Optimization for Machine Learning

Clément Lalanne

Where optimization enters

Statistical learning produces the same problem over and over: minimize a function of a parameter,

$$\hat{\theta} \approx \arg\min_{\theta \in \Theta} F(\theta), \qquad \Theta = \mathbb{R}^d.$$

Example. Regularized empirical risk minimization, with $F(\theta) = \widehat{R}_S(f_\theta) + \Omega(\theta)$, that is

$$F(\theta) = \frac{1}{n} \sum_{i \le n} \ell(y_i, f_\theta(x_i)) + \Omega(\theta).$$

The difficulty. Outside least squares and a few exponential families, the $\arg\min$ has no closed form. There is no formula to return.

The answer. Return the output of an algorithm that approximates the $\arg\min$. That is why the sign above is $\approx$ and not $\in$, and the whole lecture is about what the $\approx$ costs.

Notation for this lecture

The objective is $F : \mathbb{R}^d \to \mathbb{R}$, differentiable, minimized over $\theta \in \mathbb{R}^d$, where d is the number of parameters. Generic points of $\mathbb{R}^d$ are θ, η and ξ: many texts write η for a step size, here the step is always γ. Also $\langle \cdot, \cdot \rangle$ is the Euclidean inner product, $\| \cdot \|_2$ the Euclidean norm and $\| \cdot \|_{\mathrm{op}}$ the operator norm.

Iterates. θ_t is the iterate at step t and $\gamma_t > 0$ the step, or learning rate. In Part I only, $\theta(t)$ with a parenthesis is the solution of an ODE at time t. We write $\theta_\star$ for a minimizer of F and $F_\star = F(\theta_\star)$.

Data. $(x, y) \sim \mathbb{P}$ on $\mathcal{X} \times \mathcal{Y}$, $S = ((x_1, y_1), \dots, (x_n, y_n)) \sim \mathbb{P}^{\otimes n}$, loss ℓ, model f_θ, regularizer Ω, risks $\widehat{R}_S(f) = \frac{1}{n} \sum_{i \le n} \ell(y_i, f(x_i))$ and $R(f) = \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(y, f(x))]$, and $\theta' \in \arg\min_{\theta \in \Theta} R(f_\theta)$.

Randomness of the algorithm. $B_t \subseteq \{1, \dots, n\}$ is the batch drawn at step t, of size b, and g_t the stochastic gradient. $\mathbb{E}_{B_{1:t}}$ is the expectation over $B_1, \dots, B_t$, and $\mathbb{E}_{t-1}[\cdot] = \mathbb{E}_{B_t}[\cdot \mid B_1, \dots, B_{t-1}]$, the expectation over the batch of step t alone.

Constants. L is a smoothness constant, μ a strong convexity constant, $\kappa = L/\mu$ the condition number, G an almost sure bound on $\|g_t\|_2$ and D a bound on $\|\theta_0 - \theta_\star\|_2$.

Risk decomposition

Write $\hat{\theta}$ for what the algorithm returns and $\theta' \in \arg\min_{\theta \in \Theta} R(f_\theta)$ for the best parameter in the class. Then

$$R(f_{\hat{\theta}}) - R(f_{\theta'}) = \{R(f_{\hat{\theta}}) - \widehat{R}_S(f_{\hat{\theta}})\} + \{\widehat{R}_S(f_{\hat{\theta}}) - \widehat{R}_S(f_{\theta'})\} + \{\widehat{R}_S(f_{\theta'}) - R(f_{\theta'})\}.$$

Bounding the first and the third by a supremum, and the second by replacing θ' with the best parameter for the empirical risk,

$$R(f_{\hat{\theta}}) - R(f_{\theta'}) \leq 2 \sup_{\theta \in \Theta} |R(f_\theta) - \widehat{R}_S(f_\theta)| + \left(\widehat{R}_S(f_{\hat{\theta}}) - \inf_{\theta \in \Theta} \widehat{R}_S(f_\theta) \right).$$

Reading the decomposition

The first term is the generalization gap, uniform over the class. It is what the four previous lectures bounded: Rademacher complexity, VC dimension, metric entropy, chaining. It decreases with n and increases with the richness of Θ.

The second term is the **optimization error**, the gap between the empirical risk the algorithm reached and the best empirical risk available. It has nothing to do with $\mathbb{P}$, and it is the only term the statistician controls after the data are collected.

