

EMA算法

EMA 表示指数移动平均(Exponential Moving Average),是一种给予近期数据更高权重的平均方法,可以看作是一种数据平滑技术。

在**神经网络** 训练中使用EMA的主要目的是为了获取更加平滑和稳定的模型参数,从而提高模型的 **泛化能力** 。

EMA的计算公式如下:

```
1 | ema_t =  $\alpha$  * ema_{t-1} + (1 -  $\alpha$ ) *  $\theta_t$ 
2 |
3 | 其中:
4 | ema_t 是时刻t的指数移动平均结果
5 | ema_{t-1} 是时刻t-1的指数移动平均结果
6 |  $\theta_t$  是时刻t的模型参数
7 |  $\alpha$  是平滑系数,一般设置为0.999或0.99
8 | 可以看出,EMA就是一种对模型参数做指数加权平均的技巧。
```

EMA的工作流程是:

在每个训练迭代后,记录下当前**模型** 参数 θ_t

计算 θ_t 和上一轮EMA结果 ema_{t-1} 的加权平均,生成当前时刻的 ema_t , 不断迭代更新,记录每次训练的EMA参数。最终训练结束后,用训练过程中记录的**模型**中原始的参数,从而获得一个泛化能力更强的模型。

EMA可以看作是一种增强模型并提高其泛化能力的简单而有效的技巧。

关于**PyTorch** 中如何实现EMA,可以参考这样一个简单的代码示例:

```
1 | import torch
2 |
3 | # 假设模型参数为一个字典
4 | model_params = {'w': torch.tensor([1.], requires_grad=True),
5 |                 'b': torch.tensor([0.5], requires_grad=True)}
6 |
7 | # 设置EMA的平滑系数 $\alpha$ 
8 | alpha = 0.9
9 |
10 | ema_params = model_params.copy() # ema参数初始化为模型参数的一个拷贝
11 |
12 | for t in range(100):
13 |     # 假设在每次迭代中都会更新模型参数
14 |     model_params['w'] += 0.01
15 |     model_params['b'] += 0.01
16 |
17 |     # 更新EMA参数
18 |     for name in ema_params:
19 |         ema_params[name] = alpha * ema_params[name] + (1 - alpha) * model_params[name]
20 |
21 | # 最后使用EMA参数替换原始参数
22 | model.load_state_dict(ema_params)
```

主要步骤包括:

- 1. 初始化EMA参数为模型参数的一个拷贝
- 2. 在每次迭代中更新模型参数
- 3. 用EMA公式更新EMA参数
- 4. 最后替换模型参数为EMA参数

对于一个完整的模型,需要在每次训练迭代后调用类似上述的EMA更新代码,来维护每个参数的指数 **移动平均** 值,从而获得一个更加平滑和稳定的模型