L_G}{\partial \mathrm{D(G(z))}} \frac{\partial \mathrm{D(G(z))}}{\partial \mathrm{w}_G} \right)^T$$

Because of the chain rule, there is always a term $\frac{\partial L_G}{\partial \mathrm{D(G(z))}}$, which is the derivative of the loss function L_G w.r.t its input $\mathrm{D(G(z))}$, in the equation. This derivative can be visualized easily since it is exactly the slope of the above curve. The gradient $\nabla_{w_G} L_G$ becomes smaller as $\frac{\partial L_G}{\partial \mathrm{D(G(z))}}$ gets smaller.

By the fact that the discriminator is usually easier to train and overpowers the generator soon in the training progress (while the generator may still producing only noise-like images), $\mathrm{D(G(z))}$ becomes a small value in the early epochs.

According to the curve, $L_G = -\log D\big(G(z)\big)$ is steeper than $L_G = \log(1 - D(G(z)))$ when $\mathrm{D(G(z))}$ is small thus providing more adequate gradient for the generator to get updated in the early epochs.

Please note that this answer is provided as a reference, which means that we also accept other answers from an even completely different perspective. In lab4, we gave full marks on this question as long as the **comparison** of these two loss functions are shown along with an explanation.