声明：$\tilde{p}(x)$真实的样本分布，其具体表达式是不知道的。$p(x)$对样本分布的近似，这是我们想求出的。

引入：

在 simclr 中，我们使用了 softmax 函数来判断正样本对的概率，并希望最大化该值。

指数族分布：

在很多问题中都会出现指数族分布，即对于某个变量x的概率$p(x)$，有

$$P(x) = \frac{e^{G(x)}}{Z}$$

其中$G(x)$是 x 的某个"能量"函数，而$Z = \sum_x e^{G(x)}$是归一化函数，也称配分函数。

关于配分函数的难点：

1. 计算量太大：

在进行多分类任务时，假如类别过多，这就需要对所有的项求和，并且还得计算指数函数，所以该计算所需要的开销还是比较大的。

2. 不能计算的情况：

$$Z = \int e^{-x^2 - x^4} dx$$

一般而言，$G(x)$还带有一些未知参数的，准确而言要写成$G(x;\theta)$。

NCE 的作用：

通过 NCE 方法，可以对模型进行训练，从而估计出数据的原始分布。NCE 是一种用于 softmax 回归的替代方法，特别适用于大型数据集和大型词汇表的情况。

NCE 的基本思想是通过将一个二分类的问题转化为一个对比问题，来估计模型的参数。在 NCE 中，我们将原始任务(如语言模型的预测）转化为一个对比任务，即在给定上下文条件下，判断目标词是否来自于真实的数据分布还是从噪声分布中采样得到的。通过与噪声样本进行对比，可以更有效地估计模型参数，从而间接地了解数据的原始分布。

推导：

具体来说，我们保留$G(x;\theta)$，但是我们不去计算概率$p(x)$了，因为涉及配分函数。

我们去算$P(1|x) = \sigma(G(x;\theta) - \gamma) = \frac{1}{1+e^{-G(x;\theta)+\gamma}}$，（1）

其中θ是原来的参数，γ是需要优化的参数。

对上面公式的解释：[1]

NCE 将多分类问题转化为一系列二分类问题。具体地，NCE 先从数据真实分布$\tilde{p}(\boldsymbol{x})$采样（训练样本可以看成真实分布的采样），再从一个预先定义好的噪声分布$U(x)$中采样，噪声样本的数量是真实样本的 k 倍。实现中，真实样本只采 1 个，噪声样本采 k 个，也就是一个训练样本有 k 个噪声样本。真实样本被打上正样本的标签，即认为 1，而噪声样本被认为是负样本，认为为 0，对于总的 k+1 个样本。认为正样本的概率为：

$$P(1|x) = \frac{\tilde{p}(\boldsymbol{x})}{\tilde{p}(\boldsymbol{x}) + kU(x)}$$

认为是负样本的概率为：

$$P(0|x) = \frac{kU(x)}{\tilde{p}(\boldsymbol{x}) + kU(x)}$$

由于$\tilde{p}(\boldsymbol{x})$是未知量，所以我们还是使用$p(\boldsymbol{x})$去拟合$\tilde{p}(\boldsymbol{x})$。

那么：认为正样本的概率为：

$$P(1|x) = \frac{p(\boldsymbol{x})}{p(\boldsymbol{x}) + kU(x)}$$

认为是负样本的概率为：

$$P(0|x) = \frac{kU(x)}{p(\boldsymbol{x}) + kU(x)}$$

而

$$P(1|x) = \frac{p(\boldsymbol{x})}{p(\boldsymbol{x}) + kU(x)} = \frac{1}{1 + \frac{kU(x)}{p(\boldsymbol{x})}} = \frac{1}{1 + \exp\left(log\frac{kU(x)}{p(\boldsymbol{x})}\right)}$$
$$= \frac{1}{1 + \exp\left(-(logp(x) - logkU(x))\right)} = \sigma(logp(x) - \log\left(kU(x)\right))$$

由于我们希望利用 NCE 解决问题，为了简化计算，令配分函数为 1，则有

3.1. Dealing with normalizing constants

Our initial implementation of NCE training learned a (log-)normalizing constant (c in Eq. 8) for each context in the training set, storing them in a hash table indexed by the context.[3] Though this approach works well for small datasets, it requires estimating one parameter per context, making it difficult to scale to huge numbers of observed contexts encountered by models with large context sizes. Surprisingly, we discovered that fixing the normalizing constants to 1,[4] instead of learning them, did not affect the performance of the resulting models. We believe this is because the model has so many free parameters that meeting the approximate per-context normalization constraint encouraged by the objective function is easy.

$$\sigma\big(logp(x) - \log\big(kU(x)\big)\big) = \sigma\big(G(x;\theta) - \log\big(kU(x)\big)\big)$$

对 k 的解释:

NCE 目标函数的收敛性在 k 为任意正整数时都成立，只是收敛快慢有区别，原论文从理论和实验上均证明，k 越大收敛越快。

由于 NCE 是二分类问题，对于上述 k+1 个样本，我们可以用 bce 计算总损失：

$$Loss = arg\min_{\theta,\gamma} - E_{x\sim\tilde{p}(\boldsymbol{x})}\log p(1|x) - kE_{x\sim U(x)}logp(0|x) \qquad (2)$$

