Automatic Step-Size Control for Minimization Iterations 28 Aug. 2019

Introduction

We seek a minimizing vector x for a smooth function ƒ(x) subject to memory constraints that allow only the computations of ƒ(x) and its Gradient G(x) := dƒ(x)/dx`, the transposed Jacobian array of first derivatives. Limited memory precludes access to the Hessian array H(x) := $d^2ƒ(x)/dx^2$ of second derivatives, and precludes estimates of H from differences among very many stored instances of { G(x), x } .

Since ƒ(x) is not presumed convex, more than one locally minimizing x may exist; finding the best may entail searches for several. What determines searches' speeds? How can speeds be improved? How much?

Questions like those arise during "Training of Deep Learning" (DL) for applications of Artificial Intelligence (AI). Answers are needed that assume no prior knowledge of the Hessian's properties.

Choosing Hyper-Parameters

The minimization algorithms treated hereunder include Gradient Descent (GD) and two accelerated versions of it, Gradient Descent + Momentum (GD+M) and Anadromic Gradient Descent (AGD), all to be described in detail later. These iterations require choices for parameters called "Tuning Constants" or "Hyper-Parameters" that influence the rates of convergence of the iterations. One parameter, a "Step-Size", called a "Learning Rate" in DL, affects all three iterations the same way:

¬ If the step-size is too small, iterations converge much too slowly.
⌐ If the step-size is too big, iterations diverge or never converge.
≈ If the step-size is almost too big, iterations converge too slowly because they "Ricochet", bouncing around erratically.
Unfortunately, the best step-sizes can be almost "Almost-Too-Big".

Most attempts to choose good values for Hyper-Parameters are hindered by ignorance of things we cannot be expected to know in advance. Among these are the locations of "Stationary Points" x where the Gradient G(x) = o . These include the Minima we seek as well as local Maxima and Saddle-Points (neither maxima nor minima). How an iterate x is situated relative to stationary points determines two regimes differing in their strategies for choosing good hyper-parameters:

Regime #0: x is rather farther from a sought minimum than from other stationary points. Hessian H(x) varies enough to thwart attempts to infer good values for hyper-parameters other than step-sizes, as will be explained in detail below.

Regime #1: x is much closer to a minimum than to all other stationary points. Now the largest and least eigenvalues of H(x) , respectively ‖H(x)‖ and 1/‖H(x)^(-1)‖ , vary slowly, we hope. They are all that need be estimated to choose the best constant values for hyper-parameters. Rough estimates provided below work well enough, or are not needed at all!

The strategy for Regime #0 entails watchful responses to the behavior of by-products of each Iteration-Step. (DL calls one an "Epoch" .) Each iteration-step computes ƒ(x) and G(x) only once. They cost far more to compute than the watched by-products unless these reveal that a step-size was too big; then the iteration-step must be recomputed with a smaller step-size determined from the by-products. To compute step- sizes, formulas will be offered intended to render recomputation rare and convergence fast, but they might not.

A different strategy suits Regime #1. First comes a decision that the iteration has entered this regime. As iterates x step deeper into Regime #1, the iteration's rate of convergence comes to depend ever more exclusively upon how the choices of hyper-parameters relate to the two relevant unknown attributes of the Hessian H(x) at the minimum:
its norm ‖H‖ and its Condition Number Ç := ‖H‖·‖H^(-1)‖ .
Ç says something about the shape of the level-lines/surfaces/manifolds surrounding the minimum. They almost always resemble ellipsoids, and
Ç ≈ ((Largest Diameter)/(Least Diameter))² , so Ç ≥ 1 .
When Ç far exceeds that, the ellipsoids seem very like pancakes or long thin cigars; then H is called "Ill Conditioned", and the iteration is hyper-sensitive to roundoff and converges slowly at best no matter how the hyper-parameters were chosen. Ill Condition can be ameliorated sometimes by a "Pre-Conditioning" change of the coordinates in x-space; this topic deserves elaboration but not here.

A Gauge Function to Watch
~~~~~~~~~~~~~~~~~~~~~~~~?
Can an iteration's progress be gauged by how much it diminishes ƒ(x) ? By GD, almost. By accelerated versions, not likely, though another function Æ(x, v) will be found to be diminished by each iteration if its step-size is small enough. To serve as a guide to the choice of a good step-size, the gauge function's rate of decrease for very small step-sizes must be computable cheaply; and then an iteration-step's departure from that rate may guide the choice of a better step-size, though it is usually not small at all.

Let's see how well that works for un-accelerated GD :

Gradient Descent
~~~~~~~~~~~~~~~~?
GD is a "1st Order" discretization of the differential equation dx(τ)/dτ = -G(x(τ)) in which gradient G(x) := dƒ(x)/dx` . Along every trajectory x(τ) satisfying this differential equation,
dƒ(x(τ))/dτ = -G(x(τ))`•G(x(τ)) ≤ 0 ,
so ƒ(x(τ)) decreases until the trajectory terminates at a Stationary Point x^a where $G(x^a) = o$. Unless $x^a = \infty$, it is most likely a (local) minimum of ƒ since all other kinds of stationary points repel almost all trajectories.

Each GD iteration-step replaces x(τ) by Euler's approximation
~~~~~~~~ new x := x - G(x)·Δτ ≈ x(τ + Δτ) ± O(Δτ²) for some sufficiently small step-size (called "Learning Rate" by DL) Δτ > 0 . It changes the gauge function ƒ(x) chosen for GD to
ƒ(new x) ≈ ƒ(x) - Δτ·(‖G(x) + G(new x)‖² + 4‖G(x)‖²)/8 + O(Δτ)^3 .
~~~~~~~~

Here $\|G\|^2 := G`\bullet G$; and the multiplier of $\Delta\tau$ is the rate at which $f(\text{new } x)$ decreases for very small step-sizes. *It's not obvious*. That rate is easy to compute in the course of preparing for the next step.

However, whenever a computed $f(\text{new } x) \geq f(x)$, this new x must be discarded and recomputed from the saved x and G(x) , but now with a new step-size $\delta\tau$ smaller than $\Delta\tau$.

How much smaller?