This lecture bounds the second term, for a general differentiable F. Nothing below uses that F is an empirical risk, until the cost of one gradient is counted in Part III.

A consequence. The two terms are added. Driving the optimization error far below the first term buys nothing, since the sum stays of the same order. Part III turns that remark into a budget.

The fundamental idea of continuous optimization

Replace F near the current point by a Taylor expansion, and take the step that the expansion prescribes.

Why it can work. A Taylor expansion is a polynomial. Minimizing a polynomial of degree one over a small ball, or of degree two over $\mathbb{R}^d$, is something we can do in closed form.

Why it needs care. The expansion is only valid near the current point. The step has to be short enough that the model still describes F where we land, and that is exactly the role of γ_t, and exactly what the smoothness assumption of Part II will quantify.

Two orders, two algorithms. Order one gives gradient descent, order two gives Newton's method. Everything else in this lecture is a variation on one of the two.

First order dynamics, gradient descent

For a small displacement $\delta\theta$,

$$F(\theta + \delta\theta) = F(\theta) + \langle \nabla F(\theta), \delta\theta \rangle + o(\|\delta\theta\|_2).$$

At fixed length $\|\delta\theta\|_2 = r$, the first order term is most negative for $\delta\theta = -r\nabla F(\theta)/\|\nabla F(\theta)\|_2$, by Cauchy-Schwarz. The direction of steepest local descent is $-\nabla F(\theta)$.

Gradient descent.

- input: a starting point θ_0
- recursion: $\theta_t = \theta_{t-1} - \gamma_t \nabla F(\theta_{t-1})$, where $(\gamma_t)_{t\geq 1}$ is the sequence of steps, also called the **learning rate**

Second order dynamics, Newton's method

Expand the gradient rather than the function,

$$\nabla F(\theta + \delta\theta) = \underbrace{\nabla F(\theta) + \nabla^2 F(\theta)\delta\theta}_{T_2(\delta\theta)} + o(\|\delta\theta\|_2).$$

A critical point satisfies $\nabla F(\theta) = 0$. Asking $T_2(\delta\theta) = 0$ instead, and assuming $\nabla^2 F(\theta)$ invertible, gives $\delta\theta = -\nabla^2 F(\theta)^{-1}\nabla F(\theta)$.

Newton's method.

- input: a starting point θ_0
- recursion: $\theta_t = \theta_{t-1} - \gamma_t \nabla^2 F(\theta_{t-1})^{-1}\nabla F(\theta_{t-1})$

What one step costs

Take $F = \widehat{R}_S(f_\theta) + \Omega(\theta)$, with d parameters and n samples, and write c for the cost of one forward pass of f_θ. Backpropagation makes one gradient of one term cost $O(c)$, so

$$\nabla F: \quad O(nc)\ \text{time}, \quad O(d)\ \text{memory}, \qquad \nabla^2 F: \quad O(ncd)\ \text{time}, \quad O(d^2)\ \text{memory},$$

and solving the Newton system costs a further $O(d^3)$.

Numerically. A small network has $d = 10^7$. Its Hessian has 10^{14} entries, that is 400 terabytes in single precision, and inverting it is 10^{21} operations. Newton's method is never run as written on a learning problem.

Part I. The dynamical systems point of view

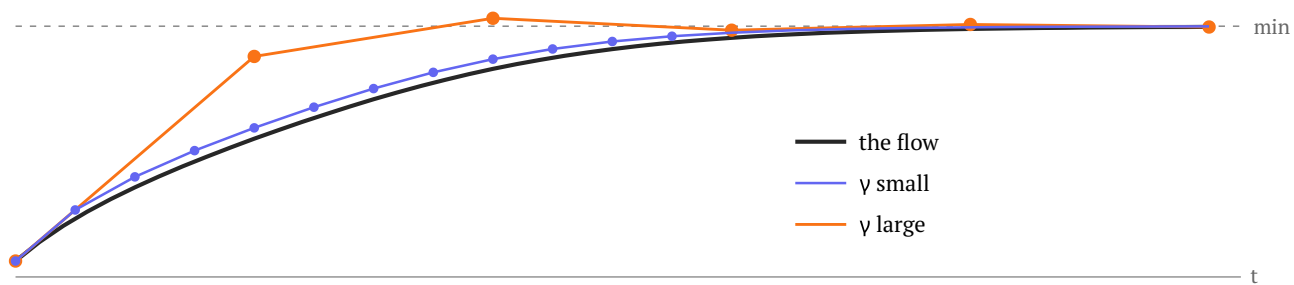
Gradient descent is an Euler scheme

When $\sup_t \gamma_t \ll 1$, the recursion $\theta_t = \theta_{t-1} - \gamma_t \nabla F(\theta_{t-1})$ is the explicit Euler discretisation, with step γ_t, of the ordinary differential equation

