Deep Learning Technology and Application

Ge Li

Peking University

Table of contents

关于学习率的优化

梯度下降过程中的权重更新：

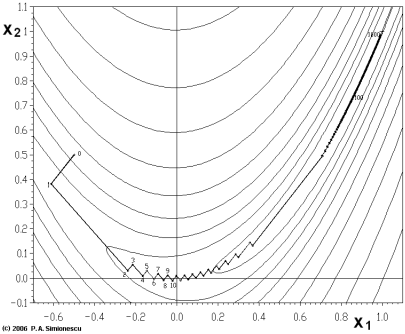
$$\theta = \theta - \alpha \nabla_\theta J(\theta)$$

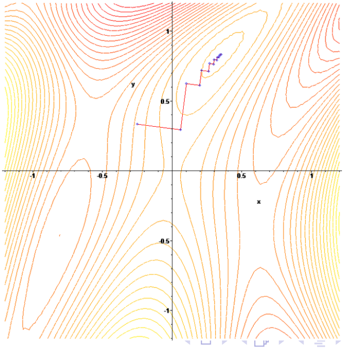
学习率的选择，是一个重要但困难的问题：

- 学习率如果太小，训练收敛会非常慢；学习率过大，也会阻碍收敛，并导致损失函数在最小值附近波动。
- 如果按照事先定义的"学习率递减方法"修改学习率，会导致最终的学习率更新失效，且很难找到符合数据集特性的固定递减系数。
- 在训练过程中，不应该使用统一的学习率，因为常常出现的情况是：有些参数已经不需要调整，但有些参数需要较大的调整。
- 学习率的调整不仅要跳出"local optimum"，还需要规避"鞍点问题"。鞍点问题常常会严重影响参数的调整，在鞍点，沿某个方向梯度上升，沿另一个方向梯度下降。

先讨论最容易出现的锯齿问题（Zigzagging Problems）：

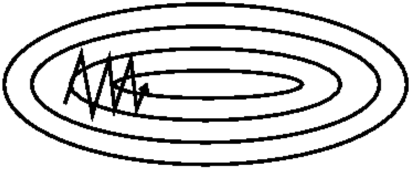
- 在一些个方向梯度不等的地带，GD 方法容易出现优化曲线的“锯齿”，从而减慢训练过程。
- 在该情况下，一边的梯度变化大，另一边较小，则会向梯度变化大的方向严重偏移，较大的梯度调整后，又会造成向相反方向调整；
- $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - {x_1}^2)^2$.
- $F(x, y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right)\cos(2x + 1 - e^y)$.

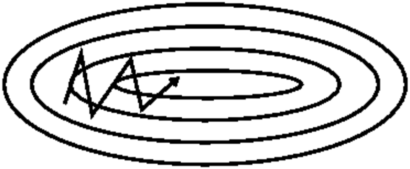




Momentum

- 可否通过学习率的调整，使 GD 的调整曲线变得更加"顺畅"?





Momentum

- 在更新模型参数时，对于那些当前梯度方向与上一次梯度方向相同的参数，进行加强，即在这些方向上的参数更新更快了；
- 对于那些当前梯度方向与上一次梯度方向不同的参数，进行削减，即在这些方向的参数更新上减慢了。
- Momentum based Gradient Descent:

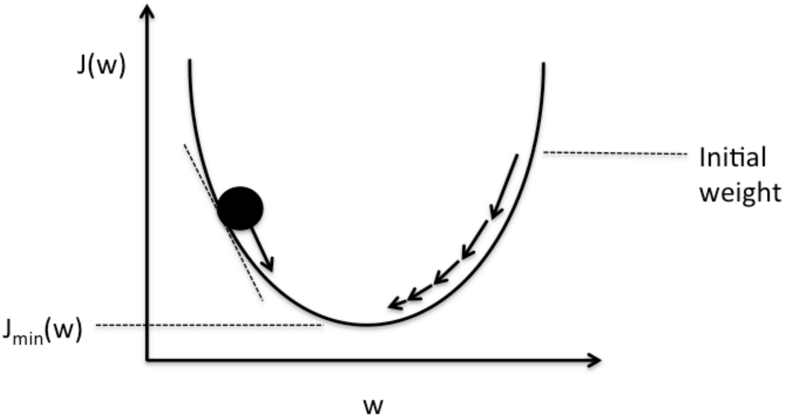
$$v_t = \gamma v_{t-1} + \alpha \nabla_\theta \left(\frac{1}{m} \sum_t^m J(x^{(i)}, y^{(i)}; \theta) \right)$$

$$\theta = \theta - v_t$$

一般情况下，动量项参数 $\gamma < 0.9$

Nesterov

- 但如果基于动量的调整过大，会不会出现调整过度的问题？



Nesterov Accelerated Gradient

- 一种解决方法是，在进行基于动量的调整前，可否“预测”一下，这次的调整会造成怎样的影响。
- 所以，可否事先按照上次基于动量的调整结果计算一下，即 $\theta - \gamma v_{t-1}$，然后用这个计算的结果对学习率的调整方向进行“小幅”的纠正？
- 由于 $t-1$ 时刻的调整会以较大的“动量”影响 t 时刻的调整，因此，即便是方向上的“小幅”调整，对整个调整方向的影响仍然是显著的。
- Nesterov 在其论文中，证明了这种调整的有效性。

