Adaptive Learning Methods

Karl Stratos

Last updated: February, 2025

Contents

1 Online Convex Optimization

At step $t = 1, 2, \ldots$, we propose $w_t \in V \subseteq \mathbb{R}^d$ where V is closed and convex. The enemy then chooses a convex and differentiable loss $l_t : \mathbb{R}^d \to \mathbb{R}$ and punishes us by $l_t(w_t) \in \mathbb{R}$. Assuming T such steps, let $u = \arg\min_{w \in V} \sum_{t=1}^T l_t(w)$ denote the best hypothesis in retrospect. We want to upper bound the total "regret" as a function of T:

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq B(T)$$

The goal is to achieve a sublinear regret bound $B(T) = o(T)$, which solves stochastic convex optimization (Appendix A). A lower bound is $\Omega(\sqrt{T})$ (Appendix B).

1.1 Mirror Descent

Let $\hat{l}_t(w) = l_t(w_t) + g_t^\top(w - w_t)$ where $g_t = \nabla l_t(w_t)$. To make the minimum finite, we regularize by the Bregman divergence $D_{\psi_t}(\cdot, w_t)$ (Appendix D) where $\psi_t : V \to \mathbb{R}$ is strictly convex and differentiable. Assuming $\eta_t > 0$, our per-step objective is

$$w_{t+1} = \arg\min_{w \in V} \; \hat{l}_t(w) + \frac{1}{\eta_t} D_{\psi_t}(w, w_t) \tag{1}$$

For instance, (1) becomes gradient descent when $V = \mathbb{R}^d$ and $\psi_t(w) = \frac{1}{2}||w||_2^2$ and exponentiated gradient descent when $V = \Delta^{d-1}$ and $\psi_t(w) = -H(w)$ (Appendix E). Instead of computing (1) directly, we may perform unconstrained minimization and project (aka. online "mirror" descent):

$$\tilde{w}_{t+1} = \arg\min_{w \in \mathbb{R}^d} \; \eta_t \hat{l}_t(w) + D_{\psi_t}(w, w_t) = \nabla \psi_t^*(\nabla \psi_t(w_t) - \eta_t g_t) \tag{2}$$

$$w_{t+1} = \arg\min_{w \in V} \; D_{\psi_t}(w, \tilde{w}_{t+1}) \tag{3}$$

where $\psi_t^* : \mathbb{R}^d \to \mathbb{R}$ is the convex conjugate of ψ_t (Fact F.4). It is easy to show that (1) and (3) are equal.[1]

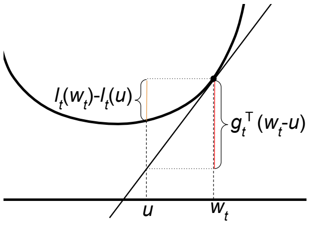
1.2 General Analysis

Since (1) is the Bregman projection of w_t onto V regularized by $\eta_t \hat{l}_t$, the Pythagorean theorem gives us $D_{\psi_t}(u, w_t) + \eta_t \hat{l}_t(u) \geq D_{\psi_t}(w_{t+1}, w_t) + D_{\psi_t}(u, w_{t+1}) + \eta_t \hat{l}_t(w_{t+1})$ (Lemma D.3), or

$$\begin{aligned}
D_{\psi_t}(u, w_t) - D_{\psi_t}(u, w_{t+1}) &\geq \eta_t \hat{l}_t(w_{t+1}) - \eta_t \hat{l}_t(u) + D_{\psi_t}(w_{t+1}, w_t) \\
&\geq \eta_t g_t^\top(w_t - u) + \eta_t g_t^\top(w_{t+1} - w_t) + \frac{\sigma_t}{2} ||w_{t+1} - w_t||_t^2 && (4) \\
&\geq \eta_t g_t^\top(w_t - u) - \eta_t ||g_t||_{t,*} ||w_{t+1} - w_t||_t + \frac{\sigma_t}{2} ||w_{t+1} - w_t||_t^2 && (5) \\
&\geq \eta_t g_t^\top(w_t - u) + \frac{\eta_t^2}{2\sigma_t} ||g_t||_{t,*}^2 && (6) \\
&\geq \eta_t (l_t(w_t) - l_t(u)) + \frac{\eta_t^2}{2\sigma_t} ||g_t||_{t,*}^2 && (7)
\end{aligned}$$

(4) assumes that ψ_t is σ_t-strongly convex with respect to some norm $||\cdot||_t$. (5) uses Hölder's inequality $w^\top v \leq ||w||_t ||v||_{t,*}$ where $||\cdot||_{t,*}$ is the dual norm of $||\cdot||_t$ (Appendix L.2.1). (6) minimizes $J(x) = (\frac{\sigma_t}{2} x^2 - \eta_t ||g_t||_* x)$ over $x \in \mathbb{R}$. Finally, (7) uses the convexity of l_t, i.e.,

$$g_t^\top(w_t - u) \geq l_t(w_t) - l_t(u)$$



[1] Since $f(z) = D_{\psi_t}(z, \tilde{w}_{t+1})$ is strictly convex and differentiable, (3) is the unique point $w^\star$ satisfying $\nabla f(w^\star)^\top(w - w^\star) \geq 0$ for all $w \in V$ (Lemma C.2). Likewise, since $h(z) = \eta_t \hat{l}_t(z) + D_{\psi_t}(z, w_t)$ is strictly convex and differentiable, (1) is the unique point $v^\star$ satisfying $\nabla h(v^\star)^\top(w - v^\star) \geq 0$ for all $w \in V$. But $\nabla f(z) = \nabla\psi_t(z) - \nabla\psi_t(\tilde{w}_{t+1}) = \nabla\psi_t(z) - \nabla\psi_t(w_t) + \eta_t g_t = \nabla h(z)$.

Rearranging, we have a per-step regret bound $l_t(w_t) - l_t(u) \leq \frac{1}{\eta_t}\left(D_{\psi_t}(u, w_t) - D_{\psi_t}(u, w_{t+1})\right) + \frac{\eta_t}{2\sigma_t}\left|\left|g_t\right|\right|_{t,*}^2$. Thus the total regret can be bounded by

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \sum_{t=1}^{T} \frac{1}{\eta_t}\left(D_{\psi_t}(u, w_t) - D_{\psi_t}(u, w_{t+1})\right) + \sum_{t=1}^{T} \frac{\eta_t}{2\sigma_t}\left|\left|g_t\right|\right|_{t,*}^2 \tag{8}$$

1.3 Euclidean Analysis

(8) applies to any Bregman divergence. However, for most practical purposes we use squared Euclidean distance $D_{\psi_t}(x, y) = \frac{1}{2}\left|\left|y - x\right|\right|_{A_t}^2$ weighted by a "preconditioner" matrix $A_t \succ 0$.[2] It is induced by $\psi_t(x) = \frac{1}{2}\left|\left|x\right|\right|_{A_t}^2$ which is 1-strongly convex wrt. $\left|\left|\cdot\right|\right|_{A_t}$, thus the second term of (8) becomes $\frac{1}{2}\sum_{t=1}^{T}\eta_t\left|\left|g_t\right|\right|_{A_t^{-1}}^2$. The first term of the bound (8) is now

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{\eta_t}\left(D_{\psi_t}(u, w_t) - D_{\psi_t}(u, w_{t+1})\right) &= \frac{1}{2}\sum_{t=1}^{T}\frac{1}{\eta_t}\left(\left|\left|w_t - u\right|\right|_{A_t}^2 - \left|\left|w_{t+1} - u\right|\right|_{A_t}^2\right) \\
&= \frac{1}{2}\sum_{t=1}^{T}\left(\frac{1}{\eta_t}\left|\left|w_t - u\right|\right|_{A_t}^2 - \frac{1}{\eta_{t-1}}\left|\left|w_t - u\right|\right|_{A_{t-1}}^2\right) - \frac{1}{\eta_T}\left|\left|w_{T+1} - u\right|\right|_{A_T}^2 \qquad (9) \\
&\leq \frac{1}{2}\sum_{t=1}^{T}\left(\frac{1}{\eta_t}\left|\left|w_t - u\right|\right|_{A_t}^2 - \frac{1}{\eta_{t-1}}\left|\left|w_t - u\right|\right|_{A_{t-1}}^2\right) \\
&= \frac{1}{2}\sum_{t=1}^{T}(w_t - u)^\top\left(\frac{1}{\eta_t}A_t - \frac{1}{\eta_{t-1}}A_{t-1}\right)(w_t - u) \qquad (10)
\end{aligned}$$

(9) uses the dummy variables $\eta_0 = \infty$ and $A_0 = 0_{d\times d}$. To get a general bound, we can apply $v^\top B v \leq \left|\left|v\right|\right|_2\left|\left|Bv\right|\right|_2 \leq \mathrm{tr}\left(B\right)\left|\left|v\right|\right|_2^2$ (i.e., the consistency between the matrix spectral norm and the vector l_2 norm) to (10):

$$\begin{aligned}
\frac{1}{2}\sum_{t=1}^{T}(w_t - u)^\top\left(\frac{1}{\eta_t}A_t - \frac{1}{\eta_{t-1}}A_{t-1}\right)(w_t - u) &\leq \frac{1}{2}\sum_{t=1}^{T}\left(\frac{1}{\eta_t}\mathrm{tr}\left(A_t\right) - \frac{1}{\eta_{t-1}}\mathrm{tr}\left(A_{t-1}\right)\right)\left|\left|w_t - u\right|\right|_2^2 \qquad (11) \\
&\leq \frac{\max_{t=1}^{T}\left|\left|w_t - u\right|\right|_2^2}{2}\underbrace{\sum_{t=1}^{T}\left(\frac{1}{\eta_t}\mathrm{tr}\left(A_t\right) - \frac{1}{\eta_{t-1}}\mathrm{tr}\left(A_{t-1}\right)\right)}_{\text{Telescopes!}} \qquad (12) \\
&= \frac{(\max_{t=1}^{T}\left|\left|w_t - u\right|\right|_2^2)\mathrm{tr}\left(A_T\right)}{2\eta_T} \qquad (13)
\end{aligned}$$

Note that step (12) also requires $\frac{1}{\eta_t}\mathrm{tr}\left(A_t\right) - \frac{1}{\eta_{t-1}}\mathrm{tr}\left(A_{t-1}\right) \geq 0$. To ensure this, we will generally assume that

$$\infty =: \eta_0 \geq \eta_1 \geq \eta_2 \geq \cdots \geq \eta_T > 0 \tag{14}$$

$$0_{d\times d} =: A_0 \prec A_1 \preceq A_2 \preceq \cdots \preceq A_T \tag{15}$$

If $A_t = A$ across t (i.e., time-invariant preconditioning), we can avoid the lossy inequality (11) since

$$\begin{aligned}
\frac{1}{2}\sum_{t=1}^{T}(w_t - u)^\top\left(\frac{1}{\eta_t}A - \frac{1}{\eta_{t-1}}A\right)(w_t - u) &= \frac{1}{2}\sum_{t=1}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\left|\left|w_t - u\right|\right|_A^2 \\
&\leq \frac{\max_{t=1}^{T}\left|\left|w_t - u\right|\right|_A^2}{2}\sum_{t=1}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \\
&= \frac{\max_{t=1}^{T}\left|\left|w_t - u\right|\right|_A^2}{2\eta_T} \qquad (16)
\end{aligned}$$

(16) is d-times sharper than (13) when $A = I_d$. If we further assume that $\eta_t = \eta > 0$ across t (i.e., fixed learning rate), we can make the bound still sharper (recall $\eta_0 = \infty$):

$$\frac{1}{2}\sum_{t=1}^{T}(w_t - u)^\top\left(\frac{1}{\eta_t}A - \frac{1}{\eta_{t-1}}A\right)(w_t - u) = \frac{\left|\left|w_1 - u\right|\right|_A^2}{2\eta} \tag{17}$$

[2] For $A \succ 0$, $\left|\left|x\right|\right|_A = \sqrt{x^\top A x}$ is a norm on $\mathbb{R}^d$ with $\left|\left|x\right|\right|_{A^{-1}}$ as the dual norm (Appendix L.3). For the sake of simplicity, we will assume that precondioners are positive-definite (e.g., add an infinitesimal value to the diagonal).

1.3.1 Algorithm and guarantee

Since $\nabla \psi_t(x) = A_t x$ and $\nabla \psi_t^*(x) = A_t^{-1} x$, we update $\tilde{w}_{t+1} = \nabla \psi_t^*(\nabla \psi_t(w_t) - \eta_t g_t) = w_t - \eta_t A_t^{-1} g_t$. In summary, we start from some initial hypothesis $w_1 \in V \subseteq \mathbb{R}^d$ (with dummy $\eta_0 = \infty$ and $A_0 = 0_{d \times d}$) and for $t = 1, 2, \ldots$

- The enemy picks a convex and differentiable loss $l_t : \mathbb{R}^d \to \mathbb{R}$ and we suffer $l_t(w_t) \in \mathbb{R}$.
- We compute the gradient $g_t = \nabla l_t(w_t)$.
- We pick a learning rate $\eta_t \leq \eta_{t-1}$ and a preconditioner $A_t \succeq A_{t-1}$.
- We compute $\tilde{w}_{t+1} = w_t - \eta_t A_t^{-1} g_t \in \mathbb{R}^d$.
- We project $w_{t+1} = \arg\min_{w \in V} \; D_{\psi_t}(w, \tilde{w}_{t+1}) \in V$.

Let $D_A = \max_{t=1}^T ||w_t - u||_A$ and $D_{A,1} = ||w_1 - u||_A$; for the special case $A = I_d$, let $D = \max_{t=1}^T ||w_t - u||_2$ and $D_1 = ||w_1 - u||_2$. After T such steps, we guarantee by (8) that $w_1 \ldots w_T \in V$ satisfy

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq \frac{D^2 \mathrm{tr}\,(A_T)}{2\eta_T} + \frac{1}{2} \sum_{t=1}^T \eta_t \, ||g_t||_{A_t^{-1}}^2 \qquad \text{(always)} \qquad (18)$$

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq \frac{D_A^2}{2\eta_T} + \frac{1}{2} \sum_{t=1}^T \eta_t \, ||g_t||_{A^{-1}}^2 \qquad \text{(if } A_t = A\text{)} \qquad (19)$$

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq \frac{D_{A,1}^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^T ||g_t||_{A^{-1}}^2 \qquad \text{(if } A_t = A \text{ and } \eta_t = \eta\text{)} \qquad (20)$$

In principle, D_A can grow as $O(T)$. This issue is usually addressed by assuming $V \subseteq \mathbb{R}^d$ to have a finite diameter $\Delta = \max_{x,y \in V} \; ||x - y||_A$ so that $D_A \leq \Delta$ is constant in T. But this is not a solution, and in practice $V = \mathbb{R}^d$ almost always (i.e., no projection). Thus we will assume $V = \mathbb{R}^d$ and treat D_A as constant in T.

2 Stochastic Gradient Descent (SGD)

SGD uses no preconditioning (i.e., $A = I_d$) and specifies the update

$$w_{t+1} = w_t - \eta_t g_t \qquad (21)$$

corresponding to constant regularization $D_\psi(x, y) = \frac{1}{2} ||y - x||_2^2$ in Euclidean space with $\psi(x) = \frac{1}{2} ||x||_2^2$. As a special case of mirror descent, it satisfies the regret bound (19) (or (20) if $\eta_t = \eta > 0$ is constant). But the derivation becomes particularly simple, so we give one below. For any t:

$$||w_t - u||_2^2 - ||w_{t+1} - u||_2^2 = ||w_t - u||_2^2 - ||w_t - u - \eta_t g_t||_2^2 = 2\eta_t \underbrace{g_t^\top (w_t - u)}_{\geq l_t(w_t) - l_t(u)} - \eta_t^2 \, ||g_t||_2^2$$

yielding the per-step bound $l_t(w_t) - l_t(u) \leq \frac{1}{2\eta_t} \left(||w_t - u||_2^2 - ||w_{t+1} - u||_2^2 \right) + \frac{\eta_t}{2} \, ||g_t||_2^2$ that telescopes and gives us (19) (assuming $\eta_t \leq \eta_{t-1}$) and (20). We have skipped the use of the Pythagorean theorem (which holds exactly in this case, check the conditions in Lemma D.3) and other inequalities in (4–6).

2.1 Analysis

Assume a bound on the gradient norm $L \geq \max_{t=1}^T ||g_t||_2$ (treated as constant in T). If $\eta_t = \eta$, we have from (20)

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq \frac{D_1^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^T ||g_t||_2^2 \leq \frac{D_1^2}{2\eta} + \frac{\eta}{2} L^2 T \qquad (22)$$

It is clear that choosing $\eta = \frac{1}{\sqrt{T}}$ makes the bound $O(\sqrt{T})$, but we can explicitly minimize it over η. The minimizer is $\eta^\star = \frac{D_1}{L\sqrt{T}}$ (not practical since D_1 and L are unknown), yielding

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq D_1 L \sqrt{T} \qquad (23)$$

If T is unknown, we can use the per-step learning rate $\eta_t = \frac{1}{\sqrt{t}}$. Since it satisfies $\eta_t \leq \eta_{t-1}$, we have from (19)

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \frac{D^2}{2\eta_T} + \frac{1}{2}\sum_{t=1}^{T} \eta_t \left|\left|g_t\right|\right|_2^2 \leq \frac{D^2}{2}\sqrt{T} + \frac{L^2}{2}\underbrace{\left(\sum_{t=1}^{T}\frac{1}{\sqrt{t}}\right)}_{\leq 2\sqrt{T}} \leq \left(\frac{D^2+2L^2}{2}\right)\sqrt{T} \tag{24}$$

where the inequality $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}$ is a special case of the following fact.