A rough amswer comes from the term " $O(\Delta\tau)^3$ " in the equation above. It is represented by " $¥(x, ?\tau)\cdot\|G(x)\|^2\cdot\Delta\tau^3$ " for a presumed slowly varying function $¥(x, \Delta\tau)$ determined from the equation above, namely
$f(\text{new } x) = f(x) - \Delta\tau\cdot(\|G(x)+G(\text{new } x)\|^2+4\|G(x)\|^2)/8 + ¥(x,\Delta\tau)\cdot\|G(x)\|^2\cdot\Delta\tau^3$.
A big leap occurs when we replace $\Delta\tau$ by some smaller $\delta\tau$ to predict
$f(\text{new } x) \approx f(x) - \delta\tau\cdot(\|G(x)+G(\text{new } x)\|^2+4\|G(x)\|^2)/8 + ¥(x,\Delta\tau)\cdot\|G(x)\|^2\cdot\delta\tau^3$
and ignore the presumed-to-be-small changes in the saved values of ¥ and $\|G(x)+G(\text{new } x)\|^2$. A usually slight under-estimate of the best smaller $\delta\tau$ minimizes the last equation's right-hand side:

$$\delta\tau := \frac{\Delta\tau}{\sqrt{\left(\max\left\{\ 0.8,\ \ 3 + \frac{24\cdot(\ f(\text{new } x) - f(x)\ )}{\Delta\tau\cdot(\ \|G(x) + G(\text{new } x)\|^2 + 4\|G(x)\|^2\ )}\right\}\right)}} \ .$$

Then discard saved values of new x , $f(\text{new } x)$ and G(new x) etc., and recompute a new new x using this smaller $\delta\tau$ in place of $\Delta\tau$.

When $f(\text{new } x) < f(x)$, keep new x and adjust the next step's step- size to $\delta\tau$ from the foregoing formula. Now its " max{ 0.8, ...} " acts to impose a bound $1.12 > \delta\tau/\Delta\tau$ restricting step-sizes' increases lest too many iteration-steps be wasted recomputing new x at step-sizes reduced because $f(\text{new } x)$ increased. Recomputing becomes rare as iterates x approach any stationary point because then $\|G(x)\|$ and $\|G(\text{new } x)\|$ dwindle, whereupon $¥(x,\Delta\tau)\cdot\|G\|^2$ tends to fluctuate among positive values: $¥(x, \Delta\tau)\cdot\|G(x)\|^2 \approx \|H(x)\bullet G(x)\|^2/8 \pm O(\|G(x)\|^3)$.

Thus adjusted, step-sizes $\delta\tau$ need not converge in Regime #1 though H(x) does. Instead $\delta\tau$ tends there to fluctuate around values that have been observed to make iterates x converge at least as fast as if step-sizes had all been assigned the same optimal _constant_ value.
When do iterates x get into Regime #1 ? A symptom is the rarity of recomputations of new x while $\|G\|$ shrinks steadily, though that happens also when iterates approach a saddle-point. Both this approach and its departure can cost many iteration-steps. Sometimes (rarely) the iteration's approach to a saddle-point is foreshadowed by an occurrence of this inequality:

$$0 \geq G(x)`\bullet(\ G(x) - G(\text{new } x)\ ) \approx \Delta\tau\cdot G(x)`\bullet(H(x) + H(\text{new } x))\bullet G(x)/2 \pm O(\Delta\tau\cdot\|G(x)\|)^3 \ ,$$

because $G`\bullet H\bullet G < 0$ whenever G enters a cone surrounding all eigen-vectors belonging to any negative eigenvalues of H . To stimulate the detection of and escape from saddle-points, occasionally add to new x a very small random perturbation orthogonal to both G(new x) and G(x) while $\|G(\text{new } x)\|$ is small but still far from tiny enough to ignore.

As iterates x go deeper into Regime #1, step-size management gets simpler because ‖G(x)‖ dwindles, Hessian H(x) becomes more nearly constant though still unknown, and f(new x) becomes ever better approximated thus:

$$f(\text{new } x) \approx f(x) - \Delta\tau\cdot\|G(x)\|^2 + \Delta\tau^2\cdot G(x)`\bullet H(x)\bullet G(x)/2 \pm O(\Delta\tau\cdot\|G(x)\|)^3 .$$

This approximation suggests that the next iteration-step use

$$\text{new } \Delta\tau := \Delta\tau/\max\{\ 0.25,\ \ 2 + 2(f(\text{new } x) - f(x))/(\|G(x)\|^2\cdot\Delta\tau)\ \}$$
$$\approx \min\{\ 4\Delta\tau,\ \ G(x)`\bullet G(x)/(\ G(x)`\bullet H(x)\bullet G(x) \pm \Delta\tau\cdot O(\|G(x)\|^3\ )\ \}$$

as its step-size regardless of whether f(new x) ≥ f(x) , as may occur occasionally. Step-sizes new Δτ computed this way can fluctuate chaotically between 1/‖H‖ and Ç/‖H‖ . However, this GD iteration converges, almost always converging ultimately almost as fast as can accelerated versions like GD+M when their hyper-parameters like step-sizes are chosen optimally. But the weasel-words "almost" and "ultimately" here deserve some explanation, albeit not yet a full explanation.

Why the last formula for new Δτ works so well has not been explained yet partly because, very rarely, that formula does not work so well.

Here is a Simple Example: Minimize the scalar $f(x) := x`\bullet H\bullet x/2$ over integrable functions x(Θ) defined only on an interval $1 \le \Theta \le Ç > 1$ whereon H is the operator that multiplies by Θ ; $H\bullet x(\Theta) = \Theta\cdot x(\Theta)$. The scalar product is an integral over the aforementioned interval:

$$\|x\|^2 = x`\bullet x := \int x(\Theta)^2\cdot d\Theta \quad \text{and} \quad x`\bullet H\bullet x = \int x(\Theta)^2\cdot\Theta\cdot d\Theta\ .$$

