Major Hyper-Parameters and Search Space

I. Learning Rate

学习率定义了SGD过程中的步长。

1. 手动调整

初始设定一个常数，随着训练的进行，逐渐手动减小学习率。

2. 学习率衰减

线性衰减

$lr = lr0 / (1+kt)$

lr0 和 k 为超参数，t可以为迭代时间或迭代次数

指数衰减

$lr = lr0 \cdot exp(-kt)$

lr0 和 k 为超参数，t可以为迭代时间或迭代次数或epoch数

问题：

1 需要训练前设定，依赖用户调参经验；

2 LR只随着时间或步数调整，所有层共享相同得LR值。

改进：LARS(每层单独调整LR，层数也为超参数)

Tips：

1 sensitivity test：比较不同超参数的影响；

2 初始LR稍大些，以便加快收敛；

3 用Log scale来更新LR, ，此时可使用指数衰减，可适用很多场景；

4 多尝试，指数衰减并不一定是最优选择

II. Optimizer

Optimizer 常用于加速梯度下降收敛，提高算法精确度。

1. mini-batch gradient descent

1. 算法原理：略

2. 优缺点

优点：与SGD相比，mini-batch GD具有向量化带来的加速，同时降噪，有助于收敛。

缺点：

引入了新的超参数mini-batch size；

较其他优化算法更耗时，收敛性依赖于初始值和LR

3. 使用建议

LR与mini-batch size互相参考设定

2. Mometum

1. 算法原理：

vdw = βvdw +(1－β)dw

w = w－lr∗vdw

2. 优缺点

优点：加快收敛速度。

3. 使用建议

beta = 0.9/0.99/0.999

3. RMSprop

1. 算法原理：

Sdw = βSdw + (1 － β)dw2

Sdb =βSdb+(1－β)db2

w = w－lr∗ dw sqrt(Sdw)

b = b－lr∗ db sqrt(Sdb)

2. 优缺点

优点：加快收敛速度。

3. 使用建议

beta = 0.9

4. Adam

1. 算法原理：

vdw = β1vdw + (1 － β1)dw

vdb = β1vdb + (1－β1)db

Sdw = β2Sdw + (1 − β2)dw2

Sdb = β2Sdb + (1 − β2)db2

vdw-corrected = vdw/(1−β1t)

vdb-corrected = vdb(1−β1t)

Sdw-corrected = Sdw/(1−β2t)

Sdb-corrected = Sdb/(1−β2t)

w = w−lr∗ vdw-corrected/ sqrt(Sdw-corrected + ε)

b= b−lr∗ vdb-corrected/ sqrt(Sdb-corrected + ε)

2. 优缺点

优点：加快收敛速度，通常是默认选择。

3. 使用建议

beta1 = 0.9

beta2= 0.999

ε = 10−8

III. hidden layers and width of layers

隐藏层单元数和网络深度设置，暂无明确建议。

IV. regularization

L1/L2/dropout 每种正则化方法会引入新的超参数

V. Activation functions

Sigmoid functions （可能会导致梯度消失）

Softmax functions（输出层）

Tanh （可能会导致梯度消失）

Relu

Leaky relu