Fact 2.1. Let $x_1 \ldots x_T \in \mathbb{R}^d$ be any vectors and let $X_t = \sum_{l=1}^{t} x_l x_l^\top \succeq 0$. Then

$$\sum_{t=1}^{T} x_t^\top X_t^{-1/2} x_t \leq 2\mathrm{tr}\left(X_T^{1/2}\right)$$

If $d = 1$, it can be stated as $\sum_{t=1}^{T}\frac{b_t}{\sqrt{B_t}} \leq 2\sqrt{B_T}$ for any nonnegative $b_1 \ldots b_T \geq 0$ where $B_t = \sum_{l=1}^{t} b_l \geq 0$.

Proof of Fact 2.1. $\mathrm{tr}\left(X^{1/2}\right) \in \mathbb{R}$ is concave in $X \succeq 0$ with gradient $\frac{1}{2}X^{-\top/2}$. Thus $\mathrm{tr}\left(A^{1/2}\right) \leq \mathrm{tr}\left(B^{1/2}\right) + \mathrm{tr}\left(\frac{1}{2}B^{-1/2}(A-B)\right)$ for any $B \succeq A$, or $\mathrm{tr}\left(B^{1/2}\right) - \mathrm{tr}\left(A^{1/2}\right) \geq \frac{1}{2}\mathrm{tr}\left(B^{-1/2}(B-A)\right)$. Since $X_t = X_{t-1} + x_t x_t^\top$, we have $\mathrm{tr}\left(X_t^{1/2}\right) - \mathrm{tr}\left(X_{t-1}^{1/2}\right) \geq \frac{1}{2}\mathrm{tr}\left(X_t^{-1/2}x_t x_t^\top\right) = \frac{1}{2}x_t^\top X_t^{-1/2} x_t$. Summing both sides over t gives the statement. □

3 AdaGrad

SGD is potentially inefficient because it does not use the gradient information. Avoid the lossy second inequality in (22) and directly minimize the first bound $\frac{D_1^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\left|\left|g_t\right|\right|_2^2$ over η. This yields a potentially much larger learning rate $\eta^\star = \frac{D_1}{\sqrt{\sum_{t=1}^{T}||g_t||_2^2}} \gg \frac{D_1}{L\sqrt{T}}$ and a tighter bound

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq D_1\sqrt{\sum_{t=1}^{T}\left|\left|g_t\right|\right|_2^2} \tag{25}$$

assuming $\sum_{t=1}^{T}\left|\left|g_t\right|\right|_2^2 \ll (\max_{t=1}^{T}\left|\left|g_t\right|\right|_2^2)T$ (i.e., big gradients are outliers). The learning rate requires the knowledge of all gradients $g_1 \ldots g_T$, so it can only be set in hindsight. But we can use the *partial sum*:

$$\eta_t = \frac{D}{\sqrt{\sum_{l=1}^{t}\left|\left|g_l\right|\right|_2^2}} \tag{26}$$

Since it satisfies $\eta_t \leq \eta_{t-1}$, in a complete analogy to (24):

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \frac{D^2}{2\eta_T} + \frac{1}{2}\sum_{t=1}^{T}\eta_t\left|\left|g_t\right|\right|_2^2 \leq \frac{D}{2}\sqrt{\sum_{t=1}^{T}\left|\left|g_t\right|\right|_2^2} + \frac{D}{2}\underbrace{\left(\sum_{t=1}^{T}\frac{\left|\left|g_t\right|\right|_2^2}{\sqrt{\sum_{l=1}^{t}\left|\left|g_l\right|\right|_2^2}}\right)}_{\leq 2\sqrt{\sum_{t=1}^{T}||g_t||_2^2}} \leq \frac{3}{2}\left(D\sqrt{\sum_{t=1}^{T}\left|\left|g_t\right|\right|_2^2}\right) \tag{27}$$

which is only ≈ 1.5 times worse than the oracle bound (25) (assuming $D \approx D_1$). We can scale (26) by $\frac{\sqrt{2}}{2}$ to slightly improve the constant to ≈ 1.4.

We can similarly find the optimal preconditioner in hindsight and work backward. We assume the update $w_{t+1} = w_t + A^{-1}g_t$ (i.e., $\eta = 1$) where $A \succ 0$ has absorbed any fixed learning rate. Denoting $\delta_1 = w_1 - u \in \mathbb{R}^d$ and $O_T = \sum_{t=1}^{T} g_t g_t^\top \succ 0$, we may consider minimizing (20) which is equivalent to

$$A^\star = \underset{A \in \mathbb{R}^{d\times d}:\ A \succeq 0}{\arg\min}\ \underbrace{\delta_1^\top A \delta_1 + \mathrm{tr}\left(A^{-1}O_T\right)}_{J(A)} \tag{28}$$

This is a proper convex problem. J is bounded below by 0 (both terms are nonnegative). J is convex (the second term is well known to be convex over $A \succ 0$). The feasible set of PSD matrices is closed and convex. Thus an infimum exists. However, no $A^\star \succeq 0$ attains that infimum. $A^\star$ cannot be a boundary point (i.e., has some zero eigenvalues) since then $J(A^\star)$ is undefined. For $A^\star \succ 0$, we must have $\langle \nabla J(A^\star), A - A^\star \rangle_F \geq 0$ for all $A \succeq 0$ (Lemma C.3) which means $\nabla J(A^\star) = 0_{d \times d}$. But this condition is

$$\delta_1 \delta_1^\top = (A^\star)^{-1} O_T (A^\star)^{-1}$$

which is impossible due to a rank mismatch. This implies that while a limit on a series of increasingly degenerate $A \succ 0$ achieves the infimum, the minimizer (28) does not exist.

3.1 Diagonal Preconditioner

AdaGrad (Duchi *et al.*, 2011) dramatically simplifies (28) by constraining the preconditioner to be *diagonal*, i.e., $A = \operatorname{diag}(a_1 \ldots a_d)$ where $a_i > 0$. With this restriction, (28) decomposes over dimensions:

$$a_1^\star \ldots a_d^\star = \operatorname*{arg\,min}_{a_1 \ldots a_d \geq 0} \sum_{i=1}^d \left(\delta_{1,i}^2 a_i + \frac{1}{a_i} \sum_{t=1}^T g_{t,i}^2 \right) \tag{29}$$

The objective is convex in each $a_i > 0$. The stationary condition implies the closed form solution

$$a_i^\star = \frac{1}{|\delta_{1,i}|} \sqrt{\sum_{t=1}^T g_{t,i}^2} \tag{30}$$

Plugging $A^\star = \operatorname{diag}(a_1^\star \ldots a_d^\star)$ in (20), we have the minimized bound

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq \sum_{i=1}^d |\delta_{1,i}| \sqrt{\sum_{t=1}^T g_{t,i}^2} \tag{31}$$

It is instructive to compare this to (25). Letting $\alpha_i = |\delta_{1,i}|$ and $\beta_i = \sqrt{\sum_{t=1}^T g_{t,i}^2}$, by Hölder's inequality

$$\sum_{i=1}^d |\delta_{1,i}| \sqrt{\sum_{t=1}^T g_{t,i}^2} = \alpha^\top \beta \leq ||\alpha||_2 \, ||\beta||_2 = D_1 \sqrt{\sum_{t=1}^T ||g_t||_2^2}$$

The equality holds iff $\alpha = \lambda \beta$ for some $\lambda > 0$ (i.e., $A^\star = \lambda I_d$). Otherwise, (31) may be much tighter than (25).

3.1.1 Per-step preconditioner

At step $t \leq T$, we again use the partial sum. For instance, if we set $A_{t,i,i} = \frac{1}{D} \sqrt{\sum_{l=1}^t g_{l,i}^2}$, we have $A_t \succeq A_{t-1}$ and can straightforwardly use (18) and Fact 2.1 to bound the regret as

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq \frac{3D}{2} \sum_{i=1}^d \sqrt{\sum_{t=1}^T g_{t,i}^2}$$

We can also argue for a tighter bound using $A_{t,i,i} = \frac{1}{\delta_i} \sqrt{\sum_{l=1}^t g_{l,i}^2}$ where $\delta_i = \max_{t=1}^T |w_{t,i} - u_i|$. The idea is to treat $A_{t,i,i}^{-1} = (26)$ as a "learning rate" for each dimension i for which we have the bound (27). We can decompose the bound using the basic fact that "a convex regret is upper bounded by the linearized regret". Formally,

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq \sum_{t=1}^T g_t^\top w_t - g_t^\top u = \sum_{i=1}^d \left(\sum_{t=1}^T g_{t,i} w_{t,i} - g_{t,i} u_i \right) \leq \frac{3}{2} \sum_{i=1}^d \delta_i \sqrt{\sum_{t=1}^T g_{t,i}^2} \tag{32}$$

where the second inequality treats $\sum_{t=1}^T g_{t,i} w_{t,i} - g_{t,i} u_i$ as the regret for the 1-dimensional (linear) losses $l_{t,i}(w_{t,i}) = g_{t,i} w_{t,i}$ for each i. This is only ≈ 1.5 times worse than (31) (assuming $|\delta_{1,i}| \approx \delta_i$).

3.2 Full Preconditioner

We can "force" a non-diagonal minimizer in (28) by regularizing the trace:

$$A^\star = \underset{A \succ 0}{\arg\min}\ \delta_1^\top A \delta_1 + \mathrm{tr}\left(A^{-1} O_T\right) \approx \underset{A \succ 0:\ \mathrm{tr}(A) \leq c}{\arg\min}\ \mathrm{tr}\left(A^{-1} O_T\right) = \frac{c}{\mathrm{tr}(O_T^{1/2})} O_T^{1/2} \tag{33}$$

(see Lemma C.3 in this note for the closed-form solution). Plug in $A^\star = O_T^{1/2}$ in (18) with $\eta_t = D$ to have

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq D\ \mathrm{tr}\left(O_T^{1/2}\right) \tag{34}$$

Compare this with plugging in the diagonal counterpart $A^\star_{i,i} = \sqrt{\sum_{t=1}^T g_{t,i}^2}$ in (18) which yields

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq D \sum_{i=1}^d \sqrt{\sum_{t=1}^T g_{t,i}^2} \tag{35}$$

Though they look similar, (34) is smaller than (35) unless O_T is diagonal (Lemma C.4). Intuitively, $O_T^{1/2}$ exploits the interplay between dimensions while $\sum_{i=1}^d \sqrt{\sum_{t=1}^T g_{t,i}^2}$ does not. In particular, (34) is $O(\sqrt{T})$ (since (35) is).

3.2.1 Per-step preconditioner

At step $t \leq T$, let $O_t = \sum_{l=1}^t g_l g_l^\top$ and use $A_t = O_t^{1/2}$. Assuming the constant learning rate $\eta_t = D$, (18) becomes

$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq \frac{D\mathrm{tr}\left(O_T^{1/2}\right)}{2} + \frac{D}{2} \sum_{t=1}^T g_t^\top O_t^{-1/2} g_t \leq \frac{D\mathrm{tr}\left(O_T^{1/2}\right)}{2} + D\mathrm{tr}\left(O_T^{1/2}\right) = \frac{3D}{2}\ \mathrm{tr}\left(O_T^{1/2}\right) \tag{36}$$

where the second inequality uses Fact 2.1. Again, we conclude that the regret bound is only 1.5 worse when using a per-step preconditioner (i.e., compared to (34)).

3.3 AdaGrad in Practice

The full AdaGrad preconditioner $A_t = (\sum_{l=1}^t g_l g_l^\top)^{1/2} \in \mathbb{R}^{d \times d}$ is unfortunately impractical for any large d, so we use the diagonal preconditioner where $A_{t,i,i} = \sqrt{\sum_{l=1}^t g_{l,i}^2}$. Then the update $w_{t+1} = w_t + \eta_t A_t^{-1} g_t$ is equivalent to per-parameter adaptive learning rates:

$$w_{t+1,i} = w_{t,i} - \frac{\eta_t}{\sqrt{\sum_{l=1}^t g_{l,i}^2}} g_{t,i} \tag{37}$$

where the learning rate shrinks based on how heavily the parameter has been updated in the past. Note that $w_{2,i} = w_{1,i} - \eta_1$ and the magnitude of the update is always at most η_t.

4 Adam

AdaGrad has inspired a whole class of per-parameter adaptive updates. One practical issue of AdaGrad is that the update can only become smaller throughout training because the denominator in (37) can only become larger. This is fine for convex problems where there is only one local optimum, but we may want to allow the update to jump back in size for nonconvex problems. One way to address this issue is by "forgetting" the far past. We may only use the past $K < t$ steps at step t to compute the denominator. Better, we may use an exponential moving average (EMA):

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

where $v_0 = 0_d$ and $\beta_2 \in [0, 1)$ is a coefficient (i.e., how much to remember). If we view g_t as iid random variables with mean $g \in \mathbb{R}^d$, we have $\mathbf{E}[v_t] = (1 - \beta_2^t) g^2$ (easy to prove by induction) which converges to the true second moment g^2 as $t \to \infty$. While at it, we can use momentum for the gradient itself which is well known to help in making SGD more stable:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

for $m_0 = 0_d$ and $\beta_1 \in [0, 1)$. Again, $\mathbf{E}[m_t] = (1 - \beta_1^t) g \to g$ as $t \to \infty$. Replacing g_t and $\sum_{l=1}^t g_{l,i}^2$ in (37) with m_t and v_t, we have **RMSProp** with momentum (Tieleman *et al.*, 2012; Graves, 2013):

$$w_{t+1} = w_t - \eta_t \frac{m_t}{\sqrt{v_t}} \tag{38}$$

Adam (Kingma and Ba, 2014) observes that since β_1, β_2 are typically close to 1, the initial updates will be small until they gain some momentum. But since

$$g = \frac{1}{1 - \beta_1^t} \mathbf{E}\left[m_t\right] \qquad\qquad g^2 = \frac{1}{1 - \beta_2^t} \mathbf{E}\left[v_t\right]$$

we can use $\bar{m}_t = \frac{1}{1-\beta_1^t} m_t$ and $\bar{v}_t = \frac{1}{1-\beta_1^t} v_t$ to correct the bias where $\mathbf{E}[\bar{m}_t] = g$ and $\mathbf{E}[\bar{v}_t] = g^2$. This yields the Adam update

$$w_{t+1} = w_t - \eta_t \frac{\bar{m}_t}{\sqrt{\bar{v}_t}} \tag{39}$$

Using the second moment $\mathbf{E}[g_t^2]$ instead of the sum $\sum_t g_t^2$ is a fundamental departure from AdaGrad. In this case, the preconditioner can be seen as a flawed approximation of the Hessian/Fisher matrix, relating Adam to Netwon's method and natural gradient descent (Appendix I).

4.1 Scale Invariance

A property of any AdaGrad-style update like Adam and RMSProp is scale invariance: the gradient can be scaled by an arbitrary constant without changing the update. More specifically, we can multiply all gradients g_t elementwise by some $c \in \mathbb{R}^d$ in Adam and have

$$w_{t+1} = w_t - \eta_t \frac{\operatorname{diag}(c)\, \bar{m}_t}{\sqrt{\operatorname{diag}(c)^2\, \bar{v}_t}} = w_t - \eta_t \frac{\bar{m}_t}{\sqrt{\bar{v}_t}}$$

Scale invariance is suspected to be important for training deep networks. The gradient of a linear function $c^\top w$ with respect to w is c, which may blow up or shrivel in top layers. With vanilla SGD, weights at the top layer may receive either huge or tiny updates compared to ones at the bottom layer. With scale invariant methods like Adam, weights at either layer will learn at a similar pace. Note that Adam is implemented with smoothing in practice: $w_{t+1,i} = w_{t,i} - \frac{\eta_t}{\sqrt{c_i^2 \bar{v}_{t,i} + \epsilon}} c_i \bar{m}_{t,i}$. It is not scale invariant for $\epsilon > 0$. But since ϵ is typically minuscule, scale invariance is approximately preserved (Zhuang *et al.*, 2022).

4.2 Convergence

In the proof of AdaGrad's convergence, we use the fact that the learning rate is nonincreasing. This is no longer true in momentum-based updates like RMSProp and Adam. In fact, there is a problem for which Adam does not converge (i.e., it has a linear regret) (Reddi *et al.*, 2019). One way to enforce a nonincreasing learning rate in Adam is to take the elementwise max $\bar{v}_t^{\max} = \max\left\{\bar{v}_{t-1}^{\max}, \bar{v}_t\right\}$ where $\bar{v}_0^{\max} = 0_d$ and use $\bar{v}_t^{\max}$ in place of $\bar{v}_t$ in the update (39) (AMSGrad). Adam with AMSGrad now has convergence guarantees and is indeed able to converge in synthetic examples that vanilla Adam spectacularly fails to converge in (Figure 1 in their paper). In practice, however, AMSGrad does not seem to make a whole lot of difference in downstream performance (see this blog).

