Adaptive Learning Methods

Karl Stratos

1 Online Convex Optimization (OCO)

We consider an online learning environment in which an algorithm, at each step $t = 1, 2, 3, \ldots$, proposes a hypothesis $w_t \in \mathbb{R}^d$ then immediately gets punished by $l_t(w_t) \in \mathbb{R}$ where $l_t : \mathbb{R}^d \to \mathbb{R}$ is some *unknown* loss function (possibly adversarial, i.e., crafted against w_t). In the beginning, the algorithm has no information and uses some initial hypothesis w_1; at step $t \geq 1$, it may use all the information so far to propose w_{t+1}. Let $u = \arg\min_{w \in \mathbb{R}^d} \sum_{t=1}^T l_t(w)$ denote the best hypothesis *in retrospect*. We want to bound the "regret" of the algoirthm as a function of the total number of steps T:

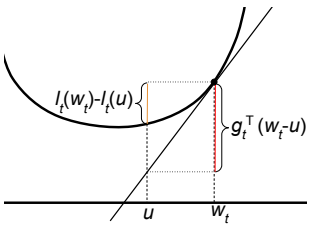
$$\sum_{t=1}^T l_t(w_t) - l_t(u) \leq B(T)$$

A good algorithm has a sublinear regret $B(T) = o(T)$; it "converges" in the sense that the average regret goes to zero (i.e., $\frac{\text{regret}}{T} \to 0$ as $T \to \infty$). This seems impossible, but we can actually do it with certain assumptions and a right update! The key is to elicit a "telescoping sum" to cancel a naive accumulation of regret over T steps. Right now there is no telescoping; none of the terms in

$$\sum_{t=1}^T l_t(w_t) - l_t(u) = l_1(w_1) - l_1(u) + l_2(w_2) - l_2(u) + \cdots + l_T(w_T) - l_T(u)$$

necessarily cancel. We need a way to abstract l_t so that we can relate them across steps. One way to achieve this abstraction is assuming that every l_t is **convex** and **differentiable** (easily extended to sub-differentiable). Let $g_t = \nabla l_t(w_t)$ denote the gradient of l_t at w_t. By the property of convexity:

$$l_t(w_t) - l_t(u) \leq \langle g_t, w_t - u \rangle$$



We further relate $\langle g_t, w_t - u \rangle$ to w_{t+1} by proposing the update rule

$$w_{t+1} = w_t - \eta g_t \tag{1}$$

where $\eta > 0$ is some learning rate. Together, they give a per-step regret bound measured by progress in Euclidean space, not dependent on the particular loss function l_t (except for a gradient term):

$$l_t(w_t) - l_t(u) \leq \frac{1}{2\eta} \left(||w_t - u||^2 - ||w_{t+1} - u||^2 \right) + \frac{\eta}{2} ||g_t||^2$$

Since the next per-step bound involves $||w_{t+1} - u||^2$, telescoping happens! We typically need an additional assumption that the gradients are bounded to bound the last term.

A mind-boggling observation is that the motivation (1) for the gradient-based update rule has nothing to do with the usual "steepest descent" interpretation of the negative gradient.

1.1 Fixed Learning Rate

We write $||v||_A = \sqrt{v^\top A v}$ for $A \succ 0$ (verify that this is a norm).

Lemma 1.1. Pick any $A \succ 0$ and $\eta > 0$. For $t = 1 \dots T$ pick $w_{t+1} = w_t - \eta A^{-1} g_t$. Then

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \frac{1}{2\eta} ||w_1 - u||_A^2 + \frac{\eta}{2} \sum_{t=1}^{T} ||g_t||_{A^{-1}}^2 \tag{2}$$

Proof. For any $t = 1 \dots T$

$$\begin{aligned} ||w_t - u||_A^2 - ||w_{t+1} - u||_A^2 &= ||w_t - u||_A^2 - \left|\left|w_t - \eta A^{-1} g_t - u\right|\right|_A^2 \\ &= 2\eta \langle g_t, w_t - u \rangle - \eta^2 ||g_t||_{A^{-1}}^2 \end{aligned}$$

