1. max 的平滑近似：

$$\mathrm{Max}(a,b) \approx \log(e^a+e^b)$$

$$\mathrm{Max}(x_1,x_2,\cdots x_n) \approx \log \sum_{i=1}^{n} e^{i}$$

$$\mathrm{Relu}(x) = \mathrm{Max}(x,0) \approx \log(1+e^x)$$

本质上是找到一个导数增长比线性更快的函数（这里是 e^x），拉开数值差距，相加后再取导函数（log）

同理也可以取先平方和再开根号，但是二次函数的导数增长没有指数快，所以近似效果没那么好

2. Softmax+交叉熵作为损失函数的推导：

$y=f(x)=(y_1,y_2,\cdots,y_n)$ 为网络输出，其中 $\boxed{y_k}$ 为真值对应的分数

我们希望 $y_k > y_i$，其中 $i\neq k$，定义损失函数 L：

$L = \max\{\max_{i\neq k}\{y_i\} - y_k, 0\}$，$\Longleftrightarrow$ 理解为希望 y_k 比其它值中最大的更大 $\max_{i\neq k}\{y_i\}$

$\Updownarrow$ $\boxed{\text{用 } \log\sum e^{y_i} \approx \max\{y_i\}\text{，平滑近似 max 函数}}$

$$L \approx \max\{\log\sum_{i\neq k} e^{y_i} - y_k, 0\}$$

$$\approx \log\left[e^0 + e^{(\log\sum_{i\neq k} e^{y_i} - y_k)}\right]$$

$$= \log\left(1 + \frac{\sum_{i\neq k} e^{y_i}}{e^{y_k}}\right)$$

$$= \log\left(\frac{e^{y_k}}{e^{y_k}} + \frac{\sum_{i\neq k} e^{y_i}}{e^{y_k}}\right)$$

$$= \log\frac{\sum e^{y_i}}{e^{y_k}}$$

$$= -\log\frac{e^{y_k}}{\sum e^{y_i}}$$

$$\Longrightarrow L \approx -\log\frac{e^{y_k}}{\sum e^{y_i}}$$

3. Margin 推导：

更进一步，我们不单希望 y_k 比 y_i 大

$\Rightarrow y_k > y_i, \ \forall i \neq k.$

还希望 y_k 比 y_i 大一个距离 m：

$\Rightarrow y_k > y_i + m, \ \forall i \neq k$

$\Rightarrow y_k - m > y_i, \ \forall i \neq k.$

将 $y_k - m$ 代入到刚推导的

$L = -\log \frac{e^{y_k}}{\sum e^{y_i}}$ 的 y_k 中（换元，重新推也可以有一样的结果）

$$\Rightarrow L = -\log \frac{e^{y_k - m}}{e^{y_k - m} + \sum_{i \neq k} e^{y_i}}$$

更一般的形式：

更一般的，m 可以不是一个固定的常数

可以取 m 为 y_k 的函数

$m = \phi(y_k) > 0$ （我们希望间隔大于0）

设 $g(y_k) = y_k - \phi(y_k) = y_k - m$

有

$$L = -\log \frac{e^{g(y_k)}}{e^{g(y_k)} + \sum_{i \neq k} e^{y_i}}$$ 为 margin 的一般形式 (1)

不难看出，arcface, normface 等都是(1)式的特例

我们可以自己构造新的 margin-softmax.

如：取 $\phi(y_k) = (1-y_k)^2$

$\Rightarrow y_k - (1-y_k)^2$

$$L_{new} = -\log \frac{e^{y_k - (1-y_k)^2}}{e^{y_k - (1-y_k)^2} + \sum_{i \neq k} e^{y_i}}$$ （随便构造的）

逻辑上合理，真实效果未测试

4. 多正样本（多对多）推导：

之前为1个正样本，n个负样本，

假设现在有m个正样本，n个负样本：{~~$m_1, m_2,$~~

设正样本下标：$\{p_1, p_2, \cdots p_m\}$，负样本下标 $\{n_1, n_2, \cdots n_n\}$.

$$L = \max\{\max_i\{y_{n_i}\} - \min_j\{y_{p_j}\}, 0\} \Longleftrightarrow$$ 希望$\underbrace{\text{最小的正分数}}_{\min_j\{y_{p_j}\}}$比$\underbrace{\text{最大负分数}}_{\max_i\{y_{n_i}\}}$还大.

$$\Updownarrow \quad \boxed{\min\{x\} = -\max\{-x\}}$$

把min换成max，再用max的平滑近似

$$L \approx \max\left\{\log\sum_i e^{y_{n_i}} + \log\sum_j e^{-y_{p_j}}, 0\right\}$$

$$= \max\left\{\log\left(\sum_i e^{y_{n_i}} \cdot \sum_j e^{-y_{p_j}}\right), 0\right\}$$

$$= \log\left(e^0 + e^{\log\left(\sum_i e^{y_{n_i}} \cdot \sum_j e^{-y_{p_j}}\right)}\right)$$

$$= \log\left[1 + \sum_i e^{y_{n_i}} \cdot \sum_j e^{-y_{p_j}}\right]$$

这就是旷视 Circle loss 最简单的形式.
有兴趣可以加上margin推一遍.