4.3 Weight Decay

Hanson and Pratt (1988) originally proposed weight decay $w \leftarrow (1 - \lambda) w$ as a way of regularizing the model size independently of the loss. In SGD, it coincides with l_2 regularization. Even here, there is an important caveat: the decay factor must be coupled with the learning rate.

$$w_{t+1} = \underbrace{w_t - \eta_t \nabla \left(l_t(w_t) - \frac{\lambda'}{2} \left|\left|w_t\right|\right|_2^2 \right)}_{\text{SGD with } l_2 \text{ regularization}} = \underbrace{w_t - \eta_t \nabla l_t(w_t) - \eta_t \lambda' w_t}_{\text{SGD with decay factor } \lambda = \eta_t \lambda'}$$

With pre-conditioning they do not coincide.

$$w_{t+1} = w_t - \eta_t A^{-1} \nabla \left(l_t(w_t) - \frac{\lambda'}{2} ||w_t||_2^2 \right) = w_t - \eta_t A^{-1} \nabla l_t(w_t) - \lambda' \eta_t A^{-1} w_t$$

where the last term is not equal to λw_t for any $\lambda > 0$ unless $A = cI_d$ is spherical. Loshchilov and Hutter (2017) thus propose to perform explicit weight decay on top of Adam, denoted as **AdamW**. The original AdamW paper describes coupled weight decay, which is presumably the reason that standard libraries multiply the decay factor with the learning rate (e.g., PyTorch). However, Wortsman *et al.* (2024) find that fully decoupling the learning rate and the decay factor makes training less sensitive to the choice of learning rate. Specifically, they use

$$w_{t+1} = w_t - s_t \left(\eta_{\max} \frac{\bar{m}_t}{\sqrt{\bar{v}_t}} - \lambda_{\max} w_t \right)$$

where $\eta_{\max}$ and $\lambda_{\max}$ are the maximum learning rate and decay factor, and $s_t \in [0, 1]$ is a schedule multiplier. The schedule typically "warms up" for T_{warmup} steps to 1, then "cools down" to some small final value. Thus the update has the form $w_{t+1} = w_t - \eta_t \frac{\bar{m}_t}{\sqrt{\bar{v}_t}} - \lambda_t w_t$ with the stepwise $\eta_t = s_t \eta_{\max}$ and $\lambda_t = s_t \lambda_{\max}$. In contrast, coupled weight decay has the form $w_{t+1} = w_t - \eta_t (\frac{\bar{m}_t}{\sqrt{\bar{v}_t}} - \lambda w_t)$.

4.4 Full Algorithm

```
AdamW
Input:
• Initial parameter value $w_1 \in \mathbb{R}^d$
• Loss functions $l_1, l_2, \ldots, l_T : \mathbb{R}^d \to \mathbb{R}$ ($l_t$ corresponds to the loss on the $t$-th minibatch)
• Schedule $s_1, s_2, \ldots, s_T \in [0, 1]$
• Maximum learning rate $\eta_{\max} > 0$ and weight decay factor $\lambda_{\max} \geq 0$
• Momentum coefficients $(\beta_1, \beta_2)$, smoothing coefficient $\epsilon \geq 0$, flag for using AMSGrad

  1. Initialize the first and second momentum estimates $(m_0, v_0, \bar{v}_0^{\max}) \leftarrow (0_d, 0_d, 0_d)$.
  2. For $t = 1 \ldots T$:
      (a) Do backprop and compute the gradient $g_t \leftarrow \nabla l_t(w_t)$.
      (b) Compute the bias-corrected EMA estimates for the gradient and squared gradient:
              $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$                $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
              $\bar{m}_t \leftarrow \frac{1}{1 - \beta_1^t} m_t$                $\bar{v}_t \leftarrow \frac{1}{1 - \beta_2^t} v_t$
      (c) $\hat{v}_t \leftarrow \bar{v}_t^{\max}$ where $\bar{v}_t^{\max} \leftarrow \max\{\bar{v}_{t-1}^{\max}, \bar{v}_t\}$ if AMSGrad, else $\hat{v}_t \leftarrow \bar{v}_t$
      (d) Compute the per-parameter update for $i = 1 \ldots d$:
              $w_{t+1,i} \leftarrow w_{t,i} - s_t \left( \frac{\eta_{\max}}{\sqrt{\hat{v}_{t,i}} + \epsilon} \bar{m}_{t,i} - \lambda_{\max} w_{t,i} \right)$
  3. Return $w_{T+1} \in \mathbb{R}^d$
```

The hyperparameters affect each other and need to be tuned jointly for the given model and dataset (e.g., a longer warmup allows for a larger value of effective $\eta_{\max}$). There are many techniques specifically designed for large-scale training. One example is "annealing", in which the final phase of training is performed on very high quality data with a schedule that linearly decays to 0. An average of the weights during annealing is used as the final model (Dubey *et al.*, 2024).

5 Approximations of Adam

A practical issue with Adam is that it requires maintaining the first/second gradient moments $m, v \in \mathbb{R}^d$. For instance, if $w \in \mathbb{R}^d$ are in `bfloat16`, maintaining m, v in `float32` increases the memory requirement for optimization from $4d$ to $12d$ bytes (excluding other overheads in backpropagation). Many works attempt to cut down the memory usage for v.

5.1 Adafactor

Adafactor (Shazeer and Stern, 2018) assumes that weights are organized as matrices $W \in \mathbb{R}^{m \times n}$ (e.g., layers) and uses a low-rank approximatation of the corresponding second moment $V \in \mathbb{R}^{m \times n}$. If there are $\ll d$ weight matrices, this effectively makes the memory overhead $O(1)$. The usual Adam update has the form (for matrix weights)

$$W_{t+1} = W_t - \eta \frac{M_t}{\sqrt{V_t}}$$

where $M_t \approx \mathbf{E}[G_t]$ and $V_t \approx \mathbf{E}[G_t^2] = G^2$ for the stochastic gradient $G_t \in \mathbb{R}^{m \times n}$. Adafactor instead proposes to perform

$$W_{t+1} = W_t - \eta \frac{M_t}{\sqrt{A_t B_t}}$$

where $A_t \in \mathbb{R}^{m \times r}$ and $B_t \in \mathbb{R}^{r \times n}$ are low-rank matrices such that $\mathbf{E}[A_t B_t] \approx G^2$. Practical considerations impose certain constraints: (1) A_t, B_t need to be updatable in an online fashion, (2) they are (ideally) strictly positive since we will divide by their square roots. This makes SVD difficult to use (though it yields an optimal solution in Frobenius norm) since it does not decompose over matrix additions and can be negative. The problem is more naturally approached as nonnegative matrix factorization (NMF) (Appendix K). It is well known that the following rank-1 NMF objective

$$a^\star, b^\star \in \underset{a \in \mathbb{R}^m_{\geq 0}, b \in \mathbb{R}^n_{\geq 0}}{\arg\min} \quad \mathrm{IDiv}(G^2, ab^\top)$$

has the solution space of $a^\star (b^\star)^\top = \frac{G^2 1_n 1_m^\top G^2}{1_m^\top G^2 1_n}$ (e.g., $a^\star = G^2 1_n$ and $b^\star = \frac{(G^2)^\top 1_m}{1_m^\top a^\star}$). To derive an online update, Adafactor maintains the EMA (with $a_0 = 0_m$ and $s_0 = 0_n$):

$$a_t = \beta a_{t-1} + (1-\beta) G_t^2 1_n \qquad \bar{a}_t = \frac{1}{1-\beta^t} a_t \qquad \widehat{V}_t = \frac{\bar{a}_t \bar{s}_t^\top}{1_m^\top \bar{a}_t} = \left(\frac{1}{1-\beta^t}\right) \frac{a_t s_t^\top}{1_m^\top a_t}$$

$$s_t = \beta s_{t-1} + (1-\beta)(G_t^2)^\top 1_m \qquad \bar{s}_t = \frac{1}{1-\beta^t} s_t$$

and uses $\widehat{V}_t$ to approximate the second moment.[3] While the rank-1 contraint can be limiting, it is easy to derive a rank-r generalization of Adafactor using EM (Appendix K.2).

5.2 TODO: Adam-mini

6 Shampoo

AdaGrad shows that the best we can do is $w_{t+1} = w_t - \eta O_t^{-1/2} g_t$ where $O_t = \sum_{l=1}^t g_l g_l^\top \in \mathbb{R}^{d \times d}$. But this incurs $O(d^3)$ compute overhead (i.e., to invert a $d \times d$ matrix). Instead of resorting to a diagonal appoximation, **Shampoo** (Gupta *et al.*, 2018) proposes a clever middle ground by assuming the hypothesis space $\mathbb{R}^{m \times n}$ of matrices. At step t, we propose $W_t \in \mathbb{R}^{m \times n}$ and receive a loss $l_t(W_t) \in \mathbb{R}$ where $l_t : \mathbb{R}^{m \times n} \to \mathbb{R}$ is convex and differentiable. Let $G_t = \nabla l_t(W_t) \in \mathbb{R}^{m \times n}$ denote the per-step gradient. Shampoo prescribes

$$W_{t+1} = W_t - \eta \underbrace{L_t^{-1/4}}_{m \times m} \underbrace{G_t}_{m \times n} \underbrace{R_t^{-1/4}}_{n \times n} \qquad L_t = \sum_{l=1}^t G_l G_l^\top \qquad R_t = \sum_{l=1}^t G_l^\top G_l \tag{40}$$

The compute overhead is now $O(m^3 + n^3)$ which is much smaller than the full conditioning overhead $O(m^3 n^3)$. For analysis, we can convert (40) to an equivalent standard form by the usual properties of Kronecker product (Appendix G),

$$w_{t+1} = w_t - \eta \underbrace{\left(L_t^{1/4} \otimes R_t^{1/4}\right)^{-1}}_{mn \times mn} \underbrace{g_t}_{mn \times 1} \tag{41}$$

[3] Note that this is a biased estimator of the optimal rank-1 decomposition since a_t and s_t are correlated. That is, $\mathbf{E}[\widehat{V}_t] = \mathbf{E}[\frac{\bar{a}_t \bar{s}_t^\top}{1_m^\top \bar{a}_t}] \neq \frac{\mathbf{E}[\bar{a}_t]\mathbf{E}[\bar{s}_t]^\top}{1_m^\top \mathbf{E}[\bar{a}_t]} = \frac{G^2 1_n 1_m^\top G^2}{1_m^\top G^2 1_n} = a^\star (b^\star)^\top$.

where $g_t = \overline{\text{vec}}(G_t)$ and $w_t = \overline{\text{vec}}(W_t)$. Thus Shampoo is “just” Euclidean mirror descent with the per-step preconditioner $A_t = L_t^{1/4} \otimes R_t^{1/4}$. Since $L_t \succeq L_{t-1}$ and $R_t \succeq R_{t-1}$, we also have $A_t \succeq A_{t-1}$ (see (75)). The obvious intuition is that $A_t \approx O_t^{1/2}$. In particular, we can show that

$$O_t^{1/2} \preceq \sqrt{r}(L_t^{1/4} \otimes R_t^{1/4}) = \sqrt{r} A_t \tag{42}$$

where $r = \max_t \text{rank}\,(G_t)$.[4] Since the vectorized losses $l_t : \mathbb{R}^{mn} \to \mathbb{R}$ (trivially) remain convex and differentiable and $A_t \succeq A_{t-1}$, we can use (18) to bound the regret:

$$\begin{aligned}
\sum_{t=1}^{T} l_t(w_t) - l_t(u) &\leq \frac{D^2 \text{tr}\,(A_T)}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} g_t^\top A_t^{-1} g_t \\
&\leq \frac{D^2 \text{tr}\,(A_T)}{2\eta} + \frac{\eta\sqrt{r}}{2} \sum_{t=1}^{T} g_t^\top O_t^{-1/2} g_t && \text{(since } A_t^{-1} \preceq \sqrt{r} G^{-1/2} \text{ by (42))} \\
&\leq \frac{D^2 \text{tr}\,(A_T)}{2\eta} + \eta\sqrt{r}\text{tr}\left(O_T^{1/2}\right) && \text{(Fact 2.1)} \\
&\leq \frac{D^2 \text{tr}\,(A_T)}{2\eta} + \eta r \text{tr}\,(A_T) && \text{(using (42) again)} \\
&= D\sqrt{2r}\ \text{tr}\left(L_T^{1/4}\right) \text{tr}\left(R_T^{1/4}\right) && \left(\text{using } \eta = \frac{D}{\sqrt{2r}}\right)
\end{aligned}$$

We can show that $\text{tr}(L_T^{1/4}) = O(T^{1/4})$ and $\text{tr}(R_T^{1/4}) = O(T^{1/4})$, thus the bound is $O(\sqrt{T})$.

6.1 Shampoo with EMA

Shampoo is derived as an approximation to the AdaGrad preconditioner (42) and therefore uses the sum of the gradient outer products (i.e., $L_t = \sum_{l \leq t} G_l G_l^\top$ and $R_t = R_t = \sum_{l \leq t} G_l^\top G_l$). As with RMSProp/Adam, in practice we benefit from replacing it with a running estimate of the expected value $L = \mathbf{E}[G_t G_t^\top]$ and $R = \mathbf{E}[G_t^\top G_t]$, e.g., bias-corrected EMA, so that the update is not made monotonically smaller (Shi *et al.*, 2023). We can then view Shampoo as

$$W_{t+1} = W_t - \eta L^{-1/4} G_t R^{-1/4} \qquad \Leftrightarrow \qquad w_{t+1} = w_t - \eta \big(\underbrace{L^{1/4} \otimes R^{1/4}}_{A_{\text{shampoo}}}\big)^{-1} g_t$$

By adapting (42), we can easily show

$$I_{\text{emp}}^{1/2} \preceq \sqrt{r} L^{1/4} \otimes R^{1/4} = \sqrt{r} A_{\text{shampoo}} \tag{43}$$

So as in RMSProp/Adam, Shampoo with EMA can be motivated as approximating $A_{\text{shampoo}} \approx I_{\text{emp}}^{1/2} \approx I_{\text{fisher}} \approx H$. Instead of using the bound (43) for approximation, Morwani *et al.* (2024) directly approximate $I_{\text{emp}}^{1/2}$ with one round of power iteration and derive the *squared* preconditioner $A_{\text{shampoo}}^2 = L^{1/2} \otimes R^{1/2}$ (Appendix I.1).

Pointers

- Introduction to Online Learning by Francesco Orabona, in particular online gradient descent and adaptive algorithms
- Lecture slides by Sham Kakade
- Lecture slides by Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky
- Blog by Sebastian Ruder
- Notes on mirror descent by Xinhua Zhang
- Course notes by Roger Grosse

[4] From Lemma G.3, we have $O_t = \sum_{l=1}^{t} g_l g_l^\top \preceq r \sum_{l=1}^{t} (G_l G_l^\top) \otimes I_n = r L_t \otimes I_n$ and similarly $O_t \preceq r I_m \otimes R_t$. Using Fact G.4, we get $O_t \preceq r(L_t \otimes I_n)^{1/2}(I_m \otimes R_t)^{1/2} = r L_t^{1/2} \otimes R_t^{1/2}$ and also $O_t^{1/2} \preceq \sqrt{r}(L_t^{1/4} \otimes R_t^{1/4})$.

References

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., *et al.* (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, **12**(7).

Finesso, L. and Spreij, P. (2006). Nonnegative matrix factorization and i-divergence alternating minimization. *Linear Algebra and its Applications*, **416**(2-3), 270–287.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Grosse, R. (2021). Adaptive gradient methods, normalization, and weight decay. https://www.cs.toronto.edu/~rgrosse/courses/csc2541_2021/readings/L05_normalization.pdf.

Gupta, V., Koren, T., and Singer, Y. (2018). Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR.

Hanson, S. and Pratt, L. (1988). Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems*, **1**.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kunstner, F., Hennig, P., and Balles, L. (2019). Limitations of the empirical fisher approximation for natural gradient descent. *Advances in neural information processing systems*, **32**.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *nature*, **401**(6755), 788–791.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Morwani, D., Shapira, I., Vyas, N., Malach, E., Kakade, S., and Janson, L. (2024). A new perspective on shampoo's preconditioner. *arXiv preprint arXiv:2406.17748*.

Reddi, S. J., Kale, S., and Kumar, S. (2019). On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*.

Sankar, A. R., Khasbage, Y., Vigneswaran, R., and Balasubramanian, V. N. (2021). A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9481–9488.

Shazeer, N. and Stern, M. (2018). Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Shi, H.-J. M., Lee, T.-H., Iwasaki, S., Gallego-Posada, J., Li, Z., Rangadurai, K., Mudigere, D., and Rabbat, M. (2023). A distributed data-parallel pytorch implementation of the distributed shampoo optimizer for training neural networks at-scale. *arXiv preprint arXiv:2309.06497*.

Tieleman, T., Hinton, G., *et al.* (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, **4**(2), 26–31.

Van Loan, C. F. and Pitsianis, N. (1993). *Approximation with Kronecker products*. Springer.

Wortsman, M., Liu, P. J., Xiao, L., Everett, K. E., Alemi, A. A., Adlam, B., Co-Reyes, J. D., Gur, I., Kumar, A., Novak, R., Pennington, J., Sohl-Dickstein, J., Xu, K., Lee, J., Gilmer, J., and Kornblith, S. (2024). Small-scale proxies for large-scale transformer training instabilities. In *The Twelfth International Conference on Learning Representations*.

Zhuang, Z., Liu, M., Cutkosky, A., and Orabona, F. (2022). Understanding adamw through proximal methods and scale-freeness. *arXiv preprint arXiv:2202.00089*.