By the convexity of l_t

$$l_t(w_t) - l_t(u) \leq \langle g_t, w_t - u \rangle$$

Rearranging, we have

$$l_t(w_t) - l_t(u) \leq \frac{1}{2\eta} \left(||w_t - u||_A^2 - ||w_{t+1} - u||_A^2 \right) + \frac{\eta}{2} ||g_t||_{A^{-1}}^2$$

Summing both sides over $t = 1 \dots T$, we have

$$\begin{aligned} \sum_{t=1}^{T} l_t(w_t) - l_t(u) &\leq \frac{1}{2\eta} \left(||w_1 - u||_A^2 - ||w_{T+1} - u||_A^2 \right) + \frac{\eta}{2} \sum_{t=1}^{T} ||g_t||_{A^{-1}}^2 \\ &\leq \frac{1}{2\eta} ||w_1 - u||_A^2 + \frac{\eta}{2} \sum_{t=1}^{T} ||g_t||_{A^{-1}}^2 \end{aligned}$$

□

1.2 Stepwise Learning Rate

Lemma 1.2. Pick any $A \succ 0$ and $\eta_1 \geq \dots \geq \eta_T > 0$. Select any closed convex set $V_A \subseteq \mathbb{R}^d$ with finite diameter $D_A \geq \max_{x,y \in V_A} ||x - y||_A$ big enough to contain u, w_1. For $t = 1 \dots T$ pick

$$w_{t+1} = \arg\min_{w \in V_A} \left|\left|w - (w_t - \eta_t A^{-1} g_t)\right|\right|_A^2 \tag{3}$$

Then

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \frac{D_A^2}{2\eta_T} + \frac{1}{2} \sum_{t=1}^{T} \eta_t ||g_t||_{A^{-1}}^2 \tag{4}$$

Proof. Let $z_{t+1} = w_t - \eta_t A^{-1} g_t$ and $\Pi_{V,A}(w) = \arg\min_{x \in V_A} ||w - x||_A^2$ so that $w_{t+1} = \Pi_{V,A}(z_{t+1})$. Since $u \in V_A$, by Lemma A.4:

$$||w_{t+1} - u||_A^2 = ||\Pi_{V,A}(z_{t+1}) - u||_A^2 \leq ||z_{t+1} - u||_A^2$$

Using this observation and taking the same steps in the proof of Lemma 1.1, we have

$$l_t(w_t) - l_t(u) \leq \frac{1}{2\eta_t} \left(||w_t - u||_A^2 - ||w_{t+1} - u||_A^2 \right) + \frac{\eta_t}{2} ||g_t||_{A^{-1}}^2$$

Summing both sides over $t = 1 \dots T$, we have

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \frac{1}{2} \underbrace{\left(\sum_{t=1}^{T} \frac{1}{\eta_t} ||w_t - u||_A^2 - \frac{1}{\eta_t} ||w_{t+1} - u||_A^2 \right)}_{①} + \frac{1}{2} \sum_{t=1}^{T} \eta_t ||g_t||_{A^{-1}}^2 \tag{5}$$

The stepwise learning rate now prevents the nice telescoping sum before, but we can write

$$\begin{aligned}
\textcircled{1} &= \frac{1}{\eta_1}\left|\left|w_1-u\right|\right|_A^2 + \sum_{t=2}^{T}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}\right)\left|\left|w_t-u\right|\right|_A^2 - \frac{1}{\eta_T}\left|\left|w_{T+1}-u\right|\right|_A^2 \\
&\leq \frac{1}{\eta_1}\left|\left|w_1-u\right|\right|_A^2 + \sum_{t=2}^{T}\underbrace{\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}\right)}_{\geq 0}\left|\left|w_t-u\right|\right|_A^2 \\
&\leq \frac{D_A^2}{\eta_1} + D_A^2\sum_{t=2}^{T}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}\right) \qquad (6) \\
&= \frac{D_A^2}{\eta_T}
\end{aligned}$$