Of course the minimum is 0 , achieved at $x \equiv 0$, but we will seek it by Gradient Descent using the gradient $G(x) := df(x)/dx` = H\bullet x$; i.e., $G(x(\Theta)) = \Theta\cdot x(\Theta)$. Each iteration-step of GD replaces x by

$$\text{new } x := x - \Delta\tau\cdot G(x)\ ,$$

so new $x(\Theta) = (1 - \Delta\tau\cdot\Theta)\cdot x(\Theta)$ for some chosen $\Delta\tau > 0$. The next iteration's new $\Delta\tau := (G(x)`\bullet G(x))/(\ G(x)`\bullet H\bullet G(x)\ )$ accords with the foregoing formula for it; i.e.,

$$\text{new } \Delta\tau := \left(\int x(\Theta)^2\cdot\Theta^2\cdot d\Theta\ \right)/\left(\int x(\Theta)^2\cdot\Theta\text{^}3\cdot d\Theta\ \right)\ .$$

Step-sizes new Δτ usually vary irregularly in the interval $1/Ç < \Delta\tau < 1$; but a peculiar choice of starting values for Δτ and x(Θ) can produce stepsizes that all stay the same, thus producing a sequence of iterates x convergent much more slowly than usual. Such a choice is $\Delta\tau := 2/(1 + Ç)$ and, for any $\Phi(|ø|) > 0$,

$$x(\Theta)^2 := \Phi(\ |Ç + 1 - 2\Theta|/(Ç - 1)\ )/\Theta^2\ .$$

This amounts to changing from variable $\Theta := (Ç + 1 + (Ç - 1)\cdot ø)/2$ to ø on $-1 \le ø \le 1$. Then a long calculation yields new Δτ = Δτ and, after a large number n of iterations, ‖G(x)‖ shrinks by a factor not much smaller than $(1 - 2/(1 + Ç))^n$ instead of a usually observed factor moderately bigger than $(1 - 2/(1 + \sqrt{Ç}))^n$. The difference is substantial when $Ç >> 1$. End of Simple Example.

Since ‖H‖ and Ç are hardly ever known, a nearly constant sequence of step-sizes new Δτ like the Example's can rarely be distinguished from the same behavior when Ç exceeds 1 only a little. To cope with uncertainty, compute new $\Delta\tau := \Delta\tau/2$ occasionally. This thwarts misbehavior like the Example's without slowing convergence much when the

formula above for new $\Delta\tau$ is used almost always. The rapidity of convergence attributed to that formula has not been proved yet but has been observed for many examples some with dimensions over 1000.

Ultimately, How Fast at Best Can Gradient-Based Iterations Converge?
~~~~~~
Only deep in Regime #1 can the question be Answered; this is what "Ultimately" means. Our Answer depends upon a rarely known attribute of the Hessian $H(x) := d^2f(x)/dx^2$ at a minimum x of $f(x)$, namely its Condition Number $Ç := \|H\| \cdot \|H^{-1}\|$; if huge, "Fast" becomes "Slowly".

"Gradient-Based Iterations" are those that compute only $f(x)$ and its gradient $G(x) := df(x)/dx$` at arbitrary points x, storing at most a few instances of them; iterations GD, GD+M and AGM are included. "Conjugate Gradients" iterations are also included though they compute $H \bullet v$ too for arbitrary vectors v and compute optimal hyper-parameters automatically. Otherwise Hessian H is inaccessible.

"Fast at Best" means that from no initial points x can the average factor by which each iteration shrinks $\|G(x)\|$ exceed the question's Answer, and from most points x the average comes close. Where an iteration requires hyper-parameters to be chosen, our Answer assumes their *best possible values* have been chosen. Most iterative methods' optimal choices require knowledge of usually unknown attributes $Ç$ and $\|H\|$, thus diluting our Answer's pertinence. Here is our Answer:

> After a large number n of Gradient-Based Iteration-steps in Regime #1, the factor by which they have reduced $\|G(x)\|$ can not exceed $(1 - 2/(1 + \sqrt{Ç}))^n$, had all hyper-parameters been chosen optimally. Only if $n/(1 + Ç) >> 1$ is n big enough.
>
> This Answer has been Proved.
> ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

As iterates x approach a minimum, our iterations come ever closer to resembling some iterations explored in "Iterative Methods for the Solution of a Linear Operator Equation in Hilbert Space - A Survey" by Major Walter M. Patterson 3rd, (1974) Lecture Notes in Mathematics #394 Springer-Verlag, Berlin. Our Answer's reduction factor comes from a 1960s theorem about Conjugate Gradients by V. Samanskii cited by Patterson on pp.162-3 and elaborated subsequently by many others. See "On Meinardus' Examples for the Conjugate Gradient Method" by Prof. Ren-Cang Li (2007) pp. 335-352 in Mathematics of Computation 77 #261, and citations therein.

Our Answer's applicability to GD+M and AGD iterations comes from a tedious computation of optimal constant values of hyper-parameters that minimize the magnitudes of extreme eigenvalues of a matrix derived from H. These optimal constants depend upon usually unknown values of $\|H\|$ and $Ç$ and are delicate if $Ç$ is huge because then constants slightly different from optimal can cause iterations to diverge in Regime #1.

Our Answer's application to GD demands $\|H\|$ and $Ç$ and a sequence of step-sizes $\Delta\tau$ derived from the zeros of a Tschebyshev polynomial whose degree exceeds the expected number of iterations needed to shrink $\|G(x)\|$ below some preassigned tolerance. Roundoff degrades the scheme badly if
~~~~~~

Ç >> 1 unless step-sizes are chosen in a proper order. For an easier way see §10.1.5 of "Matrix Computations" 3rd. ed. (1996) by Profs. G. Golub and C. Van Loan and citations therein.

On average, each GD iteration with its *best constant* step-size $\Delta\tau := (2/\|H\|)/(1 + 1/Ç)$ shrinks $\|G(x)\|$ by a factor $1 - 2/(1 + Ç)$, which far exceeds the Answer's factor $1 - 2/(1 + \sqrt{Ç})$ when Ç is huge. Then $\Delta\tau$ is almost too big; GD diverges for $\Delta\tau \geq 2/\|H\|$.

A Perplexing Miracle: Yet to be explained is GD's rapid convergence observed in Regime #1, with no further information about H , when new $\Delta\tau$ is computed from the formula above except new $\Delta\tau := \Delta\tau/2$ on random occasions. Then convergence seems almost as fast as our Answer's fastest convergence achievable by any Gradient-Based Iteration. It all seems miraculous.