A Stochastic Optimization

Let $x \sim \mathbf{pop}$ define a convex per-example loss $J_x(w) \in \mathbb{R}$ over $w \in V$. Let

$$w^\star = \arg\min_{w \in V} \mathop{\mathbf{E}}_{x \sim \mathbf{pop}} [J_x(w)] \tag{44}$$

Denote the expected loss $J(w) = \mathbf{E}_{x \sim \mathbf{pop}}[J_x(w)]$ (which remains convex) and let $J^\star = J(w^\star)$. Suppose we use an online algorithm for this loss. Starting from some initial $w_1 \in V$, for $t = 1 \ldots T$, we sample an iid $x_t \sim \mathbf{pop}$, get punished by the convex loss $J_{x_t}(w_t) \in \mathbb{R}$, and obtain w_{t+1} from the algorithm. Let Regret(T) be a regret bound of the algorithm so that

$$\sum_{t=1}^T J_{x_t}(w_t) - \min_{w \in V} \sum_{t=1}^T J_{x_t}(w) \leq \text{Regret}(T) \tag{45}$$

In particular,

$$\sum_{t=1}^T J_{x_t}(w_t) - \sum_{t=1}^T J_{x_t}(w^\star) \leq \text{Regret}(T)$$

Dividing both sides by T yields

$$\frac{1}{T}\sum_{t=1}^T J_{x_t}(w_t) - \frac{1}{T}\sum_{t=1}^T J_{x_t}(w^\star) \leq \frac{\text{Regret}(T)}{T}$$

Taking the expectation wrt. the iid samples $x_1 \ldots x_T \sim \mathbf{pop}$ (on both sides), we have

$$\frac{1}{T}\sum_{t=1}^T \mathop{\mathbf{E}}_{x_1 \ldots x_T \sim \mathbf{pop}} [J_{x_t}(w_t)] - J^\star = \frac{1}{T}\sum_{t=1}^T \mathop{\mathbf{E}}_{x \sim \mathbf{pop}, w_t} [J_x(w_t)] - J^\star = \frac{1}{T}\sum_{t=1}^T \mathop{\mathbf{E}}_{w_t} [J(w_t)] - J^\star \leq \frac{\text{Regret}(T)}{T}$$

where w_t is now also a random variable. It is independent of x_t since it is determined by $x_1 \ldots x_{t-1}$ (hence the equalities). By the convexity of J,

$$\frac{1}{T}\sum_{t=1}^T \mathop{\mathbf{E}}_{w_t} [J(w_t)] = \mathop{\mathbf{E}}_{w_1 \ldots w_T} \left[\frac{1}{T}\sum_{t=1}^T J(w_t)\right] \geq \mathop{\mathbf{E}}_{\bar{w}_T} [J_x(\bar{w}_T)]$$

where $\bar{w}_T = \frac{1}{T}\sum_{t=1}^T w_t$. Putting together, we have

$$\mathop{\mathbf{E}}_{\bar{w}_T} [J_x(\bar{w}_T)] - J^\star \leq \frac{\text{Regret}(T)}{T}$$

I.e., taking the average of $w_1 \ldots w_T$ from the online algorithm on (any sequence of) iid samples $x_1 \ldots x_T \sim \mathbf{pop}$ yields a solution $\bar{w}_T$ that on average (wrt. sampling randomness) falls behind $J^\star$ at the rate of $O(\frac{\text{Regret}(T)}{T})$. In particular, if Regret$(T) = o(T)$ (i.e., sublinear regret), the solution is guaranteed to converge to $J^\star$ asymptotically. For instance, if Regret$(T) = O(\sqrt{T})$, the solution has the convergence rate of $O(\frac{1}{\sqrt{T}})$.

B Lower Bound on Regret

Let $V = \{\pm 1\}$ denote the hypothesis space. At each step t, the enemy *randomly* picks $x_t = \pm 1$ and defines the (linear) loss $l_t(w_t) = -x_t w_t$. Then no matter what $w_1 \ldots w_T$ we choose, our expected cumulative loss is always zero by the linearity of expectation and the independence of w_t and x_t

$$\mathop{\mathbf{E}}_{x_1 \ldots x_T} \left[-\sum_{t=1}^T x_t w_t\right] = -\sum_{t=1}^T \underbrace{\mathop{\mathbf{E}}_{x_t} [x_t]}_{0} w_t = 0$$

For any choices of $x_1 \ldots x_T \in \{\pm 1\}$, the hypothesis $u \in \{\pm 1\}$ that achieves the smallest cumulative loss must minimize $-\sum_{t=1}^T x_t u$, which is either $-\sum_{t=1}^T x_t$ or $\sum_{t=1}^T x_t$. This implies that $u = \mathbf{sign}\left(\sum_{t=1}^T x_t\right)$. Thus for any $w_1 \ldots w_T$, the expected regret is

$$\mathop{\mathbf{E}}_{x_1 \ldots x_T} \left[-\sum_{t=1}^T x_t w_t + \sum_{t=1}^T x_t \mathbf{sign}\left(\sum_{t'=1}^T x_{t'}\right)\right] = \mathop{\mathbf{E}}_{x_1 \ldots x_T} \left[\left|\sum_{t=1}^T x_t\right|\right] = \Theta(\sqrt{T})$$

The last term is $\Theta(\sqrt{T})$ just by the central limit theorem.[5] Thus we have constructed a randomized enemy that achieves an $\Omega(\sqrt{T})$ expected regret for any $w_1 \ldots w_T$ asymptotically as $T \to \infty$. This implies the existence of *some* deterministic enemy that achieves an $\Omega(\sqrt{T})$ regret for any $w_1 \ldots w_T$ asymptotically as $T \to \infty$ (aka. Yao's minimax principle). The intuition is that randomization is only a handicap for the enemy, not a feature.

C Lemmas

Lemma C.1 (**First-order necessary condition in constrained optimization).** Let $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^d$. Let $V \subseteq \mathcal{X}$ be a closed convex set. Let $x^\star \in V$ be a local minimizer of f over V (i.e., $-\infty < f(x^\star) \leq f(x)$ for all $x \in V$ within an ϵ-ball around $x^\star$ for some small $\epsilon > 0$) and f is differentiable at $x^\star$. Then

$$\langle \nabla f(x^\star), x - x^\star \rangle \geq 0 \qquad \forall x \in V \tag{46}$$

The converse is not true; there may be cases where $x^\star \in V$ satisfies $\langle \nabla f(x^\star), x - x^\star \rangle \geq 0$ for all $x \in V$ but $x^\star$ is not a local minimizer of f over V.

Proof (lecture style). First, since we assume that f is differentiable at $x^\star$, we can be assured that f is continuous and nicely behaving at least at that point (but $f(x)$ could be still nondifferentiable, discontinuous, or undefined at any $x \neq x^\star$). Second, (46) is useful only because $x^\star$ is a local minimizer *constrained* to be in V; otherwise $\nabla f(x^\star) = 0_d$ and (46) is trivially satisfied. Thus (46) can be seen as simply stating that **for a minimizer at the boundary $x^\star$, any "feasible direction" $x - x^\star$ from $x^\star$ to $x \in V$ must agree with the gradient $\nabla f(x^\star)$** (i.e., both positive, or both negative).

It is best to understand this behavior by visualizing a few choices of f and $x^\star$ on a 1D closed interval $V = [a, b] \subset \mathbb{R}$ (or a 2D closed convex set $V \subset \mathbb{R}^2$ for more intuition). If $x^\star$ is a local minimizer strictly inside V, the gradient is zero and we are done. So we may only concern ourselves with the case where $x^\star \in V$ is on the *boundary* (using the closedness of V) with a *nonzero slope*. In the 1D example, either $x^\star = a$ and $f'(a) > 0$, or $x^\star = b$ and $f'(b) < 0$. In each case, the feasible direction $x - x^\star$ becomes positive or negative to make (46) true.

More formally, pick any $x \in V$. Since V is a convex set, we must have $(1-t)x^\star + tx \in V$ for all $t \in [0, 1]$. Let $\phi : [0, 1] \to \mathbb{R}$ measure the value of f as we walk from $x^\star$ to x, i.e., $\phi(t) = f(x^\star + t(x - x^\star))$. Note that $\phi(0) = f(x^\star)$. Since f is differentiable at $x^\star$, $\phi(t)$ must be differentiable at $t = 0$ from the right, i.e., the following limit exists:

$$\phi'_+(0) = \lim_{h \to 0^+} \frac{\phi(h) - \phi(0)}{h} \tag{47}$$

Since $x^\star \in V$ is a local minimizer, no matter where $x \in V$ is, we must have $\phi(0) = f(x^\star) \leq f(x^\star + t(x - x^\star)) = \phi(t)$ for a small enough $t \geq 0$. But this implies that (47) is nonnegative.

It remains to calculate (47). This is easy if f is differentiable since then $\phi'(t) = \langle \nabla f(x^\star + t(x - x^\star)), x - x^\star \rangle$ and thus $\phi'(0) = \phi'_+(0) = \langle \nabla f(x^\star), x - x^\star \rangle$. In the general case, again using the differentiability of f at $x^\star$ we have $\phi(h) = f(x^\star + h(x - x^\star)) = f(x^\star) + h \langle \nabla f(x^\star), x - x^\star \rangle + o(h)$ for a small enough $h > 0$, so

$$\phi'_+(0) = \lim_{h \to 0^+} \frac{\phi(h) - \phi(0)}{h} = \lim_{h \to 0^+} \frac{h \langle \nabla f(x^\star), x - x^\star \rangle + o(h)}{h} = \langle \nabla f(x^\star), x - x^\star \rangle$$

Finally, it is clear that (46) is not a sufficient condition for $x^\star \in V$ to be a local minimizer since a local maximizer or a saddle point satisfies $\nabla f(x^\star) = 0_d$. It is not clear if there are examples with nonzero gradients (i.e., $\langle \nabla f(x^\star), x - x^\star \rangle > 0$ for all $x \in V$ but $x^\star$ is not a local minimizer). □

Lemma C.2 (**First-order characterization of constrained convex optimization).** Let $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^d$. Let $V \subseteq \mathcal{X}$ be a closed convex set. If f is convex on V,

$$x^\star = \arg\min_{x \in V} f(x) \bigwedge f \text{ is differentiable at } x^\star \qquad \Leftrightarrow \qquad \langle \nabla f(x^\star), x - x^\star \rangle \geq 0 \qquad \forall x \in V \tag{48}$$

[5] Each x_t is an independent Rademacher variable with mean 0 and variance 1, so we have $|\sum_{t=1}^T x_t| \to \sqrt{T}|Z|$ where $Z \sim \mathcal{N}(0, 1)$. The boundedness of $|\cdot|$ implies $\mathbf{E}[|\sum_{t=1}^T x_t|] \to \sqrt{T}\mathbf{E}[|Z|]$ where $\mathbf{E}[|Z|] = \sqrt{2/\pi}$ is some constant.

Proof. The statement is true simply because the convexity of f on V eliminates the local maximizer and saddle point issues, making (46) a sufficient condition for $x^\star \in V$ to be a local (hence global) minimizer. More formally, the direction $\Rightarrow$ is given by Lemma C.1. For the direction $\Leftarrow$, since f is convex on V, for any $x \in V$ we must have

$$f(x) \geq f(x^\star) + \langle \nabla f(x^\star), x - x^\star \rangle \geq f(x^\star)$$

but this means $x^\star$ is a global minimum. □

Lemma C.3 (**First-order characterization of constrained convex optimization (matrix)**). Let $J : \mathbb{R}^{d\times d} \to \mathbb{R}$ be a convex and differentiable function. Let $V \subseteq \mathbb{R}^{d\times d}$ be a closed convex set. Then

$$A^\star = \operatorname*{arg\,min}_{A \in V} J(A) \bigwedge J \text{ is differentiable at } A^\star \qquad \Leftrightarrow \qquad \langle \nabla J(A^\star), A - A^\star \rangle_F \geq 0 \qquad \forall A \in V \tag{49}$$

where $\langle A, B \rangle_F = \operatorname{tr}\left(A^\top B\right) = \sum_{i,j} A_{i,j} B_{i,j}$ is the Frobenius inner product.

Lemma C.4. Let $O_T = \boldsymbol{G}\boldsymbol{G}^\top \in \mathbb{R}^{d\times d}$ where $\boldsymbol{G} = (g_1 \dots g_T) \in \mathbb{R}^{T\times d}$ is the matrix of gradients with rank d. Then

$$\operatorname{tr}\left(O_T^{1/2}\right) \leq \sum_{i=1}^{d} \sqrt{\sum_{t=1}^{T} g_{t,i}^2} \tag{50}$$

with equality iff O_T is diagonal; more specifically, $O_T = \operatorname{diag}\left(\sigma_1^2 \dots \sigma_d^2\right)$ where $\sigma_1 > \dots > \sigma_d > 0$ are the (distinct, for convenience) singular values of $\boldsymbol{G}$.

Proof. Let $\boldsymbol{G} = U\Sigma V^\top$ denote an SVD of $\boldsymbol{G}$. Then $O_T = V\Sigma^2 V^\top$ and thus $O_T^{1/2} = V\Sigma V^\top$, so the LHS can be expressed as $\operatorname{tr}\left(O_T^{1/2}\right) = \sum_{i=1}^d \sigma_i$ (i.e., the nuclear norm $||\boldsymbol{G}||_*$). Let $\gamma_1 \dots \gamma_d \in \mathbb{R}^T$ denote the columns of $\boldsymbol{G}$. Note that $\gamma_i = \boldsymbol{G}e_i = U\Sigma\tilde{v}_i$ where $\tilde{v}_i \in \mathbb{R}^d$ is the i-th row of $V \in \mathbb{R}^{d\times d}$ (thus $\tilde{v}_1 \dots \tilde{v}_d$ are orthonormal). Then the RHS can be expressed as $\sum_{i=1}^d ||\gamma_i||_2 = \sum_{i=1}^d ||U\Sigma\tilde{v}_i||_2 = \sum_{i=1}^d ||\Sigma\tilde{v}_i||_2$. Thus the claim (50) can be rephrased as: given any matrix with singular values Σ and right singular vectors V containing rows $\tilde{v}_1 \dots \tilde{v}_d \in \mathbb{R}^d$, we must always ahve

$$\sum_{i=1}^{d} ||\Sigma e_i||_2 \leq \sum_{i=1}^{d} ||\Sigma\tilde{v}_i||_2 \tag{51}$$

Since Σ is on both sides, we vary the choice of V. WLOG we can assume that V is a 2×2 rotation matrix for the following reasons:

- V is orthonormal. So it can be expressed as a product of Givens rotations and at most one reflection (i.e., $\operatorname{diag}(-e_i) I_d$).
- Reflection does not affect the RHS of (51).
- Thus if no rotation in a 2D subspace reduces the RHS of (51), neither does any V.

Hence assuming $\tilde{v}_1 = (\cos\theta, -\sin\theta)$ and $\tilde{v}_2 = (\sin\theta, \cos\theta)$ for some radian θ, we can write down the objective:

$$\min_{0 \leq \theta < 2\pi} \sqrt{\sigma_1^2 \cos^2\theta + \sigma_2^2 \sin^2\theta} + \sqrt{\sigma_1^2 \sin^2\theta + \sigma_2^2 \cos^2\theta}$$

We can easily check that the minimum is $\sigma_1 + \sigma_2$ and the minimizers are $\theta^\star \in \left\{0, \frac{\pi}{2}\right\}$ corresponding to $V = I_2$ and $V = [[0, 1], [1, 0]]$. The latter violates the structure of V imposed by the SVD (i.e., the ordering $\sigma_1 > \sigma_2$), thus we conclude $V = I_2$. We have established that (50) holds with equality iff $\boldsymbol{G} = U\Sigma$. This condition is equivalent to the condition in the statement. Specifically, the forward direction is $O_T = \boldsymbol{G}^\top\boldsymbol{G} = \Sigma^2$. The backward direction is: if $O_T = D$ for some diagonal $D \succ 0$, then $\boldsymbol{G}^\top\boldsymbol{G} = D$ which implies there exists some orthonormal $U \in \mathbb{R}^{T\times d}$ such that $\boldsymbol{G} = UD^{1/2}$, so D is a diagonal matrix of the squared singular values of $\boldsymbol{G}$. □

Lemma C.5. Let $z \in \mathbb{N}_0^n$ where $z_i \sim \operatorname{Poi}(\lambda_i)$ is an independent count for some rate $\lambda_i > 0$. Let $x = \sum_{i=1}^n z_i \in \mathbb{N}_0$. Then $p(z|x) = \operatorname{Mult}(x, \bar{\lambda})(z)$ where $\bar{\lambda}_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}$.

Proof. This is a consequence of the fact that $x \sim \text{Poi}(\Lambda)$ where $\Lambda = \sum_{i=1}^n \lambda_i$. Then for any $z \in \mathbb{N}_0^n$ and $x \in \mathbb{N}_0$ such that $x = \sum_{i=1}^n z$,

$$p(z, x) = p(z) = \prod_{i=1}^n \frac{\lambda_i^{z_i} e^{-\lambda_i}}{z_i!} \qquad\qquad p(x) = \frac{\Lambda^x e^{-\Lambda}}{x!}$$

so that

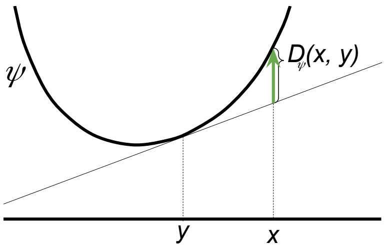
$$p(z|x) = \frac{p(z,x)}{p(x)} = \frac{\left(\prod_i \lambda_i^{z_i}\right)\left(e^{-\Lambda}\right)}{\prod_i z_i!} \frac{x!}{\Lambda^x e^{-\Lambda}} = \frac{x!}{\prod_i z_i!} \frac{\prod_i \lambda_i^{z_i}}{\Lambda^x} = \frac{x!}{\prod_i z_i!} \frac{\left(\prod_i \bar{\lambda}_i^{z_i}\right)\left(\Lambda^x\right)}{\Lambda^x} = \frac{x!}{\prod_i z_i!} \prod_i \bar{\lambda}_i^{z_i} = \text{Mult}(x, \bar{\lambda})(z)$$

□

D Bregman Divergence

Let $\psi : \Omega \to \mathbb{R}$ be a strictly convex and differentiable function over a convex set $\Omega \subseteq \mathbb{R}^d$. The associated **Bregman divergence** $D_\psi(x, y)$ (from y to x) measures the error of the first-order approximation of ψ around $y \in \Omega$ at $x \in \Omega$.

$$D_\psi(x, y) := \psi(x) - \psi(y) - \nabla\psi(y)^\top (x - y)$$



Since ψ is strictly convex, $D_\psi(x, y) \geq 0$ and zero iff $x = y$. We say ψ is **σ-strongly convex** with respect to the norm $||\cdot||$ if $D_\psi(x, y) \geq \frac{\sigma}{2} ||x - y||^2$. $D_\psi(x, y)$ is clearly assymetric. It is trivially convex and differentiable in x with the gradient $\nabla_x D_\psi(x, y) = \nabla\psi(x) - \nabla\psi(y)$. It is not necessarily convex in y. The two most important examples of Bregman divergence are as follows:

1. For any $A \succ 0$, the A-weighted Euclidean norm $\psi : \mathbb{R}^d \to \mathbb{R}$ induces the **A-weighted Euclidean distance** (Appendix L.3).

$$\psi(x) = \frac{1}{2} ||x||_A^2 \qquad \Rightarrow \qquad D_\psi(x, y) = \frac{1}{2} ||x - y||_A^2 \tag{52}$$

   Clearly, ψ is 1-strongly convex wrt. $||\cdot||_A$. In particular, $\psi(x) = \frac{1}{2} ||x||_2^2$ is 1-strongly convex wrt. the l_2 norm.