Note that in (6), we need to use not only the fact that $w_t, u \in V$ (to bound the squared norms) but also the fact that $\eta_t \leq \eta_{t-1}$ to bound the sum. Using this bound on (5), we have the statement. □

Remark 1.3. Lemma 1.2 assumes a choice of finite convex set V_A containing u, w_1 and a projection onto V_A in the update (3) to make the analysis simpler. Although this theoretical construct is a standard setting in OCO, it is detached from the real-world practice which does not involve such an explicit projection step.[1] We can effectively remove the projection step by choosing a very large V_A (e.g., with diameter equal to the maximum float value on a physical computer), but this will make the bound (4) meaningless.

1.3 Warmup: Stochastic Gradient Descent (SGD)

SGD with a fixed learning rate $\eta > 0$ dictates $w_{t+1} = w_t - \eta g_t$ and has the regret bound by Lemma 1.1

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \frac{1}{2\eta}\left|\left|w_1-u\right|\right|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\left|\left|g_t\right|\right|^2 = \frac{D^2}{2\eta} + \frac{\eta T L^2}{2}$$

where we write $D = ||w_1 - u||$ and $L = \max_{t=1}^{T} ||g_t||$. The optimal learning rate is $\eta^\star = \frac{D/L}{\sqrt{T}}$ yielding the sublinear regret $DL\sqrt{T}$. But note that any learning rate proportional to $\frac{1}{\sqrt{T}}$ achieves a sublinear regret.

If we do not know how many steps await in the future (i.e., T is unknown), we can still achieve a sublinear regret by using the per-step size $\eta_t \propto \frac{1}{\sqrt{t}}$, thanks to the handy inequality $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}$. By Lemma 1.2, assuming a big enough convex set with diameter D to ignore the projection step, SGD with a stepwise learning rate $w_{t+1} = w_t - \eta_t g_t$ where $\eta_t = \frac{1}{\sqrt{t}}$ has the regret bound

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \frac{D^2}{2\eta_T} + \frac{L}{2}\sum_{t=1}^{T}\eta_t \leq \frac{D^2\sqrt{T}}{2} + L\sqrt{T}$$

2 AdaGrad

AdaGrad (Duchi *et al.*, 2011) specifies

$$w_{t+1} = w_t - \eta_t \mathrm{diag}\left(\sum_{l=1}^{t} g_l^2\right)^{-1/2} g_t$$

where u^2 denotes the elementwise square of vector u. The i-th parameter is updated by

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

Note that $w_{2,i} = w_{1,i} - \eta_1$ and the magnitude of the update is always at most η_t. If the parameter has been heavily updated in the past, it will receive a smaller update. For this reason, AdaGrad is particularly effective when the updates are sparse (i.e., only a few parameters are updated at a single step).

[1] Alternatively, we may view some regularization tricks like weight decay as a soft version of projection.

2.1 Motivation I

Recall Lemma 1.1: we may update $w_{t+1} = w_t - \eta A^{-1} g_t$ using any $A \succ 0$ and $\eta > 0$ and achieve the regret bound

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \le \frac{1}{2\eta} ||w_1 - u||_A^2 + \frac{\eta}{2} \sum_{t=1}^{T} ||g_t||_{A^{-1}}^2$$

Consider minimizing the bound over A. Since u is unknown, we optimize only the second term $\sum_{t=1}^{T} ||g_t||_{A^{-1}}^2$. But this alone is trivially minimized to zero by setting $A = \alpha I_{d\times d}$ and taking $\alpha \to \infty$, ignoring the fact that it blows up the first term $||w_1 - u||_A^2$. Thus we must additionally constrain the size of A. Choose some $c > 0$ and define

$$A^\star = \operatorname*{arg\,min}_{A \succ 0:\ \mathrm{tr}(A) \le c} \sum_{t=1}^{T} g_t^\top A^{-1} g_t$$

The following result is by Lagrangian relaxation for generalized inequalities (see Lemma C.3 in this note).

