Deep Learning

Yu Xueqing

2022 年 7 月 27 日

1 Optimization

Optimization fail: local minima or saddle point 如何分辨?

- $L(\phi) = L(\phi') + (\phi - \phi')^T * g + \frac{1}{2} * (\phi - \phi')^T * H * (\phi - \phi')$
- $g = \nabla L(\phi)$
- H-Hessian 矩阵:$H_{ij} = \frac{\partial^2 L(\phi')}{\partial \phi_i \partial \phi_j}$
- critial point $g = 0$, $L(\phi) = L(\phi') + \frac{1}{2} * (\phi - \phi')^T * H * (\phi - \phi')$
  只需考虑 H 是正定/负定/不定，对应极小值/极大值/鞍点

高维度中 local minima 其实很少（网络参数多），多数是 saddle point.

2 Batch

- Shuffle: 每个 Epoch 的 batch 分配都不一样 small batch vs. large barch: large batch 更新比较平稳，small batch 随机性大。应用平行运算，大 batch 计算时间并不比小 batch 大太多。small batchsize 跑一个 epoch 时间长。正比于 n/batchsize
- 训练结果上 batchsize 小反而更好：不同样本的 L function 不同，optimization 不容易陷入 critical point. large batch 容易 overfit
- flat minima vs. sharp minima: small batch 更容易陷入 flat minima, large batch 更容易陷入 sharp minima
- 总结：small batch 训练时间长，optimization 和 generalization 都更好

3 Momentum

额外加上前一步移动的方向（势能）。$m^1 = \lambda m^0 - \eta g^0$, $m^2 = \lambda m^1 - \eta g^1$
有概率走出 local minima

4 Adaptive Learning Rate

- training stuck≠small gradient
- learning rate 太大-> 在低谷两端波动
  learning rate 太小-> 走不动

- original: $\phi_i^{t+1} = \phi_i^t - \eta g_i^t$
  parameter dependent: $\phi_i^{t+1} = \phi_i^t - \frac{\eta}{\sigma_i^t} g_i^t$, η-learning rate
- $\sigma_i^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t}(g_i^t)^2}$
- RMSProp:$\sigma_i^t = \sqrt{\alpha(\sigma_i^{t-1})^2 + (1-\alpha)(g_i^t)^2}$
- 常用 optimizer—Adam: RMSProp+Momentum
- Learning Rate Scheduling: η^t,η 和时间有关
  Warm Up: learning rate 先变大再变小

5 Classification

计算出 y,$y' = softmax(y)$, 与 $\hat{y}$ 计算 loss

- soft-max: $(y_i)' = \frac{exp(y_i)}{\sum exp(y_i)}$
- loss function: cross-entropy $e = \sum -y_i * ln(y_i)$