2. The negative entropy $\psi : \Delta^{d-1} \to \mathbb{R}$ induces the **KL divergence**.

$$\psi(x) = \sum_{i=1}^d x_i \log x_i \qquad \Rightarrow \qquad D_\psi(x, y) = \text{KL}(x, y) \tag{53}$$

   Pinsker's inequality gives us $\text{KL}(x, y) \geq \frac{1}{2} ||x - y||_1^2$, thus ψ is 1-strongly convex wrt. the l_1 norm.

D.1 Generalized Pythagorean Theorem

Lemma D.1. For all $x, y, z \in \Omega$,

$$D_\psi(y, x) = D_\psi(z, x) + D_\psi(y, z) + (\nabla\psi(z) - \nabla\psi(x))^\top (y - z) \tag{54}$$

Proof. Let $D_\psi(y, x) = D_\psi(z, x) + D_\psi(y, z) + C$ for some term C. Expanding by definition,

$$\begin{aligned} C &= D_\psi(y, x) - D_\psi(z, x) - D_\psi(y, z) \\ &= \{\cancel{\psi(y)} - \cancel{\psi(x)} - \nabla\psi(x)^\top (y - \cancel{x})\} - \{\cancel{\psi(z)} - \cancel{\psi(x)} - \nabla\psi(x)^\top (z - \cancel{x})\} - \{\cancel{\psi(y)} - \cancel{\psi(z)} - \nabla\psi(z)^\top (y - z)\} \\ &= (\nabla\psi(z) - \nabla\psi(x))^\top (y - z) \end{aligned}$$

□

Lemma D.2. Let $\mathcal{C} \subseteq \Omega$ be a convex and closed set. Pick any $x \in \Omega$. Let

$$p_x = \arg\min_{z \in \mathcal{C}} \; D_\psi(z, x) \tag{55}$$

denote the **Bregman projection** of $x \in \Omega$ onto $\mathcal{C}$. Then for all $y \in \mathcal{C}$,

$$D_\psi(y, x) \geq D_\psi(p_x, x) + D_\psi(y, p_x) \tag{56}$$

where the inequality is tight iff $\nabla\psi(p_x) - \nabla\psi(x)$ is orthogonal to $y - p_x$.

Proof. By Lemma D.1, we only need to show $(\nabla\psi(p_x) - \nabla\psi(x))^\top (y - p_x) \geq 0$ for all $y \in \mathcal{C}$. Since $p_x = \arg\min_{z\in\mathcal{C}} f(z)$ is the minimizer of the convex function $f(z) = D_\psi(z, x)$ over $\mathcal{C}$, it follows that $\nabla f(p_x)^\top (y - p_x) \geq 0$ (48). But $\nabla f(p_x) = \nabla\psi(p_x) - \nabla\psi(x)$. □

Example D.1 (Pythagorean theorem)**.** Let $\Omega = \mathbb{R}^d$ and $\mathcal{C} \subseteq \mathbb{R}^d$ be a subspace. Let $\psi(x) = ||x||_2^2$ which induces the squared Euclidean distance $D_\psi(x, z) = ||x - z||_2^2$ in $\mathbb{R}^d$. Then $p_x \in \mathcal{C}$ is the orthogonal projection of x onto the subspace $\mathcal{C}$ where $x - p_x$ is orthogonal to $\mathcal{C}$. In particular, $\nabla\psi(p_x) - \nabla\psi(x) = p_x - x$ is orthogonal to $y - p_x$ for any $y \in \mathcal{C}$, hence (56) holds with equality, i.e., the usual Pythagorean theorem: $||x - y||_2^2 = ||x - p_x||_2^2 + ||p_x - y||_2^2$.

D.1.1 Regularized Bregman projection

We can further extend Lemma D.2 to regularize the Bregman projection with a convex function.

Lemma D.3. Let $\mathcal{C} \subseteq \Omega$ be a convex and closed set. Let $l : \mathcal{C} \to \mathbb{R}$ be convex and differentiable. Pick any $x \in \Omega$. Let

$$p_x = \arg\min_{z \in \mathcal{C}} \; D_\psi(z, x) + l(z) \tag{57}$$

denote the Bregman projection of $x \in \Omega$ onto $\mathcal{C}$ regularized by l. Then for all $y \in \mathcal{C}$,

$$D_\psi(y, x) + l(y) \geq D_\psi(p_x, x) + D_\psi(y, p_x) + l(p_x) \tag{58}$$

where the inequality is tight iff l is affine, i.e., $g = \nabla l(z)$ for any $z \in \mathcal{C}$, and $\nabla\psi(p_x) - \nabla\psi(x) + g$ is orthogonal to $y - p_x$.

Proof. Define $f(z) = D_\psi(z, x) + l(z)$ which is convex and differentiable. Since $p_x = \arg\min_{z\in\mathcal{C}} f(z)$, it follows from (48) that

$$\begin{aligned} 0 \leq \nabla f(p_x)^\top (y - p_x) &= (\nabla\psi(p_x) - \nabla\psi(x) + \nabla l(p_x))^\top (y - p_x) \\ &= D_\psi(y, x) - D_\psi(p_x, x) - D_\psi(y, p_x) + \nabla l(p_x)^\top (y - p_x) && \text{(Lemma D.1)} \\ &\leq D_\psi(y, x) - D_\psi(p_x, x) - D_\psi(y, p_x) + l(y) - l(p_x) \end{aligned}$$

where the inequality uses the convexity of l. Rearranging the terms gives (58). The second inequality is tight iff l is affine and the first inequality is tight iff $\nabla f(p_x)^\top (y - p_x) = 0$, thus (58) is tight iff both conditions hold. □

D.2 Other Properties

Lemma D.4 (Duality)**.** Assume Ω is closed. Then $D_\psi(y, x) = D_{\psi^*}(\nabla\psi(x), \nabla\psi(y))$

Proof. Since Ω is closed, $\psi^*(p) = \sup_{x\in\Omega} p^\top x - \psi(x)$ is well defined. Strict convexity implies $\nabla\psi : \Omega \to \mathbb{R}^d$ is invertible, so $\nabla(\psi^*) = (\nabla\psi)^{-1}$ and $\psi^*(p) = p^\top (\nabla\psi)^{-1}(p) - \psi((\nabla\psi)^{-1}(p))$ by Fact F.4. The latter implies that $\psi^*(\nabla\psi(z)) = \nabla\psi(z)^\top z - \psi(z)$. We can directly verify the equality:

$$\begin{aligned} D_{\psi^*}(\nabla\psi(x), \nabla\psi(y)) &= \psi^*(\nabla\psi(x)) - \psi^*(\nabla\psi(y)) - \nabla\psi^*(\nabla\psi(y))(\nabla\psi(x) - \nabla\psi(y)) \\ &= \left\{\nabla\psi(x)^\top x - \psi(x)\right\} - \left\{\nabla\psi(y)^\top y - \psi(y)\right\} - y^\top (\nabla\psi(x) - \nabla\psi(y)) \\ &= \psi(y) - \psi(x) - \psi(y)^\top (y - x) \\ &= D_\psi(y, x) \end{aligned}$$

□

Lemma D.5 (Mean as minimizer)**.** Let p be a distribution over a closed set $S \subseteq \Omega$. Define $x^\star = \arg\min_{x \in S} \mathbf{E}_{y \sim p}[D_\psi(y, x)]$. Then $x^\star = \mu_p$ (i.e., the mean of p).

Proof. By the linearity of expectation,

$$\mathop{\mathbf{E}}_{y \sim p}[D_\psi(y, x)] = \mathop{\mathbf{E}}_{y \sim p}[\psi(y)] - \psi(x) - \nabla\psi(x)^\top(\mu_p - x)$$

To see μ_p is optimal, consider any $x \in S$ and note

$$\mathop{\mathbf{E}}_{y \sim p}[D_\psi(y, x)] - \mathop{\mathbf{E}}_{y \sim p}[D_\psi(y, \mu_p)] = \psi(\mu_p) - \psi(x) - \nabla\psi(x)^\top(\mu_p - x) = D_\psi(\mu_p, x) \geq 0$$

which is minimized to zero at $x = \mu_p$. □

E Exponentiated Gradient Descent

Lemma E.1. In mirror descent (1), choose $V = \Delta^{d-1}$ and $\psi_t(w) = -H(w) = \sum_{i=1}^d w_i \log w_i$ so that the objective reduces to (see (53))

$$w_{t+1} = \mathop{\arg\min}_{w \in \Delta^{d-1}} \; g_t^\top w + \frac{1}{\eta_t}\mathrm{KL}(w, w_t) \tag{59}$$

where $w_t \in \Delta^{d-1}$ is assumed full-support. Then w_{t+1} satisfies

$$w_{t+1,i} = \frac{w_{t,i}\exp(-\eta_t g_{t,i})}{\sum_{j=1}^d w_{t,j}\exp(-\eta_t g_{t,j})} \tag{60}$$

Proof I. The objective is convex since KL is strictly convex in the first argument. Slater's condition holds, so we can find a solution of the Lagrangian that satisfies the KKT conditions. The Lagrangian is:

$$L(w, \lambda, \tau) = \eta_t g_t^\top w + \sum_{i=1}^d w_i \log\frac{w_i}{w_{t,i}} - \lambda^\top w + \tau(1_d^\top w - 1)$$

The optimal solution satisfies the stationarity condition

$$\frac{\partial L(w, \lambda, \tau)}{\partial w_i} = \eta_t g_{t,i} + \log\frac{w_i}{w_{t,i}} + 1 - \lambda_i + \tau = 0 \qquad \Leftrightarrow \qquad w_i = w_{t,i}\exp(-\eta_t g_{t,i} - 1 + \lambda_i + \tau)$$

Since $w_i \geq 0$, we may use $\lambda = 0_d$. Enforcing the constraint $\sum_j w_j = 1$ yields $\tau = -\log(\sum_j w_{t,j}\exp(-\eta_t g_{t,j} - 1))$. Plugging it in the expression, we get

$$w_i = \frac{w_{t,i}\exp(-\eta_t g_{t,i} - 1)}{\sum_j w_{t,j}\exp(-\eta_t g_{t,j} - 1)} = \frac{w_{t,i}\exp(-\eta_t g_{t,i})}{\sum_j w_{t,j}\exp(-\eta_t g_{t,j})}$$

□

Proof II. The objective is equivalent to

$$\mathop{\mathbf{E}}_{i \sim w}\left[\eta_t g_{t,i} + \log\frac{w_i}{w_{t,i}}\right] = \mathop{\mathbf{E}}_{i \sim w}\left[\log\frac{w_i}{w_{t,i}\exp(-\eta_t g_{t,i})}\right] = \mathop{\mathbf{E}}_{i \sim w}\left[\log\frac{w_i}{u_{t,i}}\right] - \log z_t$$

where $z_t = \sum_j w_{t,j}\exp(-\eta_t g_{t,j})$ and $u_t = w_t / z_t \in \Delta^{d-1}$. Thus $w_{t+1} = \arg\min_{w \in \Delta^{d-1}} \mathrm{KL}(w, u) = u$. □

The argument in Proof II applies equally to the "KL-constrained RL problem" where the goal is to find the next policy by maximizing the expected reward $r(y) \in \mathbb{R}$ for action $y \sim \pi$ subject to the contraint $\mathrm{KL}(\pi, \pi_t) \leq C$.

$$\pi_{t+1} = \mathop{\arg\max}_{\pi \in \Delta^{d-1}} \mathop{\mathbf{E}}_{y \sim \pi}[r(y)] - \frac{1}{\eta_t}\mathrm{KL}(\pi, \pi_t) \qquad \Rightarrow \qquad \pi_{t+1}(y) \propto \pi_t(y)e^{r(y)}$$

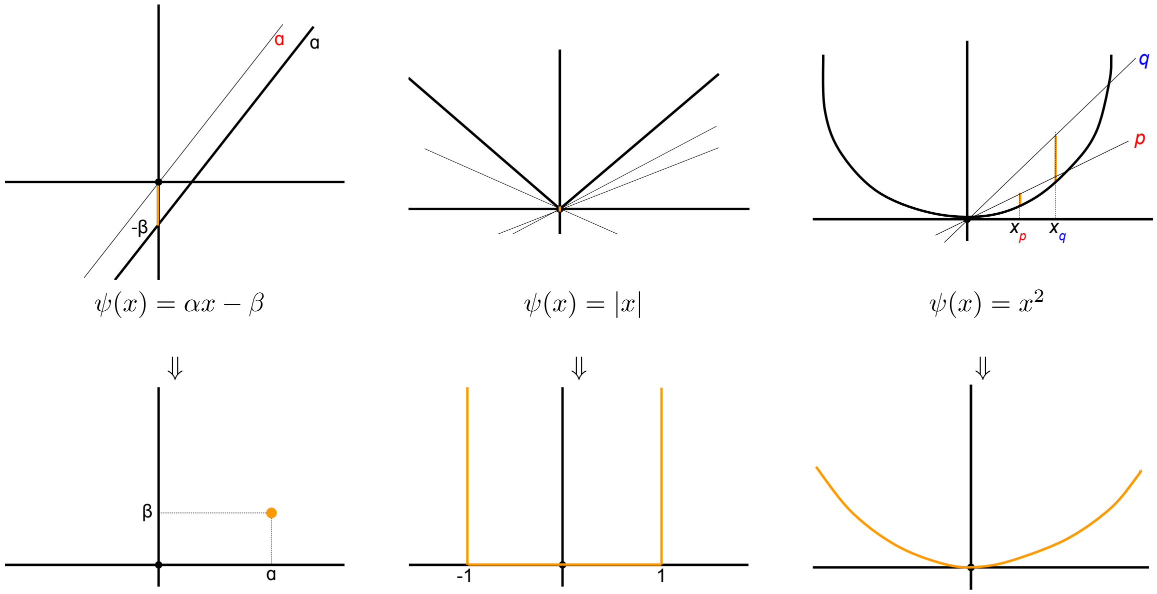
We can see that (59) is a special case where the action space is $i \in \{1 \ldots d\}$ and the reward is the gradient $g_{t,i} \in \mathbb{R}$.

F Convex Conjugate

Let $\psi : \mathbb{R} \to \mathbb{R}$. Its **convex conjugate** $\psi^* : \mathbb{R} \to \mathbb{R}$ maps a slope p to how much px can overestimate $\psi(x)$:

$$\psi^*(p) = \max_{x \in \mathbb{R}} \left\{ px - \psi(x) \right\} \tag{61}$$

ψ^* is always convex no matter what ψ is (because $\psi^*(p)$ is the pointwise maximum of affine functions of p).



$\psi(x) = \alpha x - \beta$ $\qquad$ $\psi(x) = |x|$ $\qquad$ $\psi(x) = x^2$

$\psi^*(p) = \beta$ if $p = \alpha$, else ∞ $\qquad$ $\psi^*(p) = 0$ if $|p| \leq 1$, else ∞ $\qquad$ $\psi^*(p) = \frac{p^2}{4}$

Lemma F.1. If ψ is convex and differentiable with an invertible ψ', then $\psi^*(p) = p \times (\psi')^{-1}(p) - \psi((\psi')^{-1}(p))$.

Proof. Since ψ is convex, any $x_p \in \mathbb{R}$ satisfying $\psi'(x_p) = p$ is an optimal solution in (61). (This is visually clear in the rightmost example.) Since ψ' is invertible, $x_p = (\psi')^{-1}(p)$ is unique. □

Lemma F.2. If ψ is convex and differentiable with an invertible ψ', then ψ^* is differentiable with $(\psi^*)' = (\psi')^{-1}$.

Proof. By Lemma F.1, we have $\psi^*(p) = p \times (\psi')^{-1}(p) - \psi((\psi')^{-1}(p))$. An inverse function is differentiable, so we can use the product rule and the chain rule to obtain

$$(\psi^*)'(p) = (\psi')^{-1}(p) + p \times ((\psi')^{-1})'(p) - \underbrace{\psi'((\psi')^{-1}(p))}_{p} \times ((\psi')^{-1})'(p) = (\psi')^{-1}(p)$$

□

Lemma F.3. $\psi(x) \geq \psi^{**}(x)$ for all $x \in \mathbb{R}$. If ψ is convex and differentiable with an invertible ψ', then $\psi = \psi^{**}$.

