Stochastic Gradient Descent: Fully Worked Arithmetic, Geometry, and Convergence

Jingwen Feng

October 26, 2025

Abstract

This paper presents stochastic gradient descent in a publishable, graduate level style while preserving a line by line arithmetic trace. We begin with a two parameter linear model and show every residual, gradient, average, and update. A geometric view and a convergence theorem under convexity follow, together with practical schedules and comments on the non convex case. The narrative aligns with the exposition in *Modern Mathematics of Deep Learning* within the Cambridge volume *Mathematical Aspects of Deep Learning*.

Keywords: Stochastic optimization, Empirical risk, Iterate averaging, Convergence, Learning rates

1 Setup

We observe n pairs $z^{(i)} = (x^{(i)}, y^{(i)})$. For parameters $\theta \in \mathbb{R}^p$, model f_θ, and loss $\mathcal{L}(f_\theta, z)$, the empirical risk is

$$\widehat{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\Big(f_\theta, z^{(i)}\Big). \tag{1}$$

A mini batch S_k of size m yields the unbiased stochastic gradient

$$G_k(\theta) = \frac{1}{m}\sum_{i\in S_k} \nabla_\theta\, \mathcal{L}\Big(f_\theta, z^{(i)}\Big), \qquad \mathbb{E}\big[G_k(\theta) \mid \theta\big] = \nabla \widehat{R}(\theta). \tag{2}$$

2 Algorithm

Algorithm 1 SGD with Iterate Averaging

1: **Input:** $\theta^{(0)}$, step sizes $\{\eta_k\}$, batch size m, steps K
2: **for** $k = 1, \ldots, K$ **do**
3: Sample S_k of size m uniformly without replacement
4: $G_k \leftarrow m^{-1} \sum_{i\in S_k} \nabla \mathcal{L}(f_{\theta^{(k-1)}}, z^{(i)})$
5: $\theta^{(k)} \leftarrow \theta^{(k-1)} - \eta_k\, G_k$
6: **end for**
7: Return $\bar{\theta}_K \leftarrow K^{-1} \sum_{k=1}^{K} \theta^{(k)}$

3 Worked example with full arithmetic

We fit $\hat{y} = wx + b$ to $(1, 3)$, $(2, 5)$, $(3, 7)$, $(4, 9)$ from the line $y = 2x + 1$. The per sample loss is $\mathcal{L}(w, b; x, y) = (wx + b - y)^2$ with $\partial\mathcal{L}/\partial w = 2(wx + b - y)x$ and $\partial\mathcal{L}/\partial b = 2(wx + b - y)$. We use mini batches of size two, step size $\eta = 0.1$, and start at $(w^{(0)}, b^{(0)}) = (0, 0)$. Every computation follows.

Step 1 with mini batch $\{(1, 3), (3, 7)\}$

Sample $(1, 3)$ at $(0, 0)$.

$$\hat{y} = 0 \cdot 1 + 0 = 0, \quad r = 0 - 3 = -3, \quad \nabla_w = 2(-3)(1) = -6, \quad \nabla_b = 2(-3) = -6.$$

Sample $(3, 7)$ at $(0, 0)$.

$$\hat{y} = 0 \cdot 3 + 0 = 0, \quad r = 0 - 7 = -7, \quad \nabla_w = 2(-7)(3) = -42, \quad \nabla_b = 2(-7) = -14.$$

Average and update.

$$D_w^{(1)} = \tfrac{-6+(-42)}{2} = -24, \quad D_b^{(1)} = \tfrac{-6+(-14)}{2} = -10,$$

$$(w^{(1)}, b^{(1)}) = (0, 0) - 0.1(-24, -10) = (2.4, 1.0).$$

Step 2 with mini batch $\{(2, 5), (4, 9)\}$

Sample $(2, 5)$ at $(2.4, 1.0)$.

$$\hat{y} = 2.4 \cdot 2 + 1.0 = 4.8 + 1.0 = 5.8, \quad r = 5.8 - 5 = 0.8,$$

$$\nabla_w = 2(0.8)(2) = 3.2, \quad \nabla_b = 2(0.8) = 1.6.$$

Sample $(4, 9)$ at $(2.4, 1.0)$.

$$\hat{y} = 2.4 \cdot 4 + 1.0 = 9.6 + 1.0 = 10.6, \quad r = 10.6 - 9 = 1.6,$$

$$\nabla_w = 2(1.6)(4) = 12.8, \quad \nabla_b = 2(1.6) = 3.2.$$

Average and update.

