Chapter 7: Proximal Gradient Descent

CMPUT 467/567: Machine Learning II

Winter 2026

Proximal operators

- Optimize with smooth $c(\mathbf{w})$ and non-smooth $r(\mathbf{w})$
- To solve $\min_{\mathbf{w}\in\mathbb{R}^d} c(\mathbf{w}) + r(\mathbf{w})$ we break it into two steps
- GD Step: $\tilde{\mathbf{w}}_{t+1} = \mathbf{w}_t - \eta \nabla c(\mathbf{w}_t)$
- Projection step: $\mathbf{w}_{t+1} = \text{prox}_{\eta r}(\tilde{\mathbf{w}}_{t+1})$ where the proximal operator is
- $\text{prox}_{\eta r}(\mathbf{u}) = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w} - \mathbf{u}\|_2^2 + \eta r(\mathbf{w})$

Why so many subscripts in prox?

- $\text{prox}_{\eta r}(\mathbf{u}) = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w}-\mathbf{u}\|_2^2 + \eta r(\mathbf{w})$
- Really, we mean $\text{prox}_{f}(\mathbf{u}) = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w}-\mathbf{u}\|_2^2 + f(\mathbf{w})$ for function f
- In our updates, we have functions like $\lambda\ell_1(\mathbf{w})$ and then have the stepsize out front too, so write
- $\text{prox}_{\eta\lambda\ell_1}(\mathbf{u}) = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w}-\mathbf{u}\|_2^2 + \eta\lambda\ell_1(\mathbf{w})$

Why does projection involve the stepsize?

- The GD step changes $\mathbf{w}$ by stepsize η amount, so we need to apply the projection that amount also to ensure they eventually balance each other out
- Example: $\text{prox}_{\eta\lambda\ell_1}(u) = \begin{cases} u - \eta\lambda & \text{if } u > \eta\lambda \\ u + \eta\lambda & \text{if } u < -\eta\lambda \\ 0 & \text{else} \end{cases}$
- At convergence, for larger $w_{t,j}$, GD step has does $\tilde{w}_{t+1,j} = w_{t,j} + \eta\lambda$ and the proximal update returns $w_{t+1,j} = \tilde{w}_{t+1,j} - \eta\lambda = w_{t,j}$ (no change, done opt)
- And for $w_{t,j}$ more negative, GD step has does $\tilde{w}_{t+1,j} = w_{t,j} - \eta\lambda$ and the proximal update returns $w_{t+1,j} = \tilde{w}_{t+1,j} + \eta\lambda = w_{t,j}$ (no change, done opt)

Why is it useful to break up into two steps?

- Solving $\text{prox}_{\eta r}(\mathbf{u}) = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w}-\mathbf{u}\|_2^2 + \eta r(\mathbf{w})$ with simple loss $\|\mathbf{w}-\mathbf{u}\|_2^2$ is likely a lot simpler than solving $\min_{\mathbf{w}\in\mathbb{R}^d} c(\mathbf{w}) + r(\mathbf{w})$ for arbitrary $c(\mathbf{w})$
- Example, for $r(w) = \ell_1(w)$, can reason about balancing error $(w-u)^2$ versus error $\ell - 1(w) = |w|$.
  - e.g., $u = 0.1$, then at $w = 0$, $(w-u)^2 = 0.01$. If increase w to 0.1, then $(w-u)^2 = 0$ but $|w| = 0.1$. Worse error! In fact, w = 0 is optimal because these small differences to u are squared

Why is it useful to break up into two steps?

- Solving $\text{prox}_{\eta r}(\mathbf{u}) = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w}-\mathbf{u}\|_2^2 + \eta r(\mathbf{w})$ with simple loss $\|\mathbf{w}-\mathbf{u}\|_2^2$ is likely a lot simpler than solving $\min_{\mathbf{w}\in\mathbb{R}^d} c(\mathbf{w}) + r(\mathbf{w})$ for arbitrary $c(\mathbf{w})$
- Example, for $r(w) = \ell_1(w)$, can reason about balancing error $(w-u)^2$ versus error $\ell - 1(w) = |w|$.
- For $c(w)$ the cross-entropy loss, how do we do this reasoning?