Proof. For the first claim,

$$\begin{aligned} \psi^*(p) \geq px - \psi(x) \quad \forall x, p \in \mathbb{R} \qquad &\Leftrightarrow \qquad \psi(x) \geq px - \psi^*(p) \quad \forall x, p \in \mathbb{R} \\ &\Leftrightarrow \qquad \psi(x) \geq \max_{p \in \mathbb{R}} \left\{ xp - \psi^*(p) \right\} = \psi^{**}(x) \quad \forall x \in \mathbb{R} \end{aligned}$$

For the second claim, since $\psi' : \mathbb{R} \to \mathbb{R}$ is a bijection,

$$\psi^{**}(x) = \max_{p \in \mathbb{R}} \left\{ xp - \psi^*(p) \right\} = \max_{y \in \mathbb{R}} \left\{ x\psi'(y) - \psi^*(\psi'(y)) \right\}$$

By Lemma F.1, the last term becomes

$$\psi^*(\psi'(y)) = \psi'(y) \times (\psi')^{-1}(\psi'(y)) - \psi((\psi')^{-1}(\psi'(y))) = \psi'(y)y - \psi(y)$$

Plugging this back in, we have

$$\psi^{**}(x) = \max_{y \in \mathbb{R}} \left\{ \psi(y) - \psi'(y)(y - x) \right\}$$

Using the fact that ψ is strongly convex (implied by the premise), we can easily verify that the RHS is maximized at $y = x$, thus $\psi^{**}(x) = \psi(x)$. Intuitively, the expression considers all lines tangent to ψ and picks the one that gives minimum underestimation at x. □

Exercise 1. Verify that Lemma F.1, F.2, and F.3 hold for $\psi(x) = x^2$.

F.1 Vector-Valued Input

The results for scalar-valued input easily generalize to vector-valued input $\psi : \mathbb{R}^d \to \mathbb{R}$. We summarize them below.

Fact F.4. Let $\psi : \mathbb{R}^d \to \mathbb{R}$. Its **convex conjugate** $\psi^* : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\psi^*(p) = \max_{x \in \mathbb{R}} \left\{ p^\top x - \psi(x) \right\} \tag{62}$$

ψ^* is convex and $\psi(x) \geq \psi^{**}(x)$ for all $x \in \mathbb{R}^d$. If ψ is convex and differentiable with an invertible gradient $\nabla\psi : \mathbb{R}^d \to \mathbb{R}^d$,

$$\psi^*(p) = p^\top (\nabla\psi)^{-1}(p) - \psi((\nabla\psi)^{-1}(p)) \tag{63}$$

$$\nabla(\psi^*) = (\nabla\psi)^{-1} \tag{64}$$

$$\psi = \psi^{**} \tag{65}$$

G Kronecker Product

The **Kronecker product** $C = A \otimes B$ of $A \in \mathbb{R}^{m \times d}$ and $B \in \mathbb{R}^{n \times l}$ is defined as

$$C = \begin{bmatrix} A_{1,1}B & \cdots & A_{1,n}B \\ \vdots & \ddots & \vdots \\ A_{m,1}B & \cdots & A_{m,n}B \end{bmatrix} \in \mathbb{R}^{mn \times dl}$$

(i.e., md copies of $B \in \mathbb{R}^{n \times l}$, each scaled by $A_{i,j} \in \mathbb{R}$). Specifically,

$$C_{(i_1-1)n+j_1,(i_2-1)l+j_2} = A_{i_1,i_2} \times B_{j_1,j_2} \tag{66}$$

for $i_1 \in [m]$, $j_1 \in [n]$, $i_2 \in [d]$, and $j_2 \in [l]$ (we shorthand $[N] = \{1 \ldots N\}$). One way to see this is: for each row i_1 of A, we go through all the rows j_1 of B. Let $\overline{\text{vec}} : \mathbb{R}^{m \times n} \to \mathbb{R}^{mn}$ denote row-major vectorization (e.g., $\overline{\text{vec}}([[a, b]; [c, d]]) = (a, b, c, d)$). Then (reference)

$$\overline{\text{vec}}(ABC) = (A \otimes C^\top)\overline{\text{vec}}(B) \tag{67}$$

(e.g., $\overline{\text{vec}}(uv^\top) = u \otimes v$). By the **mixed-product property**,

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{68}$$

Matrix transpose distributes without reordering as: $(A \otimes B)^\top = A^\top \otimes B^\top$.

G.1 Optimal Kronecker Decomposition

For any $A \in \mathbb{R}^{m \times d}$ and $B \in \mathbb{R}^{n \times l}$, we would like to define a permutation of the $mdnl$ values in $A \otimes B \in \mathbb{R}^{mn \times dl}$ into the shape $md \times nl$ such that

$$\textbf{rearrange}(A \otimes B) = \overline{\text{vec}}(A)\overline{\text{vec}}(B)^\top \tag{69}$$

We can reverse-engineer the correspondence. Since

$$(A \otimes B)_{(i_1-1)n+j_1,(i_2-1)l+j_2} = A_{i_1,i_2} \times B_{j_1,j_2} = (\overline{\text{vec}}(A)\overline{\text{vec}}(B)^\top)_{(i_1-1)d+i_2,(j_1-1)l+j_2}$$

for $i_1 \in [m]$, $j_1 \in [n]$, $i_2 \in [d]$, and $j_2 \in [l]$, we can define **rearrange** : $\mathbb{R}^{mn \times dl} \rightarrow \mathbb{R}^{md \times nl}$ by

$$\textbf{rearrange}(H)_{(i_1-1)d+i_2,(j_1-1)l+j_2} := H_{(i_1-1)n+j_1,(i_2-1)l+j_2} \tag{70}$$

We will assume that m, d, n, l are known when we call this function (for the given input of shape $mn \times dl$). By definition, (70) satisfies (69). A useful property of the function is that it is an involution (i.e., its own inverse):

$$\textbf{rearrange}(\textbf{rearrange}(H)) = H \tag{71}$$

One way to see this is to view the function simply as changing the way how we read the $mdnl$ input values by $(i_1, j_1, i_2, j_2) \mapsto (i_1, i_2, j_1, j_2)$ where we *swap* two axes, so applying it again recovers the original way. Van Loan and Pitsianis (1993) proposed the rearrangement for finding an optimal Kronecker decomposition of a matrix due to the following property:

Lemma G.1. Let $C \in \mathbb{R}^{mn \times dl}$. For any $A \in \mathbb{R}^{m \times d}$ and $B \in \mathbb{R}^{n \times l}$,

$$||C - A \otimes B||_F = \left|\left|\textbf{rearrange}(C) - \text{vec}(A)\text{vec}(B)^\top\right|\right|_F$$

Proof. Using (69), the obvious linearity of **rearrange**, and the fact that $||\cdot||_F$ is unaffected by rearranging values,

$$\begin{aligned}\left|\left|\textbf{rearrange}(C) - \text{vec}(A)\text{vec}(B)^\top\right|\right|_F &= ||\textbf{rearrange}(C) - \textbf{rearrange}(A \otimes B)||_F \\ &= ||\textbf{rearrange}(C - A \otimes B)||_F \\ &= ||C - A \otimes B||_F\end{aligned}$$

□

Corollary G.2. Let $C \in \mathbb{R}^{mn \times dl}$ and

$$A^\star, B^\star = \operatorname*{arg\,min}_{A \in \mathbb{R}^{m \times d}, B \in \mathbb{R}^{n \times l}} ||C - A \otimes B||_F$$

A solution is given by

$$\begin{aligned}A^\star &= a \times \text{view}(u_1, m, d) & \qquad \textbf{rearrange}(C) &= \sum_i \sigma_i u_i v_i^\top \\ B^\star &= b \times \text{view}(v_1, n, l) & &\end{aligned}$$

where $a \times b = \sigma_1$ and $M = \text{view}(u, d_1, d_2)$ arranges $u \in \mathbb{R}^{d_1 d_2}$ into a matrix of shape $M \in \mathbb{R}^{d_1 \times d_2}$ in row-major order (i.e., view in PyTorch). In other words, optimal Kronecker decomposition reduces to optimal rank-1 approximation of $\textbf{rearrange}(C) \in \mathbb{R}^{md \times nl}$, which is solvable by SVD.

G.2 Kronecker Product Between Square Matrices

If $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{n \times n}$ have eigenvalues $\lambda_1 \ldots \lambda_m$ and $\mu_1 \ldots \mu_n$, the mn eigenvalues of $A \otimes B \in \mathbb{R}^{mn \times mn}$ are $\lambda_1\mu_1 \ldots \lambda_m\mu_n$. It follows that

$$\text{tr}\,(A \otimes B) = \text{tr}\,(A)\,\text{tr}\,(B) \tag{72}$$

$$A, B \succeq 0 \qquad \Rightarrow \qquad (A \otimes B)^p = A^p \otimes B^p \qquad \forall p \in \mathbb{R} \tag{73}$$

$$A, B \succeq 0 \qquad \Rightarrow \qquad A \otimes B \succeq 0 \tag{74}$$

From (74), we can also infer that[6]

$$A \succeq A' \succeq 0, \quad B \succeq B' \succeq 0 \qquad \Rightarrow \qquad A \otimes B \succeq A' \otimes B' \tag{75}$$

[6] $A \otimes B = (A' + C) \otimes (B' + D) = A' \otimes B' + A' \otimes D + C \otimes B' + C \otimes D \succeq A' \otimes B'$ since $C = A - A' \succeq 0$ and $D = B - B' \succeq 0$.

G.3 Outer Product Bound

Lemma G.3. Let $A \in \mathbb{R}^{m\times n}$ be any matrix where $\operatorname{rank}(A) \leq r$. If $a = \overline{\mathrm{vec}}(A) \in \mathbb{R}^{mn}$,

$$\begin{aligned} aa^\top &\preceq r(AA^\top) \otimes I_n \\ aa^\top &\preceq rI_m \otimes (A^\top A) \end{aligned}$$

Proof. Let $A = \sum_{k=1}^r \sigma_k u_k v_k^\top \in \mathbb{R}^{m\times n}$ be a thin SVD. Since $\overline{\mathrm{vec}}$ is linear, $a = \sum_{k=1}^r \sigma_k \overline{\mathrm{vec}}(u_k v_k^\top) = \sum_{k=1}^r \sigma_k (u_k \otimes v_k)$. Thus

$$\begin{aligned} aa^\top = \left(\sum_{k=1}^r \sigma_k (u_k \otimes v_k)\right)\left(\sum_{k=1}^r \sigma_k (u_k \otimes v_k)\right)^\top &\preceq r\sum_{k=1}^r \sigma_k^2 (u_k \otimes v_k)(u_k \otimes v_k)^\top && (76) \\ &= r\sum_{k=1}^r \sigma_k^2 (u_k u_k^\top) \otimes (v_k v_k^\top) \\ &\preceq r\sum_{k=1}^r \sigma_k^2 (u_k u_k^\top) \otimes I_n && (77) \\ &= r(AA^\top) \otimes I_n \end{aligned}$$

(76) uses the fact that $(\sum_{i=1}^r w_i)(\sum_{i=1}^r w_i)^\top \preceq r\sum_{i=1}^r w_i w_i^\top$ for any $w_1 \ldots w_r \in \mathbb{R}^d$.[7] (77) follows from (75) since $I_n \succeq v_k v_k^\top$. □

We invoke the fact that the geometric mean of PSD matrices respects Loewner order (aka. "operator monotone"):

Fact G.4. Let $Y_1 \succeq X_1 \succeq 0$ and $Y_2 \succeq X_2 \succeq 0$ be PSD (square) matrices. Then $Y_1^\alpha Y_2^{1-\alpha} \succeq X_1^\alpha X_2^{1-\alpha}$ for all $\alpha \in [0,1]$.

Corollary G.5. Let $A \in \mathbb{R}^{m\times n}$ be any matrix where $\operatorname{rank}(A) \leq r$. If $a = \overline{\mathrm{vec}}(A) \in \mathbb{R}^{mn}$,

$$aa^\top \preceq r(AA^\top \otimes A^\top A)^{1/2} \tag{78}$$

Proof. By Lemma G.3, we have $r(AA^\top) \otimes I_n \succeq aa^\top$ and $rI_m \otimes (A^\top A) \succeq aa^\top$. Applying Fact G.4, we have $aa^\top \preceq r((AA^\top) \otimes I_n)(I_m \otimes (A^\top A)) = r(AA^\top \otimes A^\top A)$. □

H Hessian

Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ denote an input-label pair. Let $f_w : \mathcal{X} \to \mathbb{R}^K$ denote a neural network parameterized by $w \in \mathbb{R}^d$. Let $L : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ denote a loss function differentiable in the first argument. The most important loss is the cross-entropy loss with $\mathcal{Y} = \{1 \ldots K\}$ given by $L(z, y) = -\log p_z(y) = \log(\sum_k e^{z_k}) - z_y$, whose gradient is famously $\nabla_z L(z, y) = p_z - e_y \in \mathbb{R}^K$ where $e_y \in \{0,1\}^K$ is the y-th standard basis. We typically rely on the gradient

$$\nabla_w L(f_w(x), y) = \frac{\partial L(f_w(x), y)}{\partial w} \in \mathbb{R}^d \tag{79}$$

for optimizing f_w. In contrast, the Hessian

$$H_{x,y}(w) = \nabla_w^2 L(f_w(x), y) = \frac{\partial^2 L(f_w(x), y)}{\partial w^2} \in \mathbb{R}^{d\times d} \tag{80}$$

is avoided because of the d^2 size, even though it yields a much faster convergence rate (e.g., Newton's method). It is informative to analyze (80) nonetheless. As in backpropagation, we first decompose it by disentangling f_w and L via the chain rule (Appendix J). This yields

$$H_{x,y}(w) = \underbrace{\nabla_w f_w(x)}_{d\times K} \underbrace{\left(\nabla_z^2 L(z, y)\big|_{z=f_w(x)}\right)}_{K\times K} \underbrace{\nabla_w f_w(x)^\top}_{K\times d} + \underbrace{\nabla_w^2 f_w(x)}_{d\times d\times K} \underbrace{\left(\nabla_z L(z, y)\big|_{z=f_w(x)}\right)}_{K\times 1} \tag{81}$$

[7] This follows from Jensen's inequality and the convexity of $f(z) = z^2$ (i.e., $f(\sum_{i=1}^r z_i) \leq (1/r)\sum_{i=1}^r f(z_i)$). Pick any $x \in \mathbb{R}^d$ and denote $z_i = x^\top w_i$. Then $x^\top \left(\sum_{i=1}^r w_i\right)\left(\sum_{i=1}^r w_i\right)^\top x = \left(\sum_{i=1}^r z_i\right)^2 \leq r\left(\sum_{i=1}^r z_i^2\right) = r\left(\sum_{i=1}^r x^\top w_i w_i^\top x\right) = x^\top \left(r\sum_{i=1}^r w_i w_i^\top\right) x$.

The first "outer" term, involving only the $d \times K$ Jacobian of f_w and the $K \times K$ Hessian of L, is called the **Gauss-Newton (GN) component** of the Hessian. GN is empirically found to be a good approximation of the Hessian (Sankar *et al.*, 2021); it is exact if f_w is linear (since the second term vanishes). Given a population distribution **pop** over (x, y) and assuming the cross-entropy loss, we can further relate GN with the gradient (79).

Lemma H.1. Let $L(z, y) = -\log p_z(y)$. Then

$$\mathop{\mathbf{E}}_{(x,y)\sim\mathbf{pop}}\left[\nabla_w f_w(x)\left(\nabla_z^2 L(z,y)\big|_{z=f_w(x)}\right)\nabla_w f_w(x)^\top\right] = \mathop{\mathbf{E}}_{\substack{x\sim\mathbf{pop}\\ \hat{y}\sim f_w(x)}}\left[\nabla_w L(f_w(x),\hat{y})\nabla_w L(f_w(x),\hat{y})^\top\right] = I(w) \tag{82}$$

The RHS of (82) coincides with the Fisher information matrix $I(w)$ (i.e., the covariance of $\nabla_w L(f_w(x), y)$ where $y \sim f_w(x)$).[8]

Proof of Lemma H.1. Note that regardless of the label $y \in \{1 \dots K\}$, the Jacobian of the cross-entropy loss is the same as the Jacobian of the softmax function:

$$\nabla_z^2 L(z, y) = \nabla_z(p_z - e_y) = \nabla_z p_z = \operatorname{diag}(p_z) - p_z p_z^\top$$

Using the fact that $\mathbf{E}[e_{\hat{y}}] = p_z$ and $\mathbf{E}[e_{\hat{y}} e_{\hat{y}}^\top] = \operatorname{diag}(p_z)$ where $\hat{y} \sim p_z$, we can express this as a vector outer product:

$$\mathop{\mathbf{E}}_{\hat{y}\sim p_z}\left[(p_z - e_{\hat{y}})(p_z - e_{\hat{y}})^\top\right] = \operatorname{diag}(p_z) - p_z p_z^\top$$

Putting together, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{(x,y)\sim\mathbf{pop}}\left[\nabla_w f_w(x)\left(\nabla_z^2 L(z,y)\big|_{z=f_w(x)}\right)\nabla_w f_w(x)^\top\right] &= \mathop{\mathbf{E}}_{x\sim\mathbf{pop}}\left[\nabla_w f_w(x)\left(\operatorname{diag}(p_{f_w(x)}) - p_{f_w(x)} p_{f_w(x)}^\top\right)\nabla_w f_w(x)^\top\right] \\
&= \mathop{\mathbf{E}}_{\substack{x\sim\mathbf{pop}\\ \hat{y}\sim f_w(x)}}\left[\nabla_w f_w(x)(p_{f_w(x)} - e_{\hat{y}})(p_{f_w(x)} - e_{\hat{y}})^\top \nabla_w f_w(x)^\top\right] \\
&= \mathop{\mathbf{E}}_{\substack{x\sim\mathbf{pop}\\ \hat{y}\sim f_w(x)}}\left[\nabla_w L(f_w(x),\hat{y})\nabla_w L(f_w(x),\hat{y})^\top\right]
\end{aligned}$$

where the last equality is the chain rule: $\nabla_w L(f_w(x), \hat{y}) = \nabla_w f_w(x)(\nabla_z L(z, \hat{y})\big|_{z=f_w(x)})$. □