Our derivations of formulas for adjusted step-sizes $\Delta\tau$ were based on a fictional hypothesis, namely that all but the leading few terms of an expansion in powers of $\Delta\tau$ could be ignored. In fact, step-sizes $\Delta\tau$ are not that small. However, step-sizes too often chosen slightly too big can cause GD iterations to ricochet forever, bouncing around in an irregular way impossible to distinguish from slow convergence at the foregoing Answer's best possible rate for an unknown Ç >> 1 .

Other Valuable By-Products: Two by-products of GD iterations deep in Regime #1 are estimates of $\|H\|$ and Ç from which optimal hyper-parameters could be estimated for GD+M and AGM, though the motivation for using them is undermined if GD already converges almost as fast as they could at best. We'll see.

Rough estimates of $\|H\|$ come via $H(x) = dG(x)/dx$ and its implication $\|H(x)\| \approx \max\{ \|G(x+\delta x) - G(x)\|/\|\delta x\|\}$ over all δx small enough. Therefore each quotient $\|G(x) - G(\text{new } x)\|/\|x - \text{new } x\|$ under-estimates $\|H\|$ in the neighborhood of x and new x . The largest such quotient among the past several iterations offers a rough estimate of $\|H(x)\|$ for x in the neighborhood of the past several iterates in Regime #1 provided they are not yet so nearly coincident that rounding errors in dwindling gradient vectors G drown differences between them.

Very rough estimates of Ç come from the observation that our formula for computing new $\Delta\tau$ appears to make GD converge, in Regime #1, at almost the Answer's best possible rate for that condition number. Therefore, after some moderately large number n of iterations during almost all of which $\|G\|$ has shrunk, the accumulated shrinkage factor over n iterations tends to over-estimate $(1 - 2/(1+\sqrt{Ç}))^n$ a little, thus providing an over-estimate of Ç . Adequate accuracy will require $n/(1+Ç) >> 1$, which may take longer than anyone is willing to wait.

A use for estimates of Ç and $\|H\|$ more important than to estimate hyper-parameters is to describe the shape of a local minimum x^a and distinguish it from others, if any, that have been or will be found. The shape is inferred from the first few terms of a Taylor series

$$f(x^a + \delta x) \approx f(x^a) + G(x^a)\bullet\delta x + \delta x`\bullet H(x^a)\bullet\delta x/2 \pm O(\delta x\text{^}3) .$$

Gradient $G(x^a) \approx o$ at a minimum, so only the properties of $f(x^a)$ and its Hessian $H(x^a)$ determine the shape of the minimum. $f(x^a)$ is its Depth. The minimum is Sharp if $\|H(x^a)\|$ is big, Broad if small. The Condition Number Ç is the minimum's Spread, concentrated near a point if $Ç \approx 1$ or diffused along a long thin cigar or flat pancake if $Ç >> 1$. These properties affect that minimum's usefulness:

A sharp minimum restricts x^a tightly in the sense that small changes can render x^a utterly non-minimizing. A broad minimum tolerates a modest range of changes in x^a and/or f . A small spread, when Ç is not very big, implies that the direction of a small change in x^a and/or the data that defines f matters far less than its magnitude measured by an appropriate norm $\|...\|$. When Ç is huge, f takes nearly its minimum value along a line segment, like a narrow valley, or on a platter, or throughout some ramified near-ellipsoid within which accidental little imperfections in data or arithmetic determine the location of a computed x^a ; and it takes a long time to compute.

Different minima can have different shapes. Unless f is known to have some property like convexity that implies at most one minimum, prudence demands that others be sought after one is found.

How Significant is a Discretization's "Order" ?

We cannot explain it if we use the same name " x " for a given ODE's trajectories $x(\tau)$ and our numerical approximations; let's call these X . These consist of a sequence of points which we shall join to form a continuous piecewise differentiable curve. If $X(t)$ is one of these points and $X(t + \Delta\tau)$ the next, we shall connect them with the graph of $X(t + ø\cdot\Delta\tau)$ for $0 \le ø \le 1$ defined as if we had used step-size $ø\cdot\Delta\tau$ instead of $\Delta\tau$. Our graph of $X(\tau)$ has kinks at points $X(t)$.

Our kinky graph intersects some of our ODE's trajectories. For any t choose $x(\tau)$ to be the trajectory that intersects at $X(t)$; thus $x(t) = X(t)$. Here a 1st Order discretization's graph has the same forward derivative: $dx(\tau)/d\tau = dX(\tau)/d\tau+$ at $\tau = t$, but generally no higher derivative agrees. In other words, trajectories and graphs share forward tangents at intersections, but usually not curvatures.

2nd Order discretizations share forward curvatures at intersections as well as tangents: $d^2x(\tau)/d\tau^2 = d^2X(\tau)/d\tau^2+$ too at $\tau = t$.

If, given values for x° and $T > 0$, we wished to approximate $x(\tau)$ for $x(0) := x^\circ$ and $0 \le \tau \le T$, we would set $X(0) := x^\circ$ and run as many steps of our chosen discretization as needed, perhaps varying the step-sizes $\Delta\tau$, to reach $X(T)$. For tiny steps, its error would be ...

$x(T) - X(T) \approx O(\max \Delta\tau)$ for a 1st Order discretizarion,

$x(T) - X(T) \approx O(\max \Delta\tau^2)$ for a 2nd Order discretizarion.

A tempting inference is that 2nd Order would permit longer steps $\Delta\tau$ than 1st and fewer of them to reach $x(T)$ with a tolerable error.

Alas, that inference could be mistaken for any of three reasons:

1■ 2nd Order may be better only for $\Delta\tau$ far tinier than 1st needs.
2■ 2nd Order may cost much more computing time per step than 1st.
3■ All trajectories we care about end at the same places, minima, so small departures along the way don't accumulate.

Despite those three reasons, a 2nd Order discretization will turn out sometimes to be advantageous because ...

1■ We eschew very tiny step-sizes $\Delta\tau$ since they retard convergence.
2■ 2nd Order AGD costs scarcely more than 1st Order GD and GD+M.
3■ Although all trajectories we care about end at the same places, getting there can take abrupt turns and perhaps tight corkscrews through a high-dimensional space difficult to visualize.