$$\frac{d}{dt}\theta(t) = -\nabla F(\theta(t)), \quad \theta(0) = \theta_0,$$

called the **gradient flow** of F.

The point of the substitution is that the ODE has no step size and no discretisation error. Whatever is true of the flow is what gradient descent is trying to imitate, and the step size is the only thing that separates them.



The objective decreases along the flow

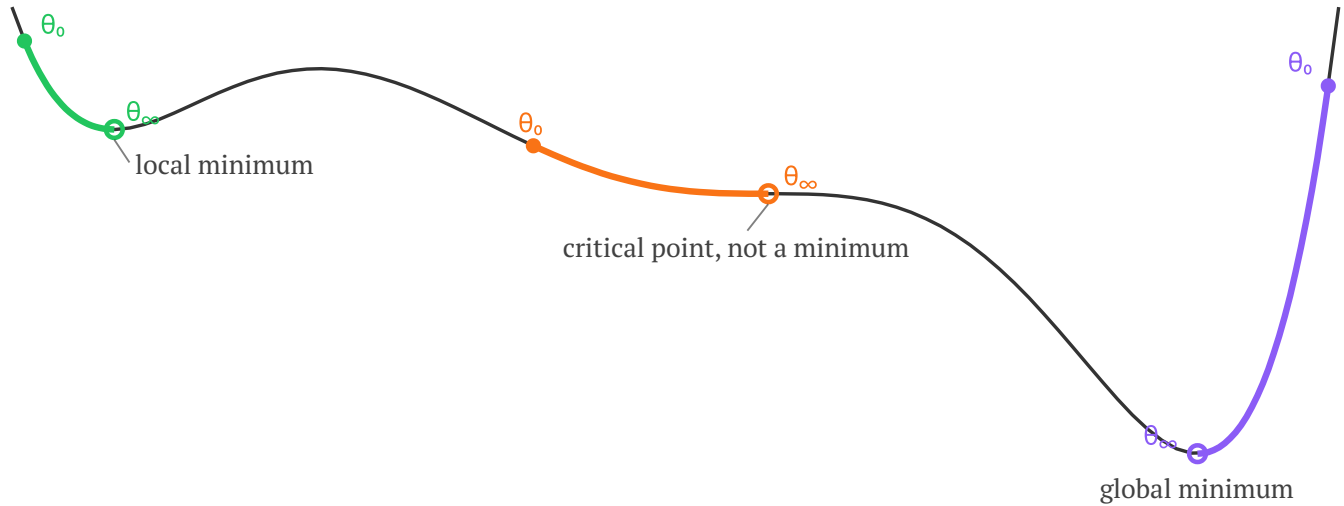
Along a solution of the gradient flow,

$$\begin{aligned}
\frac{d}{dt} F(\theta(t)) &= \left\langle \nabla F(\theta(t)), \frac{d}{dt}\theta(t) \right\rangle && \text{chain rule} \\
&= -\langle \nabla F(\theta(t)), \nabla F(\theta(t)) \rangle && \text{gradient flow} \\
&= -\|\nabla F(\theta(t))\|_2^2 \\
&\leq 0.
\end{aligned}$$

So $t \mapsto F(\theta(t))$ is non-increasing, and strictly decreasing wherever $\nabla F(\theta(t)) \neq 0$.

The flow can therefore only stop at a critical point, and it stops at the first one it meets going downhill. Which one that is depends on θ_0, and nothing here says it is a minimum.

The gradient flow, in a picture



Three starting points, three limits. The flow is deterministic and monotone in F, so the limit is decided entirely by θ_0.

