

baseline代码解析

一、代码规范：

见深度之眼比赛代码规范

二、数学基础：

- Beta(a,a) 分布
- 图像的离散傅立叶变换
- 梯度和显存

2.1 Beta分布

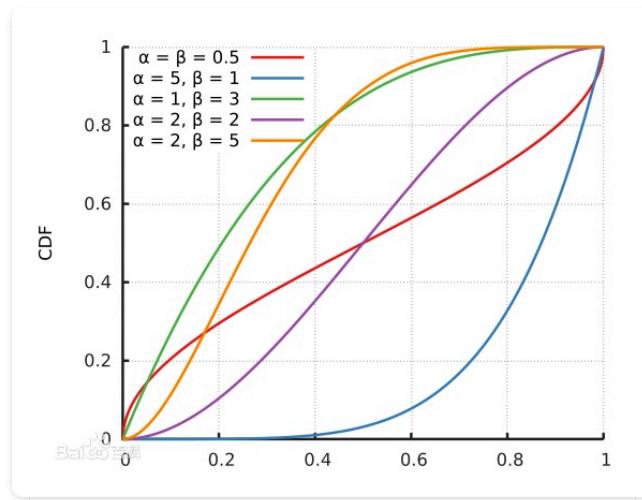
Beta 分布：

$$P(\theta|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}$$

其中 $\Gamma(x)$ 为出名的 gamma 函数，它是阶乘运算的推广：

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$





Beta 分布的随机变量的定义域在 $[0,1]$ 之间。

2.2 图像的离散傅立叶变换

时域转频域：

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

$$e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} = \cos(2\pi\frac{ux}{M}) + j \sin(2\pi\frac{vy}{N})$$

频域转时域：

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

图像等价于二维波，图像的傅立叶变换，就是将图像变为一系列二维波的叠加的和。

看一个 3D 图来感受一下。

2.3 梯度和显存

$$loss = \frac{1}{N} \sum_{i=1}^N l(y_i, \hat{y}_i)$$

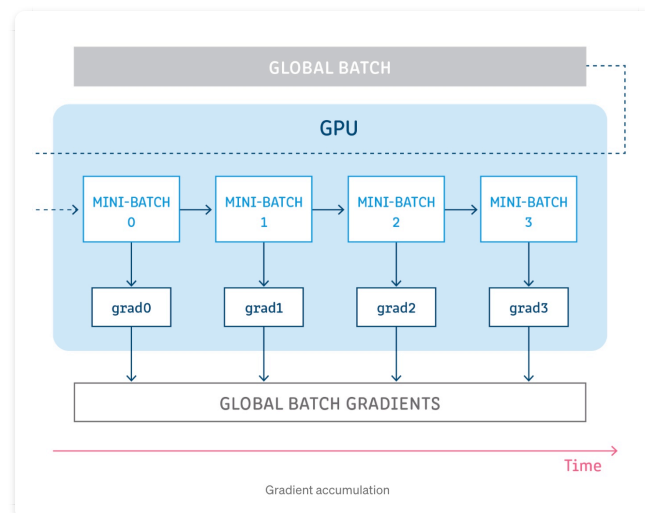
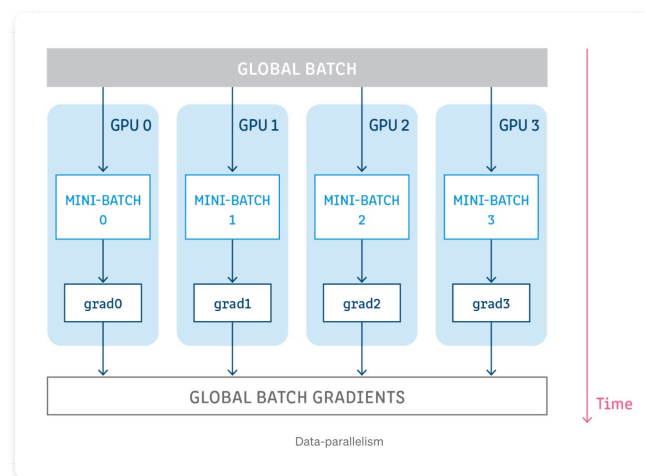
训练时，保存在显存中的变量：

- 模型参数
- 优化器的变量（一些中间变量，如 momentums）
- 中间临时变量（前向传播和后向传播的中间变量）

- 工作空间（内核实现局部变量的临时内存）

当 batchsize 变大后，样本越多，中间临时变量就会越来越多。

两种解决方法







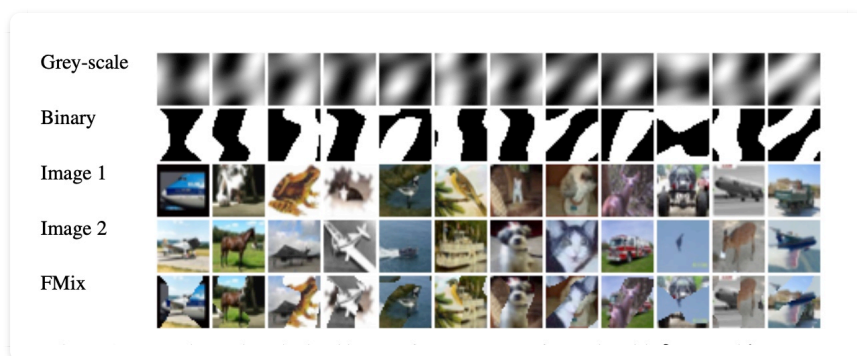
三、知识点：

- mixup
- cutmix(cutout)
- fmix

$$\text{mix}(x_1, x_2, m) = m \odot x_1 + (1 - m) \odot x_2$$

$$\text{mix}(y_1, y_2) = \text{rate} \cdot y_1 + (1 - \text{rate}) \cdot y_2$$

	ResNet-50	Mixup [48]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4



四、涉及的三方库：

- [tqdm](#)
- [albumentations](#)
- [timm](#)

五、比赛难点：

- 数据噪声与主动学习