$$D_w^{(2)} = \tfrac{3.2+12.8}{2} = 8, \quad D_b^{(2)} = \tfrac{1.6+3.2}{2} = 2.4,$$

$$(w^{(2)}, b^{(2)}) = (2.4, 1.0) - 0.1(8, 2.4) = (1.6, 0.76).$$

Step 3 with mini batch $\{(1, 3), (4, 9)\}$

Sample $(1, 3)$ at $(1.6, 0.76)$.

$$\hat{y} = 1.6 \cdot 1 + 0.76 = 1.6 + 0.76 = 2.36, \quad r = 2.36 - 3 = -0.64,$$

$$\nabla_w = 2(-0.64)(1) = -1.28, \quad \nabla_b = 2(-0.64) = -1.28.$$

Sample $(4, 9)$ at $(1.6, 0.76)$.

$$\hat{y} = 1.6 \cdot 4 + 0.76 = 6.4 + 0.76 = 7.16, \quad r = 7.16 - 9 = -1.84,$$

$$\nabla_w = 2(-1.84)(4) = -14.72, \quad \nabla_b = 2(-1.84) = -3.68.$$

Average and update.

$$D_w^{(3)} = \tfrac{-1.28+(-14.72)}{2} = -8, \quad D_b^{(3)} = \tfrac{-1.28+(-3.68)}{2} = -2.48,$$

$$(w^{(3)}, b^{(3)}) = (1.6, 0.76) - 0.1(-8, -2.48) = (2.4, 1.008).$$

Step 4 with mini batch $\{(2,5),(3,7)\}$

Sample $(2,5)$ **at** $(2.4, 1.008)$**.**

$$\hat{y} = 2.4 \cdot 2 + 1.008 = 4.8 + 1.008 = 5.808, \quad r = 5.808 - 5 = 0.808,$$

$$\nabla_w = 2(0.808)(2) = 3.232, \quad \nabla_b = 2(0.808) = 1.616.$$

Sample $(3,7)$ **at** $(2.4, 1.008)$**.**

$$\hat{y} = 2.4 \cdot 3 + 1.008 = 7.2 + 1.008 = 8.208, \quad r = 8.208 - 7 = 1.208,$$

$$\nabla_w = 2(1.208)(3) = 7.248, \quad \nabla_b = 2(1.208) = 2.416.$$

Average and update.

$$D_w^{(4)} = \tfrac{3.232+7.248}{2} = 5.24, \quad D_b^{(4)} = \tfrac{1.616+2.416}{2} = 2.016,$$

$$(w^{(4)}, b^{(4)}) = (2.4, 1.008) - 0.1(5.24, 2.016) = (1.876, 0.8064).$$

Averaging the iterates

$$\bar{w} = \tfrac{2.4+1.6+2.4+1.876}{4} = \tfrac{8.276}{4} = 2.069, \qquad \bar{b} = \tfrac{1.0+0.76+1.008+0.8064}{4} = \tfrac{3.5744}{4} = 0.8936.$$

4 Geometry and intuition

The empirical risk bowl picture explains why noisy steps still trend downhill, while iterate averaging stabilizes the path near the floor of the bowl.

5 Convergence in the convex case

Assume $r : \mathbb{R}^p \to \mathbb{R}$ is convex and differentiable, $\mathbb{E}[G_k \mid \theta^{(k-1)}] = \nabla r(\theta^{(k-1)})$, $\|G_k\| \le L$, and the iterates remain within a ball of radius R around θ^*. With $\eta_k = \eta_0 k^{-1/2}$ and $\bar{\theta}_K = K^{-1} \sum_{k=1}^{K} \theta^{(k)}$,

$$\mathbb{E}\big[r(\bar{\theta}_K)\big] - r(\theta^*) \le RL\, K^{-1/2}. \tag{3}$$

Sketch. Expand $\big\|\theta^{(k)} - \theta^*\big\|^2$, take conditional expectation, use convexity to control $\big\langle \nabla r,\, \theta^{(k-1)} - \theta^* \big\rangle$, and sum.

6 From empirical to population risk

Optimizing $\widehat{R}$ yields gradients that approximate those of the population risk $R(\theta) = \mathbb{E}[\mathcal{L}(f_\theta, Z)]$ when n is large; iterate averaging further suppresses sampling noise.

7 Conclusion

The fully explicit arithmetic trace, the geometric picture, and the convergence guarantee together provide a publishable yet hands on account of SGD.

References

[1] *Modern Mathematics of Deep Learning*. In *Mathematical Aspects of Deep Learning*. Cambridge University Press. Available online at cambridge.org.