

Exercises

§7.8: Stochastic and Mini-Batch Optimization

Python: There is an optional starter script that supports Python-based problem(s) on this sheet. Use it as scaffolding: read the comments, complete the lines marked **TODO**, and compare the output with the mathematics requested in the problem(s).

Problem 1 For $y = (1, 3, 5, 7)$ and $g_p(w) = \frac{1}{2}(w - y_p)^2$, compute all four individual gradients at $w = 2$. Average them and verify that the result equals the full gradient $g'(2)$.

Problem 2 Let I be uniformly distributed over $\{1, \dots, P\}$. Prove that $E[\nabla g_I(\mathbf{w})] = \nabla g(\mathbf{w})$ when $g = (1/P) \sum_p g_p$.

Problem 3 At $w = 4$ for the quadratic example, compute the mini-batch gradients for (a) batch $\{y = 1, y = 3\}$, (b) batch $\{y = 5, y = 7\}$, (c) batch $\{y = 1, y = 7\}$. Why can a mini-batch update move even when the full gradient is zero?

Problem 4 A dataset has $P = 103$ observations and mini-batch size $b = 20$. How many updates occur in one epoch if the final incomplete batch is kept? How many observations are in the final batch?

Problem 5 A dataset contains 50,000 observations. Compare 10 epochs of: (a) full-batch gradient descent, (b) SGD with batch size 1, (c) mini-batch descent with batch size 100. For each method state: the number of parameter updates, and the total number of observation-gradient contributions processed if every observation is used once per epoch.

Problem 6 At the full-batch minimizer $w = 4$ of the quadratic example, SGD selects observation $y = 1$ with $\alpha = 0.1$. Compute the new w . Then calculate the full objective $g(w) = \frac{1}{4} \sum_p \frac{1}{2} (w - y_p)^2$ before and after the update. What does this demonstrate?

Problem 7 Consider three learning-rate schedules: (i) $\alpha_k = \frac{1}{k+1}$, (ii) $\alpha_k = \frac{1}{\sqrt{k+1}}$, (iii) $\alpha_k = 0.1$. For each schedule determine whether $\sum_k \alpha_k$ diverges and whether $\sum_k \alpha_k^2$ converges. Which schedule satisfies both classical stochastic-approximation conditions discussed in class?

Problem 8 Write Python code that: 1. creates

```
y = np.array([1., 3., 5., 7.])
```

2. shuffles the observation indices once per epoch, 3. uses mini-batches of size 2, 4. performs the update for $g_p(w) = \frac{1}{2} (w - y_p)^2$, 5. starts at $w = 0$ with $\alpha = 0.1$, and runs for 20 epochs. Report the final value of w using random seed 0.