Let's try to visualize trajectories $x(\tau)$ of GD's ODE $dx/d\tau = -G(x)$. As they approach a minimum $x(\infty) = x^a$ where $G(x^a) = o$, almost all turn abruptly, though slowly since $\|G(x(\tau))\|$ becomes small there. To assess how abruptly, suppose $\delta x(\tau) := x(\tau) - x^a$ is so tiny that

$$d\delta x(\tau)/d\tau = -(G(x(\tau)) - G(x^a)) \approx -H(x^a)\bullet\delta x(\tau)$$

is an approximation adequate to justify another approximation

$$\delta x(\tau) \approx \exp((t-\tau)\cdot H)\bullet\delta x(t) \quad \text{for all } \tau > t .$$

Now consider the components of $\delta x(\tau)$ parallel to eigenvectors of H . If the component belonging to the least eigenvalue (it's ≥ 0) of H is nonzero, it decays exponentially slower than any other component. The slowest components belong to the smoothest trajectories that run into x^a ; almost all nearby trajectories differ from the smoothest by amounts proportional to $\|\delta x(\tau)\|\hat{}Ç$ wherein Condition Number Ç is the ratio of the largest over the least eigenvalues of H .

The bigger is Ç , the more nearly do almost all trajectories come to resemble a twisted letter " L ". Moreover, a microscope focussed on x^a would exhibit the same picture of *all* the nearby trajectories regardless of magnification. In other words, ...

> In any small neighborhood of a minimum x^a of $f(x)$,
> almost all trajectories bend abruptly, the more so if
> Ç is huge, no matter how small is the neighborhood.

Each step new $X := X - G(X)\cdot\Delta\tau$ of GD moves from X to new X along the tangent to the trajectory $x(\tau)$ through $X = x(t)$ for some t . Almost all points X near x^a lie on or near the heel of the twisted L-shaped trajectory $x(\tau)$ through X , so the tangent departs quickly from the trajectory. If $\Delta\tau$ is small enough, new X will lie close to the trajectory which we wish to follow towards a minimum of f in Regime #0. In Regime #1 step-sizes so small retard convergence; big steps required for faster convergence can follow no trajectory closely, but richocet, bouncing about seemingly randomly. And slightly bigger steps diverge. The difference between the fastest possible convergence and divergence takes many iteration-steps to discern if Ç is huge.

> In short, if a discretization's order matters at all,
> it matters only while the iteration is in Regime #0.

In Regime #0 trajectories do turn abruptly to escape from a nearby saddle-point. Otherwise a 2nd Order discretization is advantageous only in so far as it hastens an iteration's entry into Regime #1 near a chosen minimum different from others found earlier from sufficiently different starting iterates.

Gradient Descent + Momentum

Inspired perhaps by Conjugate Gradients Iteration, one way to present GD+M as an iteration to minimize $f(x)$ is this:

GD+M: new x := x - $\alpha \cdot G(x)$ + $ß \cdot$(x - old x) .

Here Gradient $G(x) := df(x)/dx`$; Momentum is proportional to x - old x ; and "Hyper-Parameters" $\alpha > 0 < ß < 1$ are two scalars that have to be chosen somehow. (The DL community calls α "The Learning Rate".) If α and $ß$ are chosen well, repeated iterations do diminish $f(x)$, but not necessarily with *each* iteration-step. Especially when the level-lines/surfaces/manifolds enclosing a minimum of f resemble very elongated or flattened ellipsoids, an iteration-step may increase f yet bring new x closer than x to a minimum.

Inspired perhaps by Hamiltonian differential equations that describe friction-free mechanical systems, GD+M has been identified with a discretization of such a differential equation to which friction has been added. For some "Drag" $\mu > 0$, the differential equation is

$$d^2x/d\tau^2 = -G(x) - \mu \cdot dx/d\tau \ ;$$

i.e., $dv/d\tau = -G(x) - \mu \cdot v$ & $dx/d\tau = v$.

Along every trajectory $[x(\tau), v(\tau)]$ satisfying these equations, a Pseudo-Hamiltonian function $Æ(x, v) := f(x) + v`\bullet v/2$, analogous to total energy (Potential + Kinetic), decreases because of the Drag:

$$dÆ(x(\tau), v(\tau))/d\tau = -\mu \cdot v(\tau)`\bullet v(\tau) < 0 \text{ so long as } v(\tau) \neq o \ .$$

Each trajectory terminates at a point $[x^a, v^a]$ where $v^a = G(x^a) = o$. Unless $x^a = \infty$, this Stationary Point is most likely a (local) minimum of f and of $Æ$, since every other kind of stationary point repels almost all trajectories.

Besides turning abruptly, trajectories $[x(\tau), v(\tau)]$ turn corkscrews unless drag $\mu \geq 2\sqrt{\|H(x)\|}$, which is bigger than μ is likely to be chosen. Larger values of the Hessian's Condition Number Ç produce tighter corkscrews more like tightly wound clock-springs, and require smaller step-sizes $\Delta\tau$ lest a discretization go utterly astray.

A simple "1st-Order" discretization of the differential equation is

GD+M: new v := v - $\Delta\tau \cdot$(G(x) + $\mu \cdot$v) & new x := x + $\Delta\tau \cdot$new v .

i.e., new x := x - G(x)$\cdot\Delta\tau^2$ + (1 - $\mu \cdot \Delta\tau$)$\cdot$(x - old x) .

This last identifies the iteration's

$$\alpha \equiv \Delta\tau^2 \ , \quad ß \equiv 1 - \mu \cdot \Delta\tau \ , \quad \text{and } x - \text{old } x \equiv ?\tau \cdot v \ .$$

These identifications are slightly problematical:

Which Hyper-Parameters, $\{\alpha, ß\}$ or $\{\Delta\tau, \mu\}$, are best used?

The question seems answered by the choice of the Pseudo-Hamiltonian

$$Æ(x, v) := f(x) + v`\cdot v/2$$

to gauge the iteration's progress. This choice $Æ$ is independent of $\{\Delta\tau, \mu\}$ and hence independent of how they are determined. A different gauge function, one independent of $\{\alpha, ß\}$, has not been found yet.

