随机梯度下降（SGD）的定义与基本概念

引言

随机梯度下降（Stochastic Gradient Descent, SGD）是一种广泛应用于机器学习和深度学习中的优化算法，用于最小化目标函数。与传统的批量梯度下降相比，SGD 每次迭代仅使用一个或少量样本来更新参数，因此在处理大规模数据集时更高效。本文将详细介绍 SGD 的定义、原始理念、算法的应用，以及不同的算法变种。

1. 随机梯度下降（SGD）

定义与算法步骤

定义 1 (随机梯度下降法 (SGD))**.** 随机梯度下降是一种迭代优化算法，用于最小化目标函数 $f(\theta)$。每次迭代中，*SGD* 使用一个或少量的样本来估计梯度并更新参数 θ:

$$\theta_{k+1} = \theta_k - \alpha_k \nabla f_{i_k}(\theta_k)$$

其中:

- θ_k 是第 k 次迭代时的参数。
- α_k 是学习率，决定每次更新的步长。
- $\nabla f_{i_k}(\theta_k)$ 是在样本 i_k 上的梯度估计。

算法步骤

SGD 的基本算法步骤如下:

1. 初始化参数 θ_0。
2. 对于每次迭代 $k = 0, 1, 2, \ldots$:
   (a) 随机选择一个或少量训练样本 i_k。
   (b) 计算在样本 i_k 上的梯度 $\nabla f_{i_k}(\theta_k)$。
   (c) 更新参数:
   $$\theta_{k+1} = \theta_k - \alpha_k \nabla f_{i_k}(\theta_k)$$
3. 重复步骤 2，直到满足停止条件（如达到最大迭代次数或收敛标准）。

例 1 (线性回归中的 SGD 应用)**.** 考虑一个线性回归问题，目标是最小化均方误差（*Mean Squared Error, MSE*）：

$$f(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

其中，$h_\theta(x) = \theta^T x$ 是假设函数，m 是样本数量。

在 *SGD* 中，每次迭代选择一个样本 $(x^{(k)}, y^{(k)})$，计算梯度并更新参数：

$$\theta_{k+1} = \theta_k - \alpha_k (h_\theta(x^{(k)}) - y^{(k)}) x^{(k)}$$

这种方法在大规模数据集上的效率远高于批量梯度下降。

SGD 的原始理念

SGD 的原始理念基于以下几点：

- **随机性与高效性**—与批量梯度下降需要使用数据集中的所有样本来计算梯度相比，SGD 每次迭代只需要随机选择一个或少量样本，因此大大降低了每次更新的计算开销。
- **梯度的方差和路径的随机性**—由于只使用部分数据，SGD 估计的梯度带有一定的随机性，这也导致参数更新时的路径有一定的“随机跳动”，这种随机性可以帮助 SGD 跳出局部最优解，更容易找到全局最优解；当然也可能让结果更糟。
- **内存效率**—SGD 只需要存储当前样本的梯度，非常适合处理大规模数据集。
- **在线学习能力**—SGD 可以不断更新模型参数，这使它在数据流和在线学习场景中应用很好。

2. 随机梯度下降的变种

为了提升 SGD 的收敛速度和稳定性，研究人员提出了多种变种：

动量法（Momentum）

动量法通过引入动量项累积之前的梯度信息，加速收敛并减少震荡：

$$v_{k+1} = \beta v_k + \nabla f_{i_k}(\theta_k)$$

$$\theta_{k+1} = \theta_k - \alpha_k v_{k+1}$$

其中，β 通常取值在 0.9 左右，帮助加速在梯度方向上的前进。

AdaGrad

AdaGrad 通过自适应调整学习率来改善收敛性。每个参数有自己的学习率，随着更新次数的增加而逐渐减小：

$$\theta_{k+1} = \theta_k - \frac{\alpha}{\sqrt{G_k + \epsilon}} \nabla f_{i_k}(\theta_k)$$

其中，G_k 是累积梯度平方和，ϵ 是一个小常数防止除零。

RMSProp

RMSProp 是 AdaGrad 的改进版本，通过引入指数衰减平均限制累积梯度的增长：

$$E[g^2]_k = \gamma E[g^2]_{k-1} + (1-\gamma)(\nabla f_{i_k}(\theta_k))^2$$

$$\theta_{k+1} = \theta_k - \frac{\alpha}{\sqrt{E[g^2]_k + \epsilon}} \nabla f_{i_k}(\theta_k)$$

其中，γ 通常取值为 0.9。

Adam（Adaptive Moment Estimation）

Adam 结合了动量法和 RMSProp 的优点，利用一阶和二阶矩估计来自适应调整学习率：

$$m_{k+1} = \beta_1 m_k + (1-\beta_1)\nabla f_{i_k}(\theta_k)$$

$$v_{k+1} = \beta_2 v_k + (1-\beta_2)(\nabla f_{i_k}(\theta_k))^2$$

$$\hat{m}_{k+1} = \frac{m_{k+1}}{1-\beta_1^{k+1}}, \quad \hat{v}_{k+1} = \frac{v_{k+1}}{1-\beta_2^{k+1}}$$

$$\theta_{k+1} = \theta_k - \frac{\alpha}{\sqrt{\hat{v}_{k+1}} + \epsilon} \hat{m}_{k+1}$$

其中，β_1 和 β_2 分别是动量和 RMSProp 的衰减率，ϵ 是小常数。

小批量 SGD（Mini-batch SGD）

每次使用一小部分样本计算梯度，结合了 BGD 和 SGD 的优点，减少了噪声，提高效率，适合硬件加速和并行计算。

3. SGD 的收敛性与收敛速度比较

SGD 的收敛性

SGD 的收敛性与学习率的设置密切相关。在凸优化问题中，如果学习率 α_k 满足 $\sum_{k=1}^{\infty} \alpha_k = \infty$ 且 $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$，则 SGD 可以收敛到全局最优解。在非凸优化问题中，SGD 也可以收敛到一个局部最优解。

SGD 的收敛路径具有随机性，这使得它在非凸优化中有可能跳出局部最优。然而，这种随机性也导致 SGD 的收敛速度较慢，通常为 $\mathcal{O}(1/\sqrt{K})$，其中 K 为迭代次数。相比之下，BGD 的收敛速度通常为 $\mathcal{O}(1/K)$，这意味着 BGD 在理论上能够更快地逼近最优值，但计算开销更大。

与其他梯度方法的收敛性与 Bound 比较

与批量梯度下降（BGD）和小批量梯度下降（Mini-batch GD）相比，SGD 的主要特点是计算效率高，但收敛速度较慢。

总结

随机梯度下降（SGD）是一种有效的优化算法，广泛应用于大规模数据集和在线学习场景中。通过详细理解其原始理念、迭代过程中的随机性以及多种变种，SGD 在深度学习和机器学习的不同应用中展现了卓越的适应能力和高效性。同时，与其他梯度方法的比较帮助我们理解如何在不同的场景中选择合适的优化算法。

参考文献

- Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
- Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul), 2121-2159.
- Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Tieleman, T., Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.