Newton's method as a flow

The same substitution applied to Newton's recursion gives

$$\frac{d}{dt}\theta(t) = -\nabla^2 F(\theta(t))^{-1}\nabla F(\theta(t)).$$

Multiplying by $\nabla^2 F(\theta(t))$ and using the chain rule on $t \mapsto \nabla F(\theta(t))$,

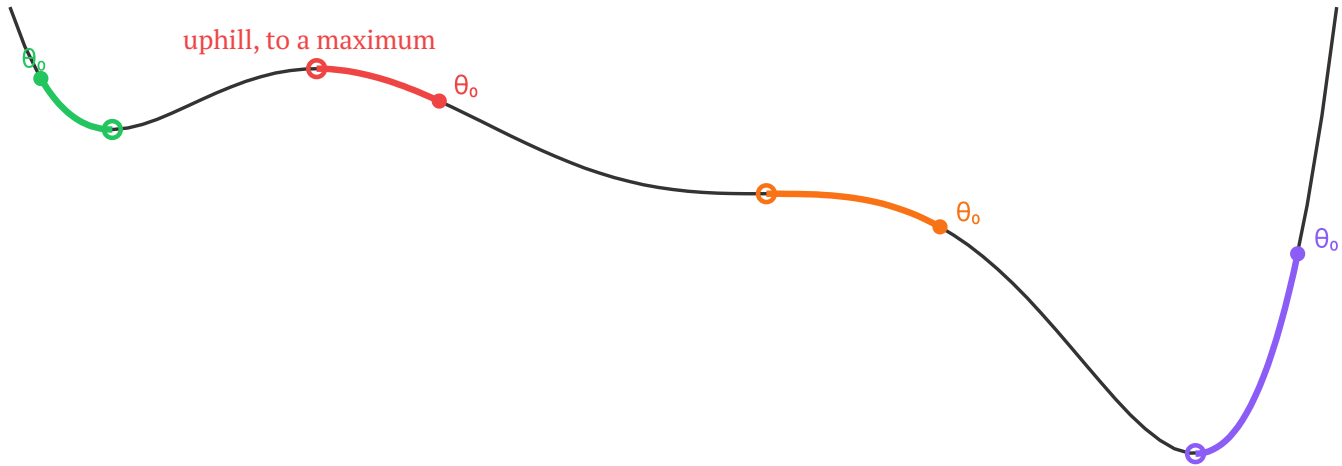
$$\frac{d}{dt}\nabla F(\theta(t)) = \nabla^2 F(\theta(t))\frac{d}{dt}\theta(t) = -\nabla F(\theta(t)),$$

hence,

$$\nabla F(\theta(t)) = e^{-t}\nabla F(\theta_0).$$

The gradient decays exponentially, at a rate that does not depend on F, on θ_0, or on the conditioning of the problem. Newton's flow solves $\nabla F = 0$, and it does so at the same speed everywhere.

What Newton's flow converges to



The four limits are the four critical points of F: a local minimum, a local maximum, a point where the derivative vanishes without changing sign, and the global minimum.

Newton is not a descent method

Warning. Nothing in $\nabla F(\theta(t)) = e^{-t}\nabla F(\theta_0)$ says that F decreases. Newton's flow converges to *a critical point*, which may be a maximum, and the picture above shows it happening twice.

Why. Gradient descent minimizes F. Newton's method solves $\nabla F = 0$, a different problem with the same solutions plus the spurious ones.

The fix. Require $\nabla^2 F(\theta) \succeq \mu I$, which forces every critical point to be the global minimum. That is the strong convexity assumption of Part II.4, and it is the setting where Newton's method is used.

In machine learning. F is not convex, $\nabla^2 F$ is indefinite, and the direction $-\nabla^2 F(\theta)^{-1}\nabla F(\theta)$ points towards saddle points. This, and the $O(d^3)$ of the previous slide, are why Newton's method is absent from the rest of the lecture.

Part II. Convergence of gradient descent

1) Smooth functions

To analyse gradient descent we must be able to trust the local approximation of F by its order one Taylor polynomial. A quantitative form of that trust is the following.

Definition. F is **L-smooth** if for every θ and every η,

$$|F(\eta) - F(\theta) - \langle \nabla F(\theta), \eta - \theta \rangle| \leq \frac{L}{2} \|\theta - \eta\|_2^2.$$

Equivalently. If ∇F is L-Lipschitz for $\|\cdot\|_2$ then F is L-smooth, and if F is C^2 the two are equivalent to $\|\nabla^2 F(\theta)\|_{\mathrm{op}} \leq L$ for every θ.