For $\Delta\tau$ small enough, each GD+M iteration-step changes $Æ(x, v)$ to

$$Æ(\text{new x}, \text{new v}) \approx Æ(x, v) - \Delta\tau \cdot \mu \cdot \|v\|^2 \pm O(\Delta\tau^2)$$
$$\approx Æ(x, v) - (1-ß) \cdot \|v\|^2 \pm O(\alpha) \ .$$

When $Æ(\text{new x}, \text{new v}) \geq Æ(x, v)$ the last equation does not suggest how best to recompute new x with a smaller α without changing $ß$ too. However the previous equation suggests replacing $\Delta\tau$ by a much smaller

$$\delta\tau := \frac{\Delta\tau}{2 + 2(Æ(\text{new } x, \text{new } v) - Æ(x, v))/(\mu\cdot\|v\|^2\cdot\Delta\tau)}$$

and then recomputing new v and new x.

When $Æ(\text{new } x, \text{new } v) < Æ(x, v)$, a slightly better formula is

$$\delta\tau := \frac{\Delta\tau}{\max\{ 0.91,\ 2 + 2(Æ(\text{new } x, \text{new } v) - Æ(x, v))/(\mu\cdot\|\text{new } v\|^2\cdot\Delta\tau) \}}$$

to adjust the step-size for the next step. Restricting $\delta\tau/\Delta\tau < 1.1$ purports to render rare any subsequent iteration-steps whose increased Æ would compell recomputations of new v and new x . Unfortunately neither formula for $\delta\tau$ suggests how to adjust the drag μ , partly explaining why neither formula has been found to work well. A further explanation is that the term " $\pm O(\Delta\tau^2)$ " is too often negative.

Another version of GD+M was introduced in the early 1980s by Y. Numerov and is now widely used; it will be called GD+MN here:

Choose initial vectors x and u , maybe u := o , and then ...

GD+MN: Iterate y := x + (1 - μ·Δτ)·Δτ·u ;
new u := u - (G(y) + μ·u)·Δτ ;
new x := x + (new u)·Δτ .

Since ƒ and G will be computed only once per iteration-step, the iteration can be reduced to its essentials thus:

Choose an initial x and u and set y := x + (1 - μ·?τ)·?τ·u ; then

GD+MN: Iterate new u := u - G(y)·Δτ + μ·Δτ·u ;
new y := y + (2-μ·Δτ)·Δτ·(new u) - (1-μ·Δτ)·Δτ·u .

Only at the end, new x := new y - (1 - μ·Δτ)·Δτ·(new u) .

GD+MN is another 1st Order discretization of the same 2nd Order Hamiltonian-like differential equations as GD+M discretized. If Δτ is small enough, each iteration-step of GD+MN reduces the Pdeudo-Hamiltonian function Æ(y, u) , used to gauge progress, to

$$Æ(\text{new } y, \text{new } u) \approx Æ(y, u) - \Delta\tau\cdot\mu\cdot\|\text{new } u\|^2 \pm O(\Delta\tau^2) .$$

This approximation suggests a formula for adjusting Δτ to δτ like the one for GD+M above, but it suffers from the same limitations partly because the term "$\pm O(\Delta\tau^2)$" is too often negative.

Surprisingly, slightly changing μ sometimes causes GD+M and GD+MN to diverge unless also Δτ is reduced. This surprise can be explained when either iteration is deep enough in Regime #1 that Hessian H(x) is almost constant. Then conditions necessary and sufficient for each iteration's convergence with constant hyper-parameters can be decribed:

For GD+M, $0 < \|H\|\cdot\alpha < 2 + 2ß < 4$ or $\|H\|\cdot\Delta\tau^2 < 4 - 2\mu\cdot\Delta\tau < 4$.

For GD+MN the conditions are more complicated but simplified by what seems to be Numerov's stipulation that $0 < \mu\cdot\Delta\tau < 1$; this leads to $0 < \Delta\tau < \sqrt{(2/\|H\|)}$, and if $\Delta\tau > 2/\sqrt{(3\|H\|)}$ then also $\mu > (3\|H\|\cdot\Delta\tau/2 - 2/\Delta\tau)/(\|H\|\cdot\Delta\tau^2 - 1)$.

When Ç >> 1 those conditions are almost violated by optimal choices for *constant* hyper-parameters. Below are the optimal choices and the average factors Ω by which each of very many iteration-steps in Regime #1 can be expected to reduce $\|G(x)\|$. Also exhibited are the values of hyper-parameters Y. Numerov chose for his GD+MN.

	Y.N's GD+MN		Best GD+MN		Best GD+M	
$\Delta\tau\cdot\sqrt{\|H\|}$:	1	<	$2/\sqrt{(3+1/Ç)}$	<	$2/(1+1/\sqrt{Ç})$	
$\mu\cdot\Delta\tau$:	$2/(1+\sqrt{Ç})$	<	$4/(2+\sqrt{(3Ç+1)})$	<	$4\sqrt{Ç}/(1+\sqrt{Ç})^2$	< 1
Ω:	$1-1/\sqrt{Ç}$	>	$1-2/\sqrt{(3Ç+1)}$	>	$1-2/(1+\sqrt{Ç})$	

Tabulated below are numerical values of Ω for a few choices $Ç$:

$Ç$	Y.N's Ω(GD+MN)	Best Ω(GD+MN)	Best Ω(GD+M)
2	0.2928932188	0.2440710539	0.1715728752
16	0.75	0.7142857142	0.6
225	0.9333333333	0.9230769230	0.875
3136	0.9821428571	0.9793814432	0.9649122807

At least in Regime #1, GD+M can converege much faster than GD+MN can, and as fast as the fastest Gradient-Based iteration (see Answer above).

When do iterates enter Regime #1? A symptom is dwindling lengths $\|G\|$ and $\|u\|$ or $\|v\|$; but they may also portend approaches to a Saddle- Point, which is sometimes foreshadowed by one of these inequalities:

GD+M: $0 > (\text{new } v)`\bullet(G(\text{new } x) - G(x))$
$\approx \Delta\tau\cdot(\text{new } v)`\bullet(H(\text{new } x)+H(x))\bullet(\text{new } v)/2 \pm O(\Delta\tau^5)$

GD+MN: $0 > (\text{new } u - u/2)`\bullet(G(\text{new } y) - G(y))$
$\approx \Delta\tau\cdot(\text{new } u - u/2)`\bullet(H(\text{new } y)+H(y))\bullet(\text{new } u - u/2) \pm O(\Delta\tau^3)$

Occasionally adding to an iterate a small random vector orthogonal to both new G and new u or new v often speeds the detection of and departure from a nearby saddle-point.

