- 与生成器的收敛度和样本质量相关的有意义的损失度量。
- 优化过程的稳定性提高。

具体来说，这篇论文针对判别器和生成器提出了一种新的损失函数。用这种损失函数代替二值交叉熵就可以让 GAN 的收敛更加稳定。从数学的角度解释这个问题超出了本书的范围，网上有很多解释该损失函数基本原理的资源。

下面让我们看看 Wasserstein 损失函数的定义。

4.5.1 Wasserstein 损失

首先我们来复习一下二值交叉熵，目前我们在训练 GAN 判别器和生成器时就采用了这种损失函数：

二值交叉熵

$$-\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log\left(p_i\right) + \left(1-y_i\right)\log\left(1-p_i\right)\right)$$

为了训练 GAN 的判别器 D，我们根据以下两者计算损失：真实图像的预测 $p_i=D(x_i)$ 与响应 $y_i=1$ 之间的比较，以及生成图像的预测 $p_i=D(G(z_i))$ 与响应 $y_i=0$ 之间的比较。因此，对于 GAN 的判别器来说，将损失函数最小化的过程可以表示为：

GAN 判别器的损失最小化

$$\min_D -\left(\mathbb{E}_{x\sim p_X}[\log D(x)] + \mathbb{E}_{z\sim p_Z}[\log(1-D(G(z)))]\right)$$

为了训练 GAN 的生成器 G，我们根据生成图像的预测 $p_i=D(G(z_i))$ 与响应 $y_i=1$ 的比较计算损失。因此，对于 GAN 的生成器来说，将损失函数最小化的过程可以表示为：