Lemma 2.1. Define $G_T = (\sum_{t=1}^{T} g_t g_t^\top)^{1/2}$. Then $A^\star = \frac{c}{\mathrm{tr}(G_T)} G_T$, yielding an adaptive regret bound

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \le \frac{1}{2\eta} ||w_1 - u||_{A^\star}^2 + \frac{\eta}{2c} \mathrm{tr}\left(G_T\right)^2$$

This is not implementable since it requires future gradients, but we may approximate $A^\star \propto (\sum_{t=1}^{T} g_t g_t^\top)^{1/2}$ with $A_t = (\sum_{l=1}^{t} g_l g_l^\top)^{1/2}$ that accumulates the gradients up to the current step. Further using a diagonal approximation of A_t, we recover the AdaGrad update (with a fixed learning rate).

2.2 Motivation II

Consider performing vanilla SGD, that is $w_{t+1} = w_t - \eta g_t$ for some $\eta > 0$. By Lemma 1.1,

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \le \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} ||g_t||^2$$

where we pick some $D \ge ||w_1 - u||$. Let us optimize η without assuming a uniform bound on the gradients as in Section 1.3. The optimal choice is

$$\eta^\star = \frac{D}{\sqrt{\sum_{t=1}^{T} ||g_t||^2}} \tag{7}$$

yielding the adaptive regret bound

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \le D \sqrt{\sum_{t=1}^{T} ||g_t||^2} \tag{8}$$

Again, (7) is not implementable since it involves future gradients. But can we approximate it with a per-step partial sum? Namely

$$\eta_t = \frac{D}{\sqrt{\sum_{l=1}^{t} ||g_l||^2}} \tag{9}$$

The following fact is a generalization of $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \le 2\sqrt{T}$ and useful for this purpose (the proof is easy by induction).

Fact 2.2. Pick any $T \ge 1$ and $b_1 \dots b_T \ge 0$. Denote the partial sum by $B_t = \sum_{l=1}^{t} b_l$. Then

$$\sum_{t=1}^{T} \frac{b_t}{\sqrt{B_t}} \le 2\sqrt{B_T}$$

Lemma 2.3. Select any closed convex set $V \subseteq \mathbb{R}^d$ with finite diameter $D \geq \max_{x,y \in V} ||x - y||$ big enough to contain u, w_1. Assume $g_1 \neq 0_d$. For $t = 1 \dots T$, compute

$$w_{t+1} = \arg\min_{w \in V} \left|\left| w - \left(w_t - \frac{\frac{\sqrt{2}}{2} D}{\sqrt{\sum_{l=1}^{t} ||g_l||^2}} g_t \right) \right|\right|^2 \tag{10}$$

Then

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \sqrt{2} D \sqrt{\sum_{t=1}^{T} ||g_t||^2}$$

Proof. Since the stepwise learning rate is positive and nonincreasing

$$\begin{aligned}
\sum_{t=1}^{T} l_t(w_t) - l_t(u) &\leq \frac{D^2}{2\eta_T} + \frac{1}{2} \sum_{t=1}^{T} \eta_t ||g_t||^2 && \text{(Lemma 1.2)} \\
&= \frac{D}{\sqrt{2}} \sqrt{\sum_{t=1}^{T} ||g_t||^2} + \frac{D\sqrt{2}}{4} \sum_{t=1}^{T} \frac{||g_t||^2}{\sqrt{\sum_{l=1}^{t} ||g_l||^2}} \\
&\leq \sqrt{2} D \sqrt{\sum_{t=1}^{T} ||g_t||^2} && \text{(Fact 2.2)}
\end{aligned}$$

□

Thus the per-step partial sum approximation yields a regret that is only $\sqrt{2} \approx 1.4$ times worse. We scaled (9) by $\frac{\sqrt{2}}{2}$ to slightly improve the constant (1.5 without the scaling).