Deep in Regime #1, convergence goes fastest if $\Delta\tau$ and $\mu\cdot\Delta\tau$ have their optimal values exhibited above; but these depend upon estimates of the Hessian's norm $\|H\|$ and condition number $Ç$. An estimate for $\|H\|$ comes from the largest quotient $\|G(x) - G(\text{new } x)\|/\|x - \text{new } x\|$ observed while deep in Regime #1. Especially if $Ç$ is big it almost defies adequately accurate estimation because GD+M's and GD+MN's sequence of values $\|G\|$ tends to behave raggedly; and the sequence of values $Æ$, though decreasing if every $\Delta\tau$ is small enough, does not often decrease smoothly to zero. We might as well seek a way to manage without any estimate of $Ç$.

Whatever way is chosen to adjust $\Delta\tau$, perhaps a formula like GD+M's for $\delta\tau$ above, any estimate of $\|H\|$ can serve to correlate μ with $\Delta\tau$ just as optimal values would be correlated, if they were known, *provided* $\Delta\tau$ is within range:

For GD+M: $\mu\cdot\Delta\tau = (2 - \sqrt{\|H\|\cdot\Delta\tau})\cdot\sqrt{\|H\|\cdot\Delta\tau} \le 1$ if $\Delta\tau\cdot\sqrt{\|H\|} < 2$

For GD+MN: $\mu\cdot\Delta\tau = 2/(1 + 1/\sqrt{(4-3\|H\|\cdot\Delta\tau^2)}) < 1$ if $1 < \Delta\tau\cdot\sqrt{\|H\|} < 2/\sqrt{3}$

But why bother with all that when ordinary GD works so well as long as its formula for new $\Delta\tau$ is used, all while knowing nothing about the Hessian? Almost nothing. The iteration has to be known to be in Regime #1 before GD's new $\Delta\tau$ can be expected to work well. What is the fasteest way to escape from Regime #0 into Regime #1 ? Perhaps GD+M and GD+MN have been created in the hope that they might escape sooner than GD. It seems to be a faint hope since they are all 1st Order diacretizations of differential equations with trajectories that corkscrew and bend abruptly.

A 2nd Order discretization might go faster; it seems worth trying. That is the motivation for AGD.

Anadromic Gradient Descent

"Anadromic" is a Nonce-word meaning "Able to return", derived from the same Greek roots as "Anadromous", which characterizes fish like salmon that return from the ocean to spawn in the same fresh-water stream where they hatched. An anadromic discretization of an ordinary differential equation shares one of its "semi-group" properties:

If the sequence of steps $\Delta\tau$ or $d\tau$ is revesed in sign and order, $x(\tau)$ returns to its initial value (except for rounding errors).

Anadromic discretizations are called "Symmetric" in Ch. V of the text "Geometric Numerical Integration -- Structure-Preserving Algorithms for Ordinary Differential Equations" 2d. ed. by E. Hairer, C. Lubich and G. Wanner (2006) Springer-Verlag, Berlin. The word "Symmetric" is overworked; for instance, differential equations called "Reversible" in the text have been called "Symmetric" elsewhere. And Hessians are symmetric arrays.

Most discretizations, GD, GD+M and GD+MN among them, are not anadromic. All anadromic discretizations have even Order, so at least 2; but not all of them are computationally economical. AGD is economical; it costs roughly one scalar division more than GD+MN :

Choose initial vectors x and v , maybe v := o , and then ...

AGD: Iterate $y := x + v\cdot\Delta\tau/2$;
$\text{new } v := v - (G(y) + \mu\cdot v)\cdot\Delta\tau/(1 + \mu\cdot\Delta\tau/2)$;
$\text{new } x := y + (\text{new } v)\cdot\Delta\tau/2$.

This AGD discretizes the same differential equation as GD+MN did; i.e., $dv/d\tau = -G(x) - \mu\cdot v$ & $dx/d\tau = v$.
Recall that along its trajectories $Æ(x, v) := ƒ(x) + \|v\|^2/2$ decreases.

If $v \approx dx(\tau)/d\tau$ and $x \approx x(\tau)$ then $y \approx x(\tau+\Delta\tau/2)$, new $x \approx x(\tau+\Delta\tau)$ and new $v \approx dx(\tau+\Delta\tau)/d\tau$. To verify that AGD is anadromic, suppose that new x and new v are known; then compute in turn

$y := \text{new } x - (\text{new } v)\cdot\Delta\tau/2$;
$v := \text{new } v + (G(y) + \mu\cdot(\text{new } v))\cdot\Delta\tau/(1 - \mu\cdot\Delta\tau/2)$;
$x := y - v\cdot\Delta\tau/2$

to recover x and v from the formula for AGD except with the sign of $\Delta\tau$ reversed. Therefore our AGD is a 2nd Order discretization of the differential equation; this inference is partially corroborated by the appearance of only even powers of $\Delta\tau$ in the expansion of ...

$\{ Æ(\text{new } x, \text{new } v) - Æ(x, v) \}/\Delta\tau \approx -\mu\cdot\|(v + \text{new } v)/2\|^2 \pm O(\Delta\tau^2)$.

We never compute this expression numerically because ƒ and G are to be computed only once per iteration-step. As we did to GD+MN abone, we reduce AGD to its essentials thus:

Choose an initial x and v and set $y := x + v\cdot\Delta\tau/2$; then

AGD: Iterate $\text{new } v := v - (G(y) + \mu\cdot v)\cdot\Delta\tau/(1 + \mu\cdot\Delta\tau/2)$;
$\text{new } y := y + (\text{new } v)\cdot\Delta\tau$.

Only at the end, $\text{new } x := \text{new } y - (\text{new } v)\cdot\Delta\tau/2$.