其中$\tilde{p}(\boldsymbol{x})$是真实样本分布，$U(x)$是噪声分布，可以是“均匀”或方便采样的分布。

现在的问题是，我们用上面 Loss 函数（2）中优化的θ是不是和原来需要优化的θ一样，根据相关文章，它们是基本一样的。

我们将上面的 Loss 函数进行改写，有：

$$Loss = -\int \tilde{p}(\boldsymbol{x})\, logp(1|x)dx - \int kU(x)\, logp(0|x)dx \quad (3)$$

$$= \int \big(\tilde{p}(\boldsymbol{x}) + kU(x)\big)\left(\tilde{p}(1|\boldsymbol{x})log\frac{\tilde{p}(1|\boldsymbol{x})}{p(1|x)} + \tilde{p}(0|\boldsymbol{x})log\frac{\tilde{p}(0|\boldsymbol{x})}{p(0|x)}\right)dx \quad (4)$$

$$= \int \big(\tilde{p}(\boldsymbol{x}) + kU(x)\big)\, KL(\tilde{p}(y|x)||p(y|x))dx \quad (5)$$

其中

$$\tilde{p}(1|\boldsymbol{x}) = \frac{\tilde{p}(\boldsymbol{x})}{\tilde{p}(\boldsymbol{x}) + kU(x)} \quad (6)$$

解释一下（3）——（4）的成立原因。
把（4）展开：
$\int \tilde{p}(\boldsymbol{x}) log\tilde{p}(1|\boldsymbol{x})dx - \int \tilde{p}(\boldsymbol{x})\, logp(1|x)dx + \int U(x)\, log\tilde{p}(0|\boldsymbol{x})dx - \int U(x)\, logp(0|x)dx$ （7）
由于$\tilde{p}(\boldsymbol{x})$和$U(x)$与参数$\theta, \gamma$无关，所以不会影响优化结果。
由于我们是希望 Loss 函数越小越好，对于（5）式，我们可以知道当 KL=0 时，loss 函数达到最小，此时有$\tilde{p}(y|x) = p(y|x)$。
备注：也可以对 Loss 函数求导，也可以求出极值点，其结果与上面是一样的，即有$\tilde{p}(y|x) = p(y|x)$。
此时

$$\tilde{p}(1|\boldsymbol{x}) = \frac{\tilde{p}(\boldsymbol{x})}{\tilde{p}(\boldsymbol{x}) + kU(x)} = p(1|x) = \sigma(G(x;\theta) - \gamma) \quad (8)$$

而

$$\tilde{p}(0|\boldsymbol{x}) = \frac{kU(x)}{\tilde{p}(\boldsymbol{x}) + kU(x)} \quad (9)$$

$$p(0|x) = 1 - \sigma(G(x;\theta) - \gamma) = \frac{e^{-G(x;\theta)+\gamma}}{1 + e^{-G(x;\theta)+\gamma}} \quad (10)$$

故由（6）/（9）式：

$$\frac{\tilde{p}(\boldsymbol{x})}{kU(x)} = \frac{\tilde{p}(1|\boldsymbol{x})}{\tilde{p}(0|\boldsymbol{x})} = \frac{p(1|x)}{p(0|x)} \quad (11)$$

即：

$$\tilde{p}(\boldsymbol{x}) = \frac{p(1|x)}{p(0|x)} * kU(x) = \frac{kU(x)}{e^{-G(x;\theta)+\gamma}} = \exp\left(G(x;\theta) - \gamma\right) kU(x)$$
$$= \exp\left(G(x;\theta) - (\gamma + logkU(x))\right)$$

这样的话，我们通过引入 NCE 就把$\tilde{p}(x)$给表达出来了而没有涉及到对配分函数的计算，其中θ, γ都是在 Loss 中进行优化的。
一个例子：

在 word2vec 模型中，我们的目标是$p(w_j|w_i) = \frac{e^{\langle u_i, v_j \rangle}}{z}$，其中$p(w_j|w_i)$代表了在给定中心词$w_i$的条件下上下文词的概率。$u_i, v_j$代表了中心词和上下文两套不同的词空间。
通过引入负采样（NCE 的变体）：
优化目标是：

$$arg\min_{u,\ v} - E_{w_j \sim \tilde{p}(w_j|w_i)} \log\sigma(\langle u_i, v_j \rangle) - E_{w_j \sim \tilde{p}(w_j)} \log\left[1 - \sigma(\langle u_i, v_j \rangle)\right]$$

与（2）相比，我们默认$\gamma = 0$。

由上面的推导可知：$\tilde{p}(w_j|w_i) = \frac{p(1|w_j, w_i)}{p(0|w_j, w_i)} p(w_j) = e^{\langle u_i, v_j \rangle} \tilde{p}(w_j)$

即：

$$log\frac{\tilde{p}(w_j|w_i)}{\tilde{p}(w_j)} = \langle u_i, v_j \rangle$$

其中左边代表点互信息，用来衡量两个词之间的关联程度，一般而言，其值越高，表示两个词之间的关联程度越高。我们并没有计算配分函数 Z。
参考：
[1] 深入理解 Noise Contrastive Estimation (NCE) - 知乎 (zhihu.com)