Where the definition comes from

Lemma. If ∇F is L-Lipschitz for $\|\cdot\|_2$, then F is L-smooth.

Proof. Write the increment as an integral along the segment and use Cauchy-Schwarz,

$$\begin{aligned} &|F(\eta) - F(\theta) - \langle \nabla F(\theta), \eta - \theta \rangle| \\ &= \left| \int_0^1 \langle \nabla F(\theta + s(\eta - \theta)) - \nabla F(\theta), \eta - \theta \rangle ds \right| && \text{fundamental theorem} \\ &\le \int_0^1 L s \|\eta - \theta\|_2 \cdot \|\eta - \theta\|_2 ds = \frac{L}{2} \|\eta - \theta\|_2^2. && \text{Lipschitz} \quad \blacksquare \end{aligned}$$

Smoothness is a bound on how fast the gradient can turn. It is the only hypothesis that Part II.1 and Part II.2 make: no convexity anywhere.

The descent lemma

Proposition (descent lemma). If F is L-smooth, the iterates of gradient descent with $\gamma_t = 1/L$ satisfy, for every $t \geq 1$,

$$F(\theta_t) \leq F(\theta_{t-1}) - \frac{1}{2L}\|\nabla F(\theta_{t-1})\|_2^2.$$

Proof. Write $v = \nabla F(\theta_{t-1})$, so that $\theta_t = \theta_{t-1} - v/L$. Smoothness applied at θ_{t-1} gives

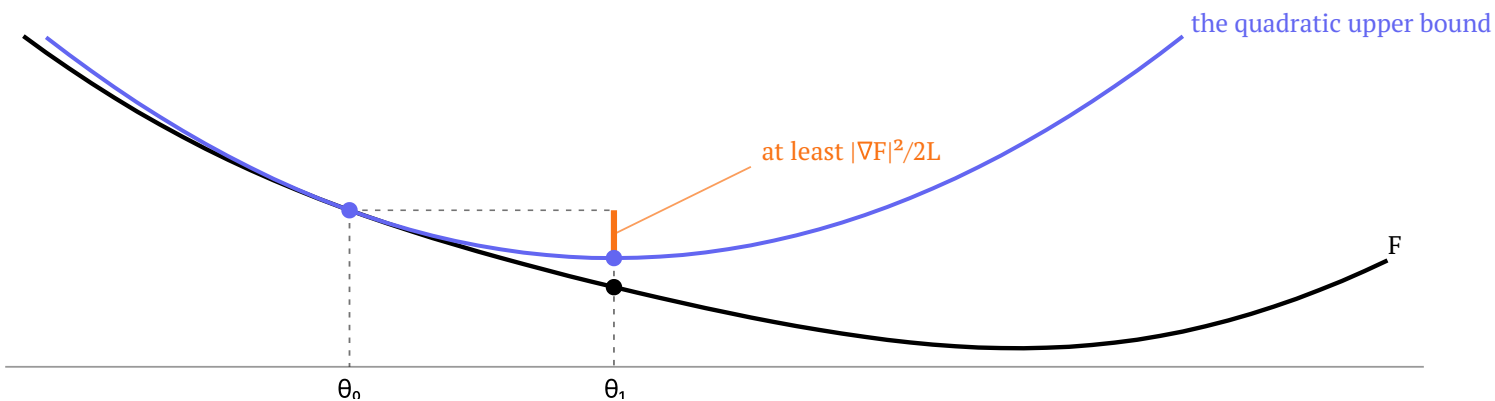
$$\begin{aligned} F(\theta_t) &\leq F(\theta_{t-1}) + \left\langle v, -\frac{v}{L} \right\rangle + \frac{L}{2}\left\| \frac{v}{L} \right\|_2^2 && L\text{-smoothness} \\ &= F(\theta_{t-1}) - \frac{\|v\|_2^2}{L} + \frac{\|v\|_2^2}{2L} = F(\theta_{t-1}) - \frac{\|v\|_2^2}{2L}. && \blacksquare \end{aligned}$$

Reading the descent lemma

It is an exact accounting. Each step buys a decrease of $\|\nabla F(\theta_{t-1})\|_2^2/2L$ up to the inequality. The bigger the gradient, the more the step is worth.