Then only even powers of $\Delta\tau$ appear in the expansion of ...
$\{ Æ(y, (v + \text{new } v)/2) - Æ(\text{old } y, (v + \text{old } v)/2) \}/\Delta\tau \approx -\mu\cdot\|v\|^2 \pm O(\Delta\tau^2)$,
but only if the same step-size $\Delta\tau$ is used to advance $\{y, v\}$ to $\{\text{new } y, \text{new } v\}$ as was used to advance $\{\text{old } y, \text{old } v\}$ to $\{y, v\}$. This

constraint will have to be circumvented if step-sizes $\Delta\tau$ are to be varied automatically.

Were AGD to be used only deep in Regime #1 where the Hessian's Ç and $\|H\|$ are almost constant, constant hyper-parameters μ and $\Delta\tau$ would provide AGD with predictable behaviors. $\Delta\tau\cdot\sqrt{\|H\|} < 2$ would be necessary and sufficient for convergence. For fastest convergence the best choices would be $\Delta\tau\cdot\sqrt{\|H\|} = 2/\sqrt{(1 + 1/Ç)}$ and $\mu\cdot\Delta\tau = 4\sqrt{Ç}/(1+Ç)$; and then the average factor by which each iteration-step would reduce $\|G\|$ ultimately would be $1 - 2/(1 + \sqrt{Ç})$, which is the smallest that the Answer above allows for any Gradient-Based iteration. Finally, if good estimates for $\|H\|$ and $\Delta\tau$ were available then an estimate for μ that mimics the correlation between optimal choices would be

$$\mu = \sqrt{(\ \|H\|\cdot(4 - \|H\|\cdot\Delta\tau^2)\ )} \quad \text{provided} \quad \Delta\tau < 2/\sqrt{\|H\|}\ .$$

But in Regime #1 AGD is not needed since GD converges there almost as quickly when its formula for new $\Delta\tau$ is invoked almost always, and with no prior knowledge of Ç nor $\|H\|$. If AGD is worth using at all, it is to escape sooner from Regime #0. That requires values of $\Delta\tau$ and μ that adapt to an unpredictable landscape so that our chosen gauge function, the Pseudo-Hamiltonian $Æ(y, (v + \text{new } v)/2)$, declines as much as it can at every iteration-step. That would be ideal.

To that end let's consider this expansion in even powers of $\Delta\tau$ of ...

$$\{\ Æ(y, (v + \text{new } v)/2) - Æ(\text{old } y, (v + \text{old } v)/2)\ \}/\Delta\tau\ + \ +\ \mu\cdot\{\ (3\|v\|^2 + (\text{old } v)\grave{}\bullet(\text{new } v))/4 + \|\text{old } v - \text{new } v\|^2\ \}\ \approx O(\Delta\tau^2)$$

because its term " $O(\Delta\tau^2)$ " tends to positive values as iterates y approach Regime #1. If $\Delta\tau$ is small enough that $O(\Delta\tau^2) \approx ¥\cdot\Delta\tau^2$ for a nearly constant ¥ , the expansion suggests changing $\Delta\tau$ to ...

$$\delta\tau := \Delta\tau/\sqrt{\max\{\ 0.8,\ \ 2 + 2\Delta Æ/(\mu\cdot V^2\cdot\Delta\tau)\ \}} \quad \text{wherein}$$

$$\Delta Æ := Æ(y, (v + \text{new } v)/2) - Æ(\text{old } y, (v + \text{old } v)/2) \quad \text{and}$$

$$V^2 := (3\|v\|^2 + (\text{old } v)\grave{}\bullet(\text{new } v))/4 + \|\text{old } v - \text{new } v\|^2\ .$$

If $\|H\|$ has been estimated and $\delta\tau < 2/\sqrt{\|H\|}$ then change μ to $\mu := \sqrt{(\ \|H\|\cdot(4 - \|H\|\cdot\delta\tau^2)\ )}$, else set $\mu := 1/\delta\tau$.

What happens next depends upon the sign of $\Delta Æ$:

If $\Delta Æ \geq 0$ then discard new y , new v , y , v , $ƒ(y)$, $G(y)$ and $Æ(y, (v + \text{new } v)/2)$; and then from saved $G(\text{old } y)$, $ƒ(\text{old } y)$ and old v recompute

$v := \text{old } v - (G(\text{old } y) + \mu\cdot(\text{old } v))\cdot\delta\tau/(1 + \mu\cdot\delta\tau/2)$;
$y := \text{old } y + v\cdot\delta\tau$;
$Æ(\text{old } y, (v + \text{old } v)/2)$; $ƒ(y)$; $G(y)$;
$\text{new } v := v - (G(y) + \mu\cdot v)\cdot\delta\tau/(1 + \mu\cdot\delta\tau/2)$;
$\text{new } y := y + (\text{new } v)\cdot\delta\tau$;
$Æ(y, (v + \text{new } v)/2)$; $\Delta Æ$; V^2 .

This recomputation is so expensive that we have to hope it happens only rarely. Otherwise change " $\sqrt{\max\{\ 0.8. \ldots}$" to " $\sqrt{\max\{\ 0.9. \ldots}$".

If $\Delta Æ < 0$ then overwrite $\Delta\tau := \delta\tau$, abandon old values and rename current and new ones. For instance, rename v to old v and new v to v . Rather than copy v onto old v etc., change pointers to arrays of large dimensions lest more time be spent on memory movement than on arithmetic.

When do iterates y escape from Regime #0 ? A symptom is dwindling lengths ‖G‖ and ‖v‖ ; but they may also portend approaches to a Saddle-Point, which is sometimes foreshadowed by this inequality:

$$0 \geq v`\bullet(G(y) - G(\text{old } y))$$
$$\approx \Delta\tau \cdot v`\bullet H(x)\bullet v \pm O(\Delta\tau^3) .$$

Occasionally adding to an iterate y a small random vector orthogonal to both G and v often speeds the detection of and departure from a nearby saddle-point.

What remains to be determined by experiments is whether AGD escapes from Regime #0 faster than GD does using its formula for $\delta\tau$.

::

Notes still under construction:
Quit when ‖G‖ becomes negligible? What's "negligible" ?
Lots of Examples of AGD and GD at work and compared with GD+M .
What's wrong with "Stochastic GD" ?