

Caffe 参数理解

Caffe是非常火热的深度学习框架之一，学习并使用它是我们在学习尤其是工程运用中是非常有必要的。

下面的内容将对**caffe**训练模型的一些配置文件加以说明，更好的帮助大家理解和运用**caffe**。**solver.prototxt**是**caffe**模型训练的配置文件。

solver.prototxt 的说明

```
net: "*_train_test.prototxt"
test_iter: 400
test_interval: 500
iter_size: 4
test_initialization: false
display: 40
average_loss: 40
base_lr: 0.00005
lr_policy: "step"
stepsize: 2000
gamma: 0.96
max_iter: 10000
momentum: 0.9
weight_decay: 0.0002
snapshot: 20
snapshot_prefix: "path"
solver_mode: CPU or GPU
```

具体说明：

在**caffe**中，一次迭代(iteration)指的是一个batch，而不是一帧图像。

test_iter: 在测试时，需要迭代的次数(每迭代**test_iter**次，显示训

训练的lr, loss), $\text{test_iter} * \text{batchsize}$ (测试集的) = 测试集的大小, 测试集batchsize可以在prototxt文件里设置。

test_interval: interval区间的意思, 该参数表示: 在训练的时候, 每迭代n次(if **test_interval**=n, 训练集迭代n次)就进行一次测试。

caffe在训练的过程中, 是边训练边测试进行的。训练中每迭代500次(32000个训练样本, $\text{batchsize}=64, \text{iter_size} * \text{train batch size} = \text{训练样本量}$), 也就是经过**test_interval**计算一次测试误差。其中, 计算一次测试误差就需要包含所有的测试图片(这里为10000)。这样, 可以认为在一个epoch里, 训练集中的所有样本都要遍历一遍, 但测试集的所有样本至少遍历一次, 具体多少次有max_iter决定。

iter_size: $\text{iter_size} * \text{train batch size} = \text{实际的batch size}$ 深度学习为了加快运算速度, 将数据分成小的batch, 但是参与batch太小, 代价函数进行权重衰减太快, 加上iter_size之后, 取iter_size次的平均再进行权重衰减, 比较具有代表性。

lr_policy: 权重衰减策略。

lr_base: 开始的时候学习速率, 会在此基础上进行衰减, 注意train_net中的lr_mult*lr_base代表那一层的实际学习速率。

lr_policy: 学习率的调整策略:

- - fixed: 保持base_lr不变.
- - step: 如果设置为step, 则还需要设置一个stepsize, 返回 $\text{base_lr} * \gamma^{\lfloor \text{iter} / \text{stepsize} \rfloor}$, 其中iter表示当前的迭代次数
- - exp: 返回 $\text{base_lr} * \gamma^{\text{iter}}$, iter为当前迭代次数
- - inv: 如果设置为inv, 还需要设置一个power, 返回 $\text{base_lr} * (1 + \gamma * \text{iter})^{-\text{power}}$
- - multistep: 如果设置为multistep, 则还需要设置一个stepvalue。这个参数和step很相似, step是均匀等间隔变化, 而multistep则是根据stepvalue值变化
- - poly: 学习率进行多项式误差, 返回 $\text{base_lr} (1 - \text{iter}/\text{max_iter})^{\text{power}}$

- - sigmoid: 学习率进行sigmoid衰减, 返回 $\text{base_lr} (1/(1 + \exp(-\gamma * (\text{iter} - \text{stepsize}))))$

迭代次数、批量大小等关系描述

1. Batch Size

每迭代一次, 网络训练图片的数量。例如: 如果你的 $\text{batchsize}=256$, 则你的网络每迭代一次, 训练256张图片; 则, 如果你的总图片张数为1280000张, 则要想将你所有的图片通过网络训练一次, 则需要 $1280000/256=5000$ 次迭代。

2. Number of Iterations

Training:

训练集60k, batch size: 64, maximum_iterations= 10k. 因此这里将有 $10k*64 = 640k$ 帧图像用来学习。这意外着, 这里将有10.6 个 epochs.(Number if epochs is hard to set, you should stop when net does not learn any more, or it is overfitting)

Val:

10k, batch size: 100, test_iterations: 100, So, $100*100: 10K$, 所有验证集里的样本都将遍历完备。

举个例子, 如果你有20k的训练样本, 你的 $\text{batch_size}=100$ and $\text{test_iterations}: 200$, 这样你的验证集才能遍历一遍。

总之, 关于测试的参数 "**test_iterations**" 和 "**batch size**" 是依赖于测试集样本的容量。训练集的参数 "**maximum_iterations**" 和 "**batch size**" 取决于你想要训练所有样本的 epoch(一个 epoch 是整个训练集遍历一遍)。

下面总结一些关系等式:

max_iters = epochs x (training_set / batch_size)

max_iters = epochs x 迭代次数