2.2.1 Per-parameter version

Lemma 2.4. Select any closed convex set $V \subseteq \mathbb{R}^d$ with finite diameter $D \geq \max_{x,y \in V} ||x - y||$ big enough to contain u, w_1. For each $i = 1 \dots d$, let $a_i = \min_{x \in V} x_i$, $b_i = \max_{x \in V} x_i$, and $D_i = b_i - a_i$; also assume $g_{1,i} \neq 0$. For $t = 1 \dots T$, compute

$$w_{t+1,i} = \arg\min_{z \in [a_i, b_i]} \left(z - \left(w_{t,i} - \frac{\frac{\sqrt{2}}{2} D_i}{\sqrt{\sum_{l=1}^{t} g_{l,i}^2}} g_{t,i} \right) \right)^2 \tag{11}$$

for $i = 1 \dots d$. Then

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

Proof. (11) is an application of (10) to *any* single-parameter loss $f_{t,i} : \mathbb{R} \to \mathbb{R}$ satisfying $f'_{t,i}(w_{t,i}) = g_{t,i}$, in particular the linear loss $x \mapsto g_{t,i} x$.[2] From Lemma 2.3, it follows that

$$\sum_{t=1}^{T} g_{t,i} w_{t,i} - g_{t,i} u_i \leq \sqrt{2} D_i \sqrt{\sum_{t=1}^{T} g_{t,i}^2} \tag{12}$$

We now use the basic fact in OCO that the convex regret is upper bounded by the "linearized" regret

$$\sum_{t=1}^{T} l_t(w_t) - l_t(u) \leq \sum_{t=1}^{T} \langle g_t, w_t \rangle - \langle g_t, u \rangle = \sum_{i=1}^{d} \sum_{t=1}^{T} g_{t,i} w_{t,i} - g_{t,i} u_i \tag{13}$$

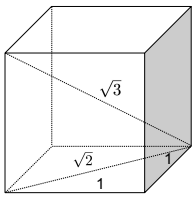
Combining (12) and (13) gives the statement. □

[2] Note that the gradient $g_t = \nabla l_t(w_t)$ is given by the update (11); we are using this retrospectively to construct a loss under which the update remains the same.

Discussion. Applying the "global AdaGrad" to individual parameters as if we have d single-parameter OCO problems yields the per-parameter AdaGrad:

$$w_{t+1} = \arg\min_{w \in V} \left|\left| w - \left(w_t - \frac{\frac{\sqrt{2}}{2} D}{\sqrt{\sum_{l=1}^{t} ||g_l||^2}} g_t \right) \right|\right|^2 \quad \text{vs.} \quad w_{t+1,i} = \arg\min_{z \in [a_i, b_i]} \left(z - \left(w_{t,i} - \frac{\frac{\sqrt{2}}{2} D_i}{\sqrt{\sum_{l=1}^{t} g_{l,i}^2}} g_{t,i} \right) \right)^2$$

where D is the diameter of V and $D_i = b_i - a_i$ is the diameter of V in the i-th dimension. Intuitively, the latter should be able to give us a tighter bound in some cases by making more fine-grained changes. More specifically, suppose $V = [0, 1]^d$ (i.e., a hypercube). In this case $D_i = 1$ while $D = \sqrt{d}$, for instance with $d = 3$:



Thus the global AdaGrad has the regret guarantee of (omitting the constant $\sqrt{2}$) $\sqrt{d}\sqrt{\sum_t ||g_t||^2}$ (Lemma 2.3) whereas the per-parameter AdaGrad has $\sum_i \sqrt{\sum_t \bar{g}_{t,i}^2}$ (Lemma 2.4). Here, the gradients g_t and $\bar{g}_t$ under the two updates are generally different! But, for illustration purposes assuming that they are the same we have the following inequalities

$$\sqrt{\sum_t ||g_t||^2} \leq \sum_i \sqrt{\sum_t g_{t,i}^2} \leq \sqrt{d} \sqrt{\sum_t ||g_t||^2}$$

(the first is $\sqrt{\sum_i x_i} \leq \sum_i \sqrt{x_i}$, the second is the Cauchy-Schwarz inequality $\langle v_1, v_2 \rangle \leq ||v_1||\,||v_2||$). This shows that the per-parameter update can be a factor of $\sqrt{d}$ better than the global update depending on the geometry of the convex set and the gradient sequence.