I The Hessian View

Let $w \in \mathbb{R}^d$ denote the weight of (some layer of) a neural network. Given a labeled input $(x, y) \in \mathcal{X} \times \{1 \dots K\}$, let $f_w(x) \in \mathbb{R}^K$ denote the final logits and $L(f_w(x), y) = -\log p_{f_w(x)}(y) \in \mathbb{R}$ the cross-entropy loss (for simplicity we will pretend that this is convex in w). Let $g_{x,y}(w) = \frac{\partial L(f_w(x),y)}{\partial w} \in \mathbb{R}^d$ and $H_{x,y}(w) = \frac{\partial^2 L(f_w(x),y)}{\partial w^2} \in \mathbb{R}^{d\times d}$ the gradient/Hessian on (x, y). Let $g(w) = \mathbf{E}[g_{x,y}(w)]$ and $H(w) = \mathbf{E}[H_{x,y}(w)]$ denote the expected gradient/Hessian where $(x, y) \sim \mathbf{pop}$. We will assume that the fastest way to converge to $w^\star = \arg\min_w E_{(x,y)\sim\mathbf{pop}}[L(f_w(x), y)]$ is Newton's method:

$$w \leftarrow w - \eta H(w)^{-1} g(w) \tag{83}$$

Can we estimate the Hessian using only the gradient? We have

$$H(w) \overset{(81)}{\approx} H_{\mathrm{GN}}(w) \overset{(\mathrm{H.1})}{=} I(w) = \mathop{\mathbf{E}}_{\substack{x\sim\mathbf{pop}\\ \hat{y}\sim f_w(x)}}\left[g_{x,\hat{y}}(w) g_{x,\hat{y}}(w)^\top\right] = \mathop{\mathrm{Cov}}_{\substack{x\sim\mathbf{pop}\\ \hat{y}\sim f_w(x)}}(g_{x,\hat{y}}(w)) \tag{84}$$

where $H_{\mathrm{GN}}(w)$ is the Gauss-Newton component and $I(w)$ is the Fisher matrix. The well-known covariance characterization follows since $\mathbf{E}[g_{x,\hat{y}}(w)] = 0_d$ for $\hat{y} \sim f_w(x)$. The Fisher matrix is difficult to estimate, so we typically approximate it by swapping the model distribution with the label distribution, i.e., the "empirical Fisher" matrix:

$$I_{\mathrm{emp}}(w) = \mathop{\mathbf{E}}_{(x,y)\sim\mathbf{pop}}\left[g_{x,y}(w) g_{x,y}(w)^\top\right] = \mathop{\mathrm{Cov}}_{(x,y)\sim\mathbf{pop}}(g_{x,y}(w)) + g(w) g(w)^\top \tag{85}$$

[8] The expected Hessian $H(w) = E_{(x,y)\sim\mathbf{pop}}[H_{x,y}(w)] = I(w) + E_{(x,y)\sim\mathbf{pop}}[\nabla_w^2 f_w(x)(\nabla_z L(z,y)\big|_{z=f_w(x)})]$ is still not exactly Fisher since the second term does not vanish in general. Nonetheless, many works on second-order optimization assume $H(w) \approx I(w)$.

The motivation is clear: if this is a faithful approximation of $H(w)$, we can easily estimate it by only using the gradient on the labeled data. It is somewhat justified by the fact that $I(w) \to I_{\text{emp}}(w)$ assuming $w \to w^\star$ (since $p_{f_{w^\star}(x)} = \mathbf{pop}(\cdot|x)$ and $g(w^\star) = 0_d$), but in general it is highly flawed (see Section 2.1 of Grosse (2021)) and results in strange behaviors when used directly in (83). For instance, if $g_{x,y}(w)$ happens to have small covariance but large on average, we have $I_{\text{emp}}(w) \approx g(w)g(w)^\top$ which specifies an "inverse gradient scaling" $w \leftarrow w - \eta(g(w)g(w)^\top)^{-1}g(w)$ where the weights with the largest gradient values are updated the *least* (Kunstner *et al.*, 2019).[9] We can heuristically prevent the quadratic inverse scaling by taking the square root, i.e., use

$$H(w) \approx I^{1/2}_{\text{emp}}(w) = \mathop{\mathbf{E}}_{(x,y)\sim\mathbf{pop}} \left[g_{x,y}(w)g_{x,y}(w)^\top\right]^{1/2}$$

Further using the usual diagonal approximation $g_{x,y}(w)g_{x,y}(w)^\top \approx \text{diag}(g^2_{x,y}(w))$ for practicality, we have

$$H(w) \approx I^{1/2}_{\text{emp,diag}}(w) = \mathop{\mathbf{E}}_{(x,y)\sim\mathbf{pop}} \left[\text{diag}\left(g^2_{x,y}(w)\right)\right]^{1/2} = \text{diag}\left(\mathop{\mathbf{E}}_{(x,y)\sim\mathbf{pop}} \left[g^2_{x,y}(w)\right]^{1/2}\right) \tag{86}$$

Let $v \approx \mathbf{E}_{(x,y)\sim\mathbf{pop}}[g^2_{x,y}(w)] \in \mathbb{R}^d$ denote a finite-sample estimator (e.g., Adam uses a bias-corrected EMA of $g^2_{x_1,y_1}(w)\ldots g^2_{x_N,y_N}(w) \in \mathbb{R}^d$). Using the estimator in $w \leftarrow w - \eta I^{-1/2}_{\text{emp,diag}}(w)g(w)$, we obtain the per-parameter update

$$w_j \leftarrow w_j - \frac{\eta}{\sqrt{v_j}}g(w_j) \qquad \text{(RMSProp/Adam)}$$

Since $I^{1/2}_{\text{emp,diag}}(w)$ is also intended to approximate $I(w)$ (not just the Hessian), Adam can be further motivated as an approximation of natural gradient descent $w \leftarrow w - I(w)^{-1}g(w)$ which optimizes the loss in an information-based transformation of the coordinate system (thereby invariant to the underlying geometry).

I.1 The Hessian View of Shampoo

Let $W \in \mathbb{R}^{m\times n}$ denote a weight matrix of a neural network. Let $L : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a random loss function for this weight (i.e., random in data). Let $G = \nabla L(W) \in \mathbb{R}^{m\times n}$. Let $w = \overline{\text{vec}}(W) \in \mathbb{R}^{mn}$ and $g = \overline{\text{vec}}(G) \in \mathbb{R}^{mn}$, with the corresponding reshaped loss $l : \mathbb{R}^{mn} \to \mathbb{R}$ defined as $l(\overline{\text{vec}}(W)) = L(W)$. Newton's method corresponds to $w \leftarrow w - \eta H^{-1}g$ where $H = \nabla^2 l(w) \in \mathbb{R}^{mn\times mn}$. Again we approximate $H \approx I_{\text{fisher}} \approx I^{1/2}_{\text{emp}}$ where $I_{\text{emp}} = \mathbf{E}[gg^\top] \in \mathbb{R}^{mn\times mn}$. Now suppose there are matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{n\times n}$ such that $I_{\text{emp}} \approx A \otimes B$. Then we can update $w \leftarrow w - \eta(A^{-1/2} \otimes B^{-1/2})g$ which is equivalent to

$$W \leftarrow W - \eta A^{-1/2}G(B^{-1/2})^\top \tag{87}$$

by (67). Thus we seek

$$A^\star, B^\star = \mathop{\arg\min}_{A\in\mathbb{R}^{m\times m},\ B\in\mathbb{R}^{n\times n}} \left|\left|I_{\text{emp}} - A \otimes B\right|\right|_F \tag{88}$$

By Corollary G.2, $\overline{\text{vec}}(A^\star) \propto u_1$ and $\overline{\text{vec}}(B^\star) \propto v_1$ where $u_1 \in \mathbb{R}^{m^2}$ and $v_1 \in \mathbb{R}^{n^2}$ are the top singular vectors of $\widetilde{I}_{\text{emp}} = \mathbf{rearrange}(I_{\text{emp}}) \in \mathbb{R}^{m^2\times n^2}$. We can easily verify that $\widetilde{I}_{\text{emp}} = \mathbf{E}[G \otimes G]$ using (69) and (71). We can estimate the top singular vectors of $\widetilde{I}_{\text{emp}}$ by the power method: use some initial $l_0 \in \mathbb{R}^{m^2}$ and $r_0 \in \mathbb{R}^{n^2}$ and repeat $l_{i+1} = \widetilde{I}_{\text{emp}}r_i$ and $r_{i+1} = (\widetilde{I}_{\text{emp}})^\top l_i$. It is well known that $\frac{l_i}{||l_i||} \to u_1$ and $\frac{r_i}{||r_i||} \to v_1$ (assuming $\sigma_1 > \sigma_2$ for simplicity). Now choose $l_0 = \overline{\text{vec}}(I_m)$ and $r_0 = \overline{\text{vec}}(I_n)$. Then one iteration yields

$$\begin{aligned} l_1 &= \mathbf{E}[G \otimes G]r_0 = \overline{\text{vec}}(\mathbf{E}[GG^\top]) \\ r_1 &= \mathbf{E}[G \otimes G]^\top l_0 = \overline{\text{vec}}(\mathbf{E}[G^\top G]) \end{aligned}$$

[9] The problem worsens if we use the batched gradient estimator $g_B(w) = \frac{1}{|B|}\sum_{(x,y)\in B}\frac{\partial L(f_w(x),y)}{\partial w}$ in (85) (which is closer to the practice). Since $\mathbf{E}[g_B(w)] = g(w)$ and $\text{Cov}_B(g_B(w)) = \frac{1}{|B|}\text{Cov}_{x,y}(g_{x,y}(w))$, the empirical Fisher estimator using the batched gradient estimator becomes

$$\mathop{\mathbf{E}}_{B\sim\mathbf{pop}^{|B|}}\left[g_B(w)g_B(w)^\top\right] = \frac{1}{|B|}I_{\text{emp}}(w) + \left(1 - \frac{1}{|B|}\right)g(w)g(w)^\top$$

which shows that $g(w)g(w)^\top$ dominates the estimate as the batch size grows.

Treating these as rough estimates of (some scaling of) u_1 and v_1, we can argue that using (88) in (87) yields (with an appriate η)

$$W \leftarrow W - \eta \mathbf{E}[GG^\top]^{-1/2}\, G\, \mathbf{E}[G^\top G]^{-1/2} \tag{89}$$

This corresponds to using the square of the Shampoo preconditioner $A^2_{\text{shampoo}} = L^{1/2} \otimes R^{1/2}$, since Shampoo with EMA specifies (Section 6.1)

$$W \leftarrow W - \eta \mathbf{E}[GG^\top]^{-1/4}\, G\, \mathbf{E}[G^\top G]^{-1/4}$$

Morwani *et al.* (2024) justify the identity initialization as a way of making $\cos(l_1, u_1) = \frac{(v_1^\top r_0)\sigma_1}{\sqrt{(v_i^\top r_0)^2\sigma_i^2}}$ closer to 1 (similarly for r_1, v_1). Specifically, they show that $r_0 = \overline{\text{vec}}(I_n)$ yields $v_1^\top r_0 > v_i^\top r_0$ for $i \geq 2$.

I.1.1 Exact decomposition

Lemma I.1 (Morwani *et al.* (2024))**.** If $\widetilde{I}_{\text{emp}} = \textbf{rearrange}(I_{\text{emp}}) = \mathbf{E}[G \otimes G] \in \mathbb{R}^{m^2 \times n^2}$ is rank-1,

$$I_{\text{emp}} = \frac{\mathbf{E}[GG^\top] \otimes \mathbf{E}[G^\top G]}{\operatorname{tr}\left(\mathbf{E}[GG^\top]\right)} \in \mathbb{R}^{mn \times mn} \tag{90}$$

Proof. Let $\widetilde{I}_{\text{emp}} = \sigma u v^\top$ be a rank-1 SVD. This implies $I_{\text{emp}} = \sigma U \otimes V$ where $u = \overline{\text{vec}}(U)$ and $v = \overline{\text{vec}}(V)$. Shampoo's iteration gives us $\widetilde{I}_{\text{emp}} r_0 = \sigma u v^\top r_0 = \overline{\text{vec}}(\mathbf{E}[GG^\top])$ where $r_0 = \overline{\text{vec}}(I_n)$. Let $v_\perp = r_0 - \text{Proj}_{\text{span}(v)}(r_0)$ where

$$\text{Proj}_{\text{span}(v)}(r_0) = v v^\top r_0 = (\overline{\text{vec}}(I_n)^\top \overline{\text{vec}}(V)) v = \operatorname{tr}(V)\, v$$

Then

$$\begin{aligned} \overline{\text{vec}}(\mathbf{E}[GG^\top]) &= \sigma u v^\top (v_\perp + \operatorname{tr}(V)\, v) = \sigma \operatorname{tr}(V)\, u \\ \mathbf{E}[GG^\top] &= \sigma \operatorname{tr}(V)\, U \end{aligned}$$

which also implies $\operatorname{tr}\left(\mathbf{E}[GG^\top]\right) = \sigma \operatorname{tr}(V) \operatorname{tr}(U)$. Similarly, $\mathbf{E}[G^\top G] = \sigma \operatorname{tr}(U)\, V$. We now obtain (90) by re-expressing $I_{\text{emp}} = \sigma U \otimes V$. □

The rank-1 assumption in Lemma I.1 is unrealistically strong (likely holds only for linear logistic regressor where $G \in \mathbb{R}^{m \times 1}$). But it suggests the following "idealized" shampoo iteration which corresponds to the Newton step $w \leftarrow w - \eta H^{-1} g$ on $w = \overline{\text{vec}}(W) \in \mathbb{R}^{mn}$ using $H \approx I_{\text{fisher}} \approx I^{1/2}_{\text{emp}} = \frac{1}{\operatorname{tr}(\mathbf{E}[GG^\top])} \mathbf{E}[GG^\top]^{1/2} \otimes \mathbf{E}[G^\top G]^{1/2}$.

```
IdealizedShampooIteration
Input: Current $W \in \mathbb{R}^{m \times n}$, random loss $L : \mathbb{R}^{m \times n} \to \mathbb{R}$, learning rate $\eta > 0$
  1. Compute $G = \nabla L(W) \in \mathbb{R}^{m \times n}$.
  2. Update the estimates $\mathbf{E}[GG^\top] \in \mathbb{R}^{m \times m}$ and $\mathbf{E}[G^\top G] \in \mathbb{R}^{n \times n}$ (e.g., bias-corrected EMA).
  3. $W \leftarrow W - \eta \operatorname{tr}\left(\mathbf{E}[GG^\top]\right)\ \mathbf{E}[GG^\top]^{-1/2}\, G\, \mathbf{E}[G^\top G]^{-1/2}$
```

J Vector Calculus Scratch Pad

Let $w \in \mathbb{R}^d$. We can verify

$$\begin{aligned} \nabla_w g(f(w)) &= (\nabla_w f(w))(\nabla_{f(w)} g(f(w))) && \forall f : \mathbb{R}^d \to \mathbb{R}^M,\ g : \mathbb{R}^M \to \mathbb{R}^K && \textbf{(chain rule)} \\ \nabla_w (F(w) g(w)) &= (\nabla_w F(w)) g(w) + F(w)(\nabla_w g(w))^\top && \forall g : \mathbb{R}^D \to \mathbb{R}^K,\ F : \mathbb{R}^d \to \mathbb{R}^{D \times K} && \textbf{(product rule)} \end{aligned}$$

where $\nabla_w F(w) \in \mathbb{R}^{d \times D \times K}$ is the Jacobian of $F(w) \in \mathbb{R}^{D \times K}$. Let $z = z(w) \in \mathbb{R}^K$ (activations) and $L = L(z) \in \mathbb{R}$ (loss). We write

$$\nabla_w L \in \mathbb{R}^d : (\nabla_w L)_i = \frac{\partial L}{\partial w_i} \qquad \text{(gradient of the loss wrt. the weights)}$$

$$\nabla_z L \in \mathbb{R}^K : (\nabla_z L)_k = \frac{\partial L}{\partial z_k} \qquad \text{(gradient of the loss wrt. the activations)}$$

$$\nabla_w z \in \mathbb{R}^{d \times K} : (\nabla_w z)_{i,k} = \frac{\partial z_k}{\partial w_i} \qquad \text{(Jacobian of the activations wrt. the weights)}$$

$$\nabla_w^2 L \in \mathbb{R}^{d \times d} : (\nabla_w^2 L)_{i,j} = \frac{\partial^2 L}{\partial w_i \partial w_j} \qquad \text{(Hessian of the loss wrt. the weights)}$$

$$\nabla_z^2 L \in \mathbb{R}^{K \times K} : (\nabla_z^2 L)_{k,l} = \frac{\partial^2 L}{\partial z_k \partial z_l} \qquad \text{(Hessian of the loss wrt. the activations)}$$

$$\nabla_w^2 z \in \mathbb{R}^{d \times d \times K} : (\nabla_w^2 z)_{i,j,k} = \frac{\partial^2 z_k}{\partial w_i \partial w_j} \qquad \text{(Hessians of the activations wrt. the weights)}$$

By the chain rule, we have

$$\nabla_w L = (\nabla_w z)(\nabla_z L)$$
$$\nabla_w(\nabla_z L) = (\nabla_w z)(\nabla_z^2 L)$$

By the product rule, we have

$$\nabla_w^2 L = (\nabla_w^2 z)(\nabla_z L) + (\nabla_w z)(\nabla_z^2 L)(\nabla_w z)^\top$$

K Nonnegative Matrix Factorization (NMF)

Let $C \in \mathbb{R}_{\geq 0}^{m \times n}$ and $r \in \mathbb{N}$. We wish to find $A \in \mathbb{R}_{\geq 0}^{m \times r}$ and $B \in \mathbb{R}_{\geq 0}^{r \times n}$ such that $C \approx AB$. A natural divergence between nonnegative values is the I-divergence (Finesso and Spreij, 2006; Lee and Seung, 1999). For any $a, b \geq 0$, the I-divergence is defined as

$$\text{IDiv}(a, b) = a \log \frac{a}{b} - a + b \tag{91}$$

where $\frac{0}{0} = 0$ and $0 \log 0 = 0$. (91) is nonnegative due to the convexity of $x \log x$. For multi-dimensional inputs p, q (of the same shape), we define $\text{IDiv}(p, q) = \sum_{i=1}^n \text{IDiv}(p_i, q_i)$. In particular, $\text{IDiv}(p, q) = \text{KL}(p, q)$ if p, q are distributions. Minimizing $\text{IDiv}(C, AB)$ over nonnegative A, B is equivalent to

$$A^\star, B^\star = \operatorname*{arg\,min}_{A \in \mathbb{R}_{\geq 0}^{m \times r}, B \in \mathbb{R}_{\geq 0}^{r \times n}} \underbrace{\sum_{i=1}^m \sum_{j=1}^n -C_{i,j} \log \left(\sum_{k=1}^r A_{i,k} B_{k,j} \right) + \sum_{k=1}^r A_{i,k} B_{k,j}}_{J_C(A,B)} \tag{92}$$

We have a manifold of optimal solutions $J_C(A^\star, B^\star) = J_C(\alpha A^\star, \frac{1}{\alpha} B^\star)$. The objective is biconvex. Putting aside the nonnegative constraints for now, the gradient is given by (we focus on A since B is analogous):

$$\frac{\partial J_C(A, B)}{\partial A_{i,k}} = -\frac{\sum_{j=1}^n C_{i,j} B_{k,j}}{\sum_{l=1}^r A_{i,l} B_{l,j}} + \sum_{j=1}^n B_{k,j}$$

In general, there is no closed-form solution for a stationary point. We can still do projected gradient descent on A, but a more popular approach is the multiplicative update (95) which preserves nonnegativity.