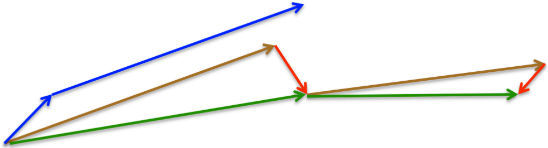
  Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence o(1/k2). Doklady ANSSSR (translated as Soviet.Math.Docl.), vol. 269, pp. 543– 547.

Nesterov Accelerated Gradient

- NAG:

$$v_t = \gamma v_{t-1} + \alpha \nabla_\theta \left(\frac{1}{m} \sum_{t}^{m} J(\theta - \gamma v_{t-1}) \right)$$

$$\theta = \theta - v_t$$



- 论文证明，这种调整方法对很多 RNN 相关任务，效果非常明显。

Bengio, Yoshua, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. "Advances in optimizing recurrent networks." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

Adagrad

- 上述调整方法均针对所有参数的调整量进行，然而在训练过程中，对不同的参数应该采取不同的调整策略；
- 对于调整频繁出现的训练数据，应该适当减小其对参数调整的影响，而对相对稀疏的参数，应该适当增大其对参数调整的影响。
- Adagrad (Adaptive Gradient) 方法即根据这一原理展开。

  Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12, 2121–2159.
- 这种方法很快被证实，对于训练数据分布不均衡的数据集非常有效。

  Pennington, J., Socher, R., Manning, C. D. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 1532–1543.

Adagrad

- 在上述模型中，每个模型参数 θ_i 使用相同的学习速率 α，而 Adagrad 在每一个更新步骤中对于每一个模型参数 θ_i 使用不同的学习速率 α_i；
- 设第 t 次更新步骤中，目标函数的参数 θ_i 梯度为 $g_{t,i}$，即：

$$g_{t,i} = \nabla_\theta J(\theta_i)$$

- 则，传统的 SGD 的更新方程表示为：

$$\theta_{t+1,i} = \theta_{t,i} - \alpha \cdot g_{t,i}$$

- 而，Adagrad 对每一个参数使用不同的学习率，则其更新方程变为：

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

其中，$G_t \in \mathcal{R}^{d\times d}$ 是一个对角矩阵，其中第 i 行的对角元素 e_{ii} 为过去到当前第 i 个参数 θ_i 的梯度的平方和，ϵ 是一个平滑参数，为了使得分母不为 0（如可取 $\epsilon = 1e-8$）

Adagrad

写成矩阵形式：

$$\Delta\theta_t = -\frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

- Adagrad 主要优势在于它能够为每个参数自适应不同的学习速率，而一般的人工都是设定为 0.01。
- 其缺点在于:
  - 由于需要计算参数的整个梯度序列的平方和，在训练数据较大时，计算量较大，
  - 学习速率趋势是不断衰减最终达到一个非常小的值，开始很大，最后很小，以致失效。

Adadelta

- Adadelta 提出的目的也是为了避免 Adagrad 对学习速率的调整过于“鲁莽”的问题；

  Zeiler, Matthew D. ”ADADELTA: AN ADAPTIVE LEARNING RATE METHOD.”
- 同时，为了避免计算整个梯度序列的平方和，Adadelta 采用了“窗口”技术，即，仅对固定窗口内的 w 个梯度序列进行计算；
- 当前的梯度平方的平均值（$E[g^2]_t$）仅依赖于前一个时刻的平均值和当前的梯度；

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma)g_t^2$$

- 可以把 γ 设为一个类似于动量的值，如 0.9 附近.

Adadelta

- 下面给出 Adadelta 的表达式：从 $\Delta\theta_t$ 的表达式开始:

$$\Delta\theta_t = -\alpha \cdot g_{t,i}$$
$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

对比 Adagrad 的公式：

$$\Delta\theta_t = -\frac{\alpha}{\sqrt{G_t + \epsilon}} \odot g_t$$

用 $E[g_t^2]$ 替换 G_t:

$$\Delta\theta_t = -\frac{\alpha}{\sqrt{E[g_t^2] + \epsilon}} g_t$$

Adadelta

- 为简便，直接将分母换为梯度的均方根（Root Meam Square），简短表示为：$RMS[g]_t$ 得：

$$\Delta\theta_t = -\frac{\alpha}{RMS[g]_t} \cdot g_t$$

- 还注意到，梯度的更新中 α 与分母仍然可能不成比例，继续做以下替换:

$$\text{定义：} E[\Delta\theta^2]_t = \gamma E[\theta^2]_{t-1} + (1-\gamma)\Delta\theta_t^2$$