3 Adam

AdaGrad has inspired a whole class of per-parameter adaptive updates that scale the gradient by the square root of the sum of past squared gradients. One practical issue of AdaGrad is that the update can only become smaller throughout training because the denominator in

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

can only become larger. This is fine for convex problems (from which AdaGrad is derived) 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 $v_0 \in \mathbb{R}^d$ is some initial value (typically zero) and $\beta_2 \in [0, 1]$ is a momentum coefficient (i.e., how much to remember). 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 some $m_0 \in \mathbb{R}^d$ and $\beta_1 \in [0, 1]$. Plugging these in the AdaGrad update, we have RMSProp with momentum (Tieleman *et al.*, 2012; Graves, 2013):

$$w_{t+1,i} = w_{t,i} - \frac{\eta_t}{\sqrt{v_{t,i}}} m_{t,i}$$

If v_0, m_0 are zero and β_1, β_2 are large, the initial updates will be very small until they gain some momentum. One way to formulate a solution to this problem is to assume a stochastic setting in which the gradient $g_t \sim \mathcal{G}(g)$ at each step is sampled from some stationary distribution with mean $g \in \mathbb{R}^d$. Our goal now to recover g for use in the AdaGrad update. At any step t, assuming $\beta_1, \beta_2 > 0$ by the property of EMA (Corollary A.2)

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

Since β_1, β_2 are typically close to 1, this bias correction greatly upweights the EMA estimates initially. This yields Adam (Kingma and Ba, 2014):

$$w_{t+1,i} = w_{t,i} - \frac{\eta_t}{\sqrt{\bar{v}_{t,i}}}\bar{m}_{t,i} \qquad\qquad \bar{m}_t = \frac{1}{1-\beta_1^t}m_t, \quad \bar{v}_t = \frac{1}{1-\beta_2^t}v_t \tag{14}$$

3.1 Scale Invariance

A property of an 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, if we multiply g_t elementwise by some $c \in \mathbb{R}^d$ in Adam, we have

$$w_{t+1,i} = w_{t,i} - \frac{\eta_t}{\sqrt{c_i^2\bar{v}_{t,i}}}c_i\bar{m}_{t,i} = w_{t,i} - \frac{\eta_t}{\sqrt{\bar{v}_{t,i}}}\bar{m}_{t,i}$$

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 (Section 3.4)

$$w_{t+1,i} = w_{t,i} - \frac{\eta_t}{\sqrt{c_i^2\bar{v}_{t,i}} + \epsilon}c_i\bar{m}_{t,i}$$

which is not scale invariant for $\epsilon > 0$. But since ϵ is typically minuscule, scale invariance is approximately preserved in practice (Zhuang *et al.*, 2022).

3.2 Convergence

In the proof of AdaGrad's convergence (Lemma 2.3), we use the fact that the learning rate is nonincreasing (Lemma 1.2). This is no longer true in EMA-based updates like RMSProp and Adam. In fact, there is an OCO 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

$$\bar{v}_t^{\max} = \max\left\{\bar{v}_{t-1}^{\max}, \bar{v}_t\right\} \qquad\qquad \text{(AMSGrad)}$$

where $\bar{v}_0^{\max} = 0_d$ and use $\bar{v}_t^{\max}$ in place of $\bar{v}_t$ in the update (14). 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).

3.3 Weight Decay

Weight decay is originally proposed as a regularization technique separate from the l_2-norm regularization (Hanson and Pratt, 1988). In SGD, they coincide:

$$w_{t+1} = \underbrace{w_t - \eta_t\nabla l_t(w_t) - \eta_t\lambda w_t}_{\text{SGD with weight decay}} = \underbrace{w_t - \eta_t\nabla\left(l_t(w_t) - \frac{\lambda}{2}\left|\left|w_t\right|\right|^2\right)}_{\text{SGD with } l_2 \text{ regularization (no weight decay)}}$$

In general with pre-conditioning, they do not coincide. The l_2-norm regularization with strength λ' is in this case

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

The last term is not equal to $\lambda\eta_t w_t$ (for any given weight decay strength λ) no matter what λ' we choose unless A is some scaling matrix (i.e., $A = cI_{d\times d}$). It is thus customary to have an explicit weight decay on top of Adam ("AdamW" (Loshchilov and Hutter, 2017)).