Rank one. If $r = 1$, the stationary point has a closed-form solution.[10]

$$A_{i,1} = -\frac{\sum_{j=1}^n C_{i,j}}{A_{i,1}} + \sum_{j=1}^n B_{1,j} = 0 \qquad \Leftrightarrow \qquad A_{i,1} = \frac{\sum_{j=1}^n C_{i,j}}{\sum_{j=1}^n B_{1,j}} \qquad \Leftrightarrow \qquad A = \frac{C 1_n}{B 1_n}$$

[10] From the generative perspective of the next section, this happens largely because we remove the "latent variable" and the "sum inside log".

Similarly, we have the stationary $B = \frac{1_m^\top C}{1_m^\top A}$. We may constrain $A \in \mathbb{R}^{m\times 1}$ to satisfy $1_m^\top A = 1_m^\top C 1_n$ (using scale invariance) so that $A = C1_n$ and $B = \frac{1_m^\top C}{1_m^\top C 1_n}$. Since they are nonnegative, they are a solution to (92). Thus rank-one NMF is easy (even though it is still technically nonconvex).

K.1 A Generative Story

We assume a model parameterized by $A \in \mathbb{R}_{\geq 0}^{m\times r}$ and $B \in \mathbb{R}_{\geq 0}^{r\times n}$. It generates the latent variable $Z \in \mathbb{N}_0^{m\times n\times r}$ by

$$Z_{i,j,k} \sim \mathrm{Poi}(A_{i,k}B_{kj})$$

Then it generates the observation $C \in \mathbb{N}_0^{m\times n}$ by $C_{i,j} = \sum_{k=1}^r Z_{i,j,k}$. Since $C_{i,j} \sim \mathrm{Poi}(\sum_{k=1}^r A_{i,k}B_{kj})$ by the usual property of Poisson, the marginal distribution over C is

$$p_{A,B}(C) = \prod_{i,j} \frac{(\sum_{k=1}^r A_{i,k}B_{kj})^{C_{i,j}} e^{-\sum_{k=1}^r A_{i,k}B_{kj}}}{C_{i,j}!}$$

The joint distribution over Z and C satisfying $C_{i,j} = \sum_{k=1}^r Z_{i,j,k}$ is

$$p_{A,B}(Z,C) = \prod_{i,j,k} \frac{(A_{i,k}B_{kj})^{Z_{i,j,k}} e^{-A_{i,k}B_{kj}}}{Z_{i,j,k}!}$$

The posterior over $Z_{i,j} \in \mathbb{N}_0^r$ conditioned on $C_{i,j}$ follows the multinomial distribution (Lemma C.5):

$$p_{A,B}(Z_{i,j}|C_{i,j}) = \mathrm{Mult}\left(C_{i,j}, \left(\frac{A_{i,k}B_{kj}}{\sum_l A_{i,l}B_{lj}}\right)_{k=1}^r\right)(Z_{i,j}) \tag{93}$$

We seek the MLE, i.e., the maximizer of the marginal log-likelihood:

$$\begin{aligned} A^\star, B^\star &= \underset{A\in\mathbb{R}_{\geq 0}^{m\times r}, B\in\mathbb{R}_{\geq 0}^{r\times n}}{\arg\max} \log p_{A,B}(C) \\ &= \underset{A\in\mathbb{R}_{\geq 0}^{m\times r}, B\in\mathbb{R}_{\geq 0}^{r\times n}}{\arg\min} \sum_{i=1}^m \sum_{j=1}^n -C_{i,j}\log\left(\sum_{k=1}^r A_{i,k}B_{kj}\right) + \sum_{k=1}^r A_{i,k}B_{kj} \end{aligned} \tag{94}$$

We see that (94) and (92) are the same. But now that we have a generative story, we can do EM. At any A, B, we can maximize the ELBO using the exact posterior (93) to find

$$\begin{aligned} A', B' &= \underset{\widehat{A}\in\mathbb{R}_{\geq 0}^{m\times r}, \widehat{B}\in\mathbb{R}_{\geq 0}^{r\times n}}{\arg\max} \underset{Z\sim p_{A,B}(\cdot|C)}{\mathbf{E}} \left[\log p_{\widehat{A},\widehat{B}}(Z,C)\right] \\ &= \underset{\widehat{A}\in\mathbb{R}_{\geq 0}^{m\times r}, \widehat{B}\in\mathbb{R}_{\geq 0}^{r\times n}}{\arg\max} \sum_{i,j,k} C_{i,j}\left(\frac{A_{i,k}B_{kj}}{\sum_l A_{i,l}B_{lj}}\right)\log(\widehat{A}_{i,k}\widehat{B}_{k,j}) - \widehat{A}_{i,k}\widehat{B}_{k,j} \end{aligned}$$

As usual with EM, the sum inside log is moved outside. Solving the stationary condition, we have the blockwise update[11]

$$A'_{i,k} = A_{i,k} \times \left(\frac{\sum_j C_{i,j}B_{k,j}}{\sum_l A_{i,l}B_{lj}}\right) \Big/ \left(\sum_j B_{k,j}\right) \qquad B'_{k,j} = B_{k,j} \times \left(\frac{\sum_i C_{i,j}A_{i,k}}{\sum_l A_{i,l}B_{lj}}\right) \Big/ \left(\sum_i A_{i,k}\right) \tag{95}$$

Note that the multiplicative update preserves nonnegativity (assuming A, B are nonnegative). In matrix form, the update is

$$R = C \oslash AB \qquad \begin{aligned} A' &= A \odot (RB^\top \oslash 1_n^\top B^\top) \\ B' &= B \odot (AR^\top \oslash A^\top 1_m) \end{aligned} \tag{96}$$

where $\oslash$ is elementwise division (broadcasted) and $\odot$ is elementwise multiplication.

[11] There's a bit more going on here, since we update one variable while holding the other fixed. This version of EM is so-called "Generalized EM". It simply means breaking the M-step into sub-updates for blocks of parameters; as long as the sub-updates do not decrease the MLL, the convergence property of EM remains.

K.2 AdamNMF

Using (96), we can easily motivate a rank-r generalization of Adafactor (Section 5.1). A pseudocode is given below. The memory overhead in estimating the second gradient moment is $O((m+n)r)$ as opposed to $O(mn)$ in Adam.

```
AdamNMF
Input: initial layer weight $W_1 \in \mathbb{R}^{m \times n}$, rank $r \geq 1$, learning rate $\eta > 0$, initialization range $\epsilon > 0$
  1. $A_0 \sim \mathrm{Unif}(0, \epsilon)^{m \times r}$, $B_0 \sim \mathrm{Unif}(0, \epsilon)^{r \times n}$
  2. For $t = 1 \ldots T$:
      (a) Receive the gradient $G_t \in \mathbb{R}^{m \times r}$, compute the elementwise square $G_t^2$.
      (b) Do one round of EM to decompose $G_t^2 \approx A_t B_t$ using $A_{t-1}$ and $B_{t-1}$ as initialization:
              $R_{t-1} \leftarrow G_t^2 \oslash A_{t-1} B_{t-1}$
              $A_t \leftarrow A_{t-1} \odot \left(R_{t-1} B_{t-1}^\top \oslash 1_n^\top B_{t-1}^\top\right)$
              $B_t \leftarrow B_{t-1} \odot \left(A_{t-1} R_{t-1}^\top \oslash A_{t-1}^\top 1_m\right)$
      (c) $W_{t+1} \leftarrow W_t - \eta \frac{G_t}{\sqrt{A_t B_t}}$
  3. Return $W_{T+1} \in \mathbb{R}^{m \times n}$
```

L Vector Spaces

L.1 Normed Spaces

The function $||\cdot|| : \mathcal{V} \rightarrow \mathbb{R}_{\geq 0}$ is a **norm** on a vector space $\mathcal{V}$ if it satisfies (i) the triangle inequality $||u+v|| \leq ||u|| + ||v||$, (ii) the *absolute* homogeneity $||\alpha u|| = |\alpha| \cdot ||u||$, and (iii) the point-separating property $||u|| = 0 \Rightarrow u = 0_d$. For $\mathcal{V} = \mathbb{R}^d$, a broad family of norms is given by the l_p-norm:

$$||w||_p := \left(\sum_{i=1}^d |w_i|^p\right)^{1/p} \qquad \forall p \geq 1$$

This includes the popular l_2, l_1, l_∞ norms:

$$||w||_2 = \sqrt{\sum_i w_i^2} \qquad \text{(Euclidean)}$$

$$||w||_1 = \sum_i |w_i| \qquad \text{(taxicab)}$$

$$||w||_\infty := \lim_{p \rightarrow \infty} ||w||_p = \max_i |w_i| \qquad \text{(maximum)}$$

On the other hand, the "l_0 norm" defined as $||w||_0 := |\{i : w_i \neq 0\}|$ is often mentioned in the context of promoting sparsity, but it is not a norm (e.g., violates the traingle inequality).

L.2 Inner Product Spaces

The function $\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ is an **inner product** on the (real) vector space $\mathcal{V}$ if it is symmetric, linear in the first argument, and positive-definite (i.e., $\langle u, u \rangle \geq 0$ with equality iff u is zero). An inner product induces a **canonical norm** by $||u|| = \sqrt{\langle u, u \rangle}$, thus an inner product space always a normed space. The most important inner product on $\mathcal{V} = \mathbb{R}^d$ is the **dot product** $\langle u, v \rangle = u^\top v = \sum_i u_i v_i$ which induces the l_2 norm. In contrast, there is no inner product that induces the l_1 or l_∞ norm.

L.2.1 Dual norm

For any norm $||w||$, the **dual norm** $||\cdot||_* : \mathcal{V} \to \mathbb{R}_{\geq 0}$ is defined as[12]

$$||v||_* := \sup_{w \in \mathcal{V}:\ ||w|| \leq 1} \quad w^\top v \tag{97}$$

The definition arises naturally in an effort to bound the dot product since

$$w^\top v \leq ||w||\, ||v||_* \tag{98}$$

for all $v, w \in \mathcal{V}$. (98) is referred to as "generalized Cauchy-Schwarz" or more accurately (the finite-dimensional version of) **Hölder's inequality**. It can be verified that the dual norm is a norm itself and an involution (i.e., $||w||_{**} = ||w||$).

L.3 Weighted Euclidean Norm

For any $d \times d$ positive-definite matrix $A \succ 0$, we define a "weighted Euclidean norm" by

$$||u||_A := \left|\left|A^{1/2} u\right|\right|_2 = \sqrt{u^\top A u} \tag{99}$$

We can directly check that $||\cdot||_A$ is a norm on $\mathcal{V} = \mathbb{R}^d$.[13] To derive the dual norm, we observe

$$\begin{aligned}
||v||_* &= \max_{w \in \mathbb{R}^d:\ w^\top A w = 1} \quad w^\top v \\
&= \max_{u \in \mathbb{R}^d:\ u^\top u = 1} \quad u^\top A^{-1/2} v && (u = A^{1/2} w) \\
&\leq \max_{u \in \mathbb{R}^d:\ u^\top u = 1} \quad ||u||_2 \left|\left|A^{-1/2} v\right|\right|_2 && \text{(Cauchy-Schwarz)} \\
&= \sqrt{v^\top A^{-1} v} \\
&= ||v||_{A^{-1}}
\end{aligned}$$

Choosing $u \propto A^{-1/2} v$ yields a solution that makes the bound tight, thus $||v||_* = ||v||_{A^{-1}}$. See this note for a proof using the method of Lagrangian multipliers.

L.3.1 General $A \succeq 0$

Let $A \succeq 0$ with $r = \text{rank}\,(A) \leq d$. Let $A = V \Lambda V^\top$ denote a thin eigendecomposition where $V \in \mathbb{R}^{d \times r}$ is an orthonormal basis of range (A) and $\Lambda = \text{diag}\,(\lambda_1 \ldots \lambda_r)$ for $\lambda_i > 0$. Pick any $w \in \text{range}\,(A)$. Then $w = Vx$ for some nonzero $x \in \mathbb{R}^r$, so that

$$w^\top A w = x^\top V^\top V \Lambda V^\top V x = x^\top \Lambda x > 0$$

Thus $||u||_A = \sqrt{u^\top A u}$ is a norm on $\mathcal{V} = \text{range}\,(A)$. To derive the dual norm, we can take similar steps:

$$\begin{aligned}
||v||_* &= \max_{w \in \mathbb{R}^d:\ w^\top A w = 1} \quad w^\top v \\
&= \max_{x \in \mathbb{R}^r:\ x^\top \Lambda x = 1} \quad x^\top V^\top v \\
&= \max_{u \in \mathbb{R}^r:\ u^\top u = 1} \quad u^\top \Lambda^{-1/2} V^\top v && (u = \Lambda^{1/2} x) \\
&\leq \max_{u \in \mathbb{R}^d:\ u^\top u = 1} \quad ||u||_2 \left|\left|\Lambda^{-1/2} V^\top v\right|\right|_2 && \text{(Cauchy-Schwarz)} \\
&= \sqrt{v^\top V \Lambda^{-1} V v} \\
&= \sqrt{v^\top A^+ v} \\
&= ||v||_{A^+}
\end{aligned}$$

[12] This is a *different* definition of the dual norm from Hilbert spaces (i.e., inner product spaces in infinite dimensions). There, the dual norm is defined as $||v||_* := \sup_{w \in \mathcal{V}:\ ||w|| \leq 1} \langle w, v \rangle$. One can verify that $||v||_* = ||v||$ ("self-dual") using the standard Cauchy-Schwarz inequality $|\langle u, v \rangle| \leq ||u||\, ||v||$. (The Cauchy-Schwarz inequality can be proved directly without dual norms, so there is no circular argument here.)

[13] We can also view $||u||_A$ as the canonical norm of the inner product $\langle u, v \rangle_A := u^\top A v$ on $\mathcal{V} = \mathbb{R}^d$.

We can again verify that choosing $u \propto \Lambda^{-1/2} V^\top v$ achieves this bound, thus $||v||_* = ||v||_{A^+}$. This subsumes the above analysis when $r = d$. When $r < d$ (i.e., A is rank-deficient), we have $\mathbb{R}^d = \text{range}\,(A) \perp \text{null}(A)$ (since A is symmetric) with a nontrivial null space. In particular, there exist nonzero $w \in \text{null}(A)$ such that $||w||_A = \sqrt{w^\top A w} = 0$, thus $||\cdot||_A$ fails to satisfy the point-separating property on null(A) (i.e., it becomes a "seminorm" on $\mathcal{V} = \mathbb{R}^d$). Pick any nonzero $v \in \text{null}(A)$. Assuming $r > 0$, we can select some $w_0 \in \text{range}\,(A)$ such that $w_0^\top A w_0 = 1$. Define $w(\alpha) = w_0 + \alpha v$ and note that $w(\alpha)^\top A w(\alpha) = 1$ for all $\alpha \in \mathbb{R}$. Thus

$$\begin{aligned} ||v||_* &= \max_{w \in \mathbb{R}^d:\, w^\top A w = 1} \; w^\top v \\ &\geq \max_{\alpha \in \mathbb{R}} \; w(\alpha)^\top v \\ &= \max_{\alpha \in \mathbb{R}} \; w_0^\top v + \alpha \, ||v||_2^2 \\ &= \infty \end{aligned}$$

(i.e., $||v||_*$ is not finite for $v \notin \text{range}\,(A)$.)