- 于是，得到：

$$RMS[\Delta\theta]_t = \sqrt{E[\Delta\theta^2]_t + \epsilon}$$

Adadelta

- 又因为 t 时刻的 $RMS[\Delta\theta]_t$ 并不知道，于是用 t 时刻之前的参数更新的 RMS 来代替，即：用 $RMS[\Delta\theta]_{t-1}$ 代替 α，最终得到 Adadelta 的更新规则：

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

RMSprop

RMSprop 由 Geoff Hinton 在他的 Coursera 课程中提出。
RMSprop 与 Adadelta 几乎在同一时间提出，只是可以看做 Adadelta 的简化版本：

- Adadelta:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma)g_t^2$$

- RMSprop

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$$
$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{E[g_t^2] + \epsilon}} \cdot g_t$$

- 可见，RMSprop 方法也是用“衰减的梯度均方根误差”去除学习率。Hinton 建议 γ 可以设置为 0.9,α 可以设置为 0.001.

Adam-Adaptive Moment Estimation

- Adam（自适应的矩估计）也是一种不同参数自适应不同学习速率方法，与 Adadelta 与 RMSprop 区别在于，它计算历史梯度衰减方式不同，不使用历史平方衰减，其衰减方式类似动量：

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

- m_t 与 v_t 分别是梯度的一阶矩和二阶矩的估计值，初始为 0 向量；

Adam-Adaptive Moment Estimation

- 然而，它们通常被偏置化为趋向于 0 的向量，特别是当衰减因子(衰减率)β_1, β_2 接近于 1 时；
- 为了改进这个问题，可以改进上式中的偏置项：利用经过偏置修正的一阶和二阶矩估计来计算 m_t 与 v_t：

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

- 类似 Adadelta 与 RMSprop 方法，可以得到 Adam 方法的更新规则：

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t$$

- Adam 提出者建议 $\beta_1 = 0.9, \beta_2 = 0.9999, \epsilon = 10^{-8}$. 实验证实，Adam 方法较其他方法有更好的应用效果。

Nadam - accelerated Adaptive Moment Estimation

- 如同 NAG 方法一样，我们可以利用 m_{t-1} 对调整结果进行预估：

$$g_t = \nabla_{\theta_t} J(\theta_t - \gamma m_{t-1})$$
$$m_t = \gamma m_{t-1} + \alpha g_t$$
$$\theta_{t+1} = \theta_t - m_t$$

我们发现，在上式中，我们用了两次动量调整，可以简化：

$$g_t = \nabla_{\theta_t} J(\theta_t)$$
$$m_t = \gamma m_{t-1} + \alpha g_t$$
$$\theta_{t+1} = \theta_t - (\gamma m_t + \alpha g_t)$$

将如上结果，代入 Adam 算法中：

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon}\left(\frac{\beta_1 m_{t-1}}{1-\beta_1^t} + \frac{(1-\beta_1)g_t}{1-\beta_1^t}\right)$$

Nadam - accelerated Adaptive Moment Estimation

刚刚的结论：

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon}\left(\frac{\beta_1 m_{t-1}}{1-\beta_1^t} + \frac{(1-\beta_1)g_t}{1-\beta_1^t}\right)$$

且我们看到，$\frac{\beta_1 m_{t-1}}{1-\beta_1^t}$ 可以看作是对前一步的一个修正
因此可以换为 $m\hat{_{t-1}}$，由此可以得到 Nadam 的修正公式：

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon}\left(\beta_1 \hat{m}_{t-1} + \frac{(1-\beta_1)g_t}{1-\beta_1^t}\right)$$

Adam 的问题

- 最近，在 Adam 系列方法的应用中，研究者发现在某些情况下，Adam 系列方法的效果并不好，甚至不如基于动量的 SGD 方法。
- Reddi 等人经过研究发现，这是由于在 Adam 系列方法中，通过求前序梯度的平均来调整学习率，如此以来，对于有些低频出现却能够提供较大权重调整信息的 minibatches 变得不公平，它们的影响被平均值"掩盖"了。

  Reddi, Sashank J., Kale, Satyen, Kumar, Sanjiv. On the Convergence of Adam and Beyond. Proceedings of ICLR 2018.

- 基于此，最近人们提出了一种对 Adam 方法进行调整的新方法：AMSGrad.

AMSGrad

Adam 中：

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2$$

换成：

$$\hat{v}_t = max(\hat{v_{t-1}}, v_t)$$

最终得到 AMSGrad 的更新公式：

$$\begin{aligned} \hat{m}_t &= \beta_1 m_{t-1} + (1-\beta_1)g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \\ \hat{v}_t &= max(\hat{v_{t-1}}, v_t) \\ \theta_{t+1} &= \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t \end{aligned}$$

Thanks.