3.4 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)
- Learning rate schedule $\eta_1, \eta_2, \ldots, \eta_T > 0$
- Momentum coefficients (β_1, β_2) (defaults to $(0.9, 0.999)$)
- Smoothing coefficient $\epsilon \geq 0$ (defaults to 10^{-8})
- Weight decay coefficient $\lambda \geq 0$ (defaults to 0)
- Flag for using AMSGrad

1. Initialize the moving averages $(m_0, v_0, \bar{v}_0^{\max}) \leftarrow (0_d, 0_d, 0_d)$.
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:

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t \qquad v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$
$$\bar{m}_t \leftarrow \frac{1}{1-\beta_1^t} m_t \qquad \bar{v}_t \leftarrow \frac{1}{1-\beta_2^t} v_t$$

   (c) Decide if we will do AMSGrad:

$$\hat{v}_t \leftarrow \begin{cases} \bar{v}_t^{\max} & \text{if AMSGrad} \\ \bar{v}_t & \text{otherwise} \end{cases} \qquad \bar{v}_t^{\max} \leftarrow \max\{\bar{v}_{t-1}^{\max}, \bar{v}_t\}$$

   (d) Compute the per-parameter update for $i = 1 \ldots d$:

$$w_{t+1,i} \leftarrow w_{t,i} - \frac{\eta_t}{\sqrt{\hat{v}_{t,i}} + \epsilon} \bar{m}_{t,i} - \eta_t \lambda w_{t,i}$$

3. Return $w_{T+1} \in \mathbb{R}^d$

Note the extra hyperparameter ϵ that helps with numerical stability in case the squared gradient EMA is too small. There are some works, notably the original Transformer paper (Vaswani *et al.*, 2017), that use $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ instead of the default values. Weight decay with $\lambda = 0.01$ seems popular in some language modeling experiments, notably BERT (Devlin *et al.*, 2018) (also see the Fairseq example). It typically helps to have an explicit learning rate schedule which starts from some small initial value, heats it up to the max value (typically of form $c10^{-p}$ where $c \in \{5, 2, 1\}$ and $p \in \{3, 4, 5\}$ need to be tuned) after a specified number of warmup steps (e.g., $\frac{T}{10}$), then cools it down to some small final value. This seems to alleviate fluctuations caused by variations between minibatches in the early and late phases of training. The rate of heating and cooling depends on the choice of schedule (e.g., linear increase followed by inverse square root decrease).

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

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

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

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*.

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

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

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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

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

A Lemmas

Lemma A.1. Let $X_1, X_2, \ldots \in \mathbb{R}^d$ denote independent random vectors with means $\mu_1, \mu_2, \ldots \in \mathbb{R}^d$. Pick constants $\beta \in [0, 1]$ and $u \in \mathbb{R}^d$. Define

$$\begin{aligned} Z_0 &:= u \\ Z_t &:= \beta Z_{t-1} + (1-\beta) X_t && \forall t \geq 1 \end{aligned}$$

Then

$$\mathbf{E}\left[Z_t\right] = \mu_t + \beta^t (u - \mu_t) + (1-\beta)\left(\sum_{i=1}^{t-1} \beta^{t-i}(\mu_i - \mu_t)\right) \qquad \forall t \geq 1$$

Proof. By the observation

$$\begin{aligned} Z_1 &= \beta u + (1-\beta) X_1 \\ Z_2 &= \beta^2 u + \beta(1-\beta) X_1 + (1-\beta) X_2 = \beta^2 u + (1-\beta)(\beta X_1 + X_2) \\ Z_3 &= \beta^3 u + \beta(1-\beta)(\beta X_1 + X_2) + (1-\beta) X_3 = \beta^3 u + (1-\beta)(\beta^2 X_1 + \beta X_2 + X_3) \\ &\vdots \end{aligned}$$

we can write

$$Z_t = \beta^t u + (1-\beta)\left(\sum_{i=1}^{t} \beta^{t-i} X_i\right)$$

Thus

$$\begin{aligned} \mathbf{E}\left[Z_t\right] &= \mathbf{E}\left[\beta^t u + (1-\beta)\left(\sum_{i=1}^{t} \beta^{t-i} X_i\right)\right] \\ &= \beta^t u + (1-\beta)\left(\sum_{i=1}^{t} \beta^{t-i} \mu_i\right) && \text{(linearity of expectation)} \\ &= \beta^t u + (1-\beta)\left(\sum_{i=1}^{t} \beta^{t-i}\right)\mu_t + (1-\beta)\left(\sum_{i=1}^{t} \beta^{t-i}\left(\mu_i - \mu_t\right)\right) \end{aligned}$$

The middle term involves a telescoping sum:

$$(1-\beta)\left(1+\beta+\cdots+\beta^{t-2}+\beta^{t-1}\right) = 1-\beta+\beta-\beta^2+\cdots+\beta^{t-1}-\beta^{t-1}+\beta^{t-1}-\beta^t = 1-\beta^t$$

This implies the statement.

□

Corollary A.2. Let $X_1, X_2, \ldots \in \mathbb{R}^d$ denote independent random vectors with mean $\mu \in \mathbb{R}^d$. Pick $\beta \in [0,1)$ and define

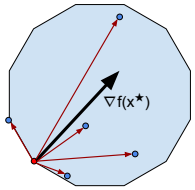
$$\begin{aligned} Z_0 &:= 0_d \\ Z_t &:= \beta Z_{t-1} + (1-\beta) X_t \qquad\qquad \forall t \geq 1 \end{aligned}$$

Then

$$\mu = \frac{1}{1-\beta^t}\mathbf{E}\left[Z_t\right] \qquad\qquad \forall t \geq 1$$

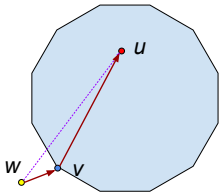
Fact A.3. Let $V \subseteq \mathbb{R}^d$ be a nonempty closed convex set and $f : \mathbb{R}^d \to \mathbb{R}$ a convex function differentiable on V. Then

$$x^\star = \arg\min_{x \in V} f(x) \quad \Leftrightarrow \quad \langle \nabla f(x^\star), y - x^\star \rangle \geq 0 \qquad \forall y \in V$$



Lemma A.4. Let $V \subseteq \mathbb{R}^d$ be a nonempty closed convex set. Let $A \succ 0$. Pick any $w \in \mathbb{R}^d$ and let $v = \arg\min_{x \in V} ||w - x||_A^2$. Then

$$||u - v||_A^2 \leq ||u - w||_A^2 \qquad \forall u \in V$$



Proof. The proof is a version of Pythagorean theorem for nonorthogonal vectors:

$$||u - w||_A^2 = ||u - v + v - w||_A^2 = ||u - v||_A^2 + ||v - w||_A^2 + 2\langle A(u-v), v - w \rangle$$

To show the last term is nonnegative, note that the gradient of $f(x) = ||w - x||_A^2 = \langle Aw - Ax, w - x \rangle$ is $\nabla f(x) = 2A(w-x)$. Since f is a convex function and v minimizes f within V, $\langle 2A(w-v), y - v \rangle \geq 0$ for all $y \in V$ (Fact A.3). Using $y = u$ and rearranging, we have $2\langle A(u-v), v-w \rangle \geq 0$. Thus

$$\begin{aligned} ||u - w||_A^2 \geq ||u - v||_A^2 + ||v - w||_A^2 \qquad &\Leftrightarrow \qquad ||u - v||_A^2 \leq ||u - w||_A^2 - ||v - w||_A^2 \\ &\Rightarrow \qquad ||u - v||_A^2 \leq ||u - w||_A^2 \end{aligned}$$

□