The step $1/L$ is not arbitrary. Smoothness gives the upper bound $F(\theta_{t-1}) + \langle v, \delta\theta\rangle + \frac{L}{2}\|\delta\theta\|_2^2$, a quadratic in $\delta\theta$ whose minimizer is exactly $\delta\theta = -v/L$. Gradient descent with $\gamma_t = 1/L$ is the exact minimization of the upper bound.

No convexity was used. The lemma is true on any smooth landscape.



2) The non-convex smooth case

Summing the descent lemma over the first t steps and telescoping,

$$\frac{1}{2L}\sum_{i=1}^{t}\|\nabla F(\theta_{i-1})\|_2^2 \leq F(\theta_0) - F(\theta_t) \leq F(\theta_0) - F_\star,$$

so that, dividing by t,

$$\frac{1}{2Lt}\sum_{i=1}^{t}\|\nabla F(\theta_{i-1})\|_2^2 \leq \frac{F(\theta_0) - F_\star}{t}.$$

The trajectory converges to a critical point in quadratic Cesàro mean.

Reading the non-convex rate

What is proved. The average squared gradient is $O(1/t)$. In particular

$$\min_{i<t} \|\nabla F(\theta_i)\|_2 \leq \sqrt{\frac{2L(F(\theta_0) - F_\star)}{t}},$$

so reaching a point with $\|\nabla F\|_2 \leq \epsilon$ takes $O(L(F(\theta_0) - F_\star)\epsilon^{-2})$ iterations.

What is not proved. Nothing about $F(\theta_t) - F_\star$. A small gradient is compatible with an arbitrarily large excess value, at a saddle or on a plateau.

Why this is the relevant statement in deep learning. The objective is not convex, the global minimum is out of reach and out of scope, and a critical point of low value is the whole of what an optimizer promises.

3) The convex smooth case

Proposition (co-coercivity). If F is L-smooth and convex, then for every θ and every η,

$$F(\theta) \geq F(\eta) + \langle \nabla F(\eta), \theta - \eta \rangle + \frac{1}{2L} \|\nabla F(\theta) - \nabla F(\eta)\|_2^2$$

$$\text{and} \quad \frac{1}{L} \|\nabla F(\theta) - \nabla F(\eta)\|_2^2 \leq \langle \nabla F(\theta) - \nabla F(\eta), \theta - \eta \rangle.$$

The first inequality sharpens the convexity inequality by a quadratic term. The second says that the gradient map is not just monotone, it is monotone with a quantitative margin.

Proof of co-coercivity

Fix θ and η. For every ξ, convexity at η and smoothness at θ give $m(\xi) \le F(\xi) \le M(\xi)$, where

$$m(\xi) = F(\eta) + \langle \nabla F(\eta), \xi - \eta \rangle, \quad M(\xi) = F(\theta) + \langle \nabla F(\theta), \xi - \theta \rangle + \frac{L}{2}\|\xi - \theta\|_2^2.$$

Hence $M - m \ge 0$ everywhere. It is a quadratic in ξ with Hessian LI, so it is minimized where its gradient vanishes,

$$\nabla F(\theta) - \nabla F(\eta) + L(\xi - \theta) = 0, \quad \text{that is} \quad \xi = \theta - \frac{1}{L}\left(\nabla F(\theta) - \nabla F(\eta)\right).$$

Proof of co-coercivity, continued

Write $\Delta = \nabla F(\theta) - \nabla F(\eta)$, so the minimizer is $\xi = \theta - \Delta/L$. Substituting,

$$\begin{aligned} 0 \le M(\xi) - m(\xi) &= F(\theta) - \frac{\langle \nabla F(\theta), \Delta \rangle}{L} + \frac{\|\Delta\|_2^2}{2L} - F(\eta) - \langle \nabla F(\eta), \theta - \eta \rangle + \frac{\langle \nabla F(\eta), \Delta \rangle}{L} \\ &= F(\theta) - F(\eta) - \langle \nabla F(\eta), \theta - \eta \rangle - \frac{\|\Delta\|_2^2}{2L}, \end{aligned}$$

which is the first inequality.

For the second, write the first inequality once as it stands and once with θ and η exchanged, and add. The values of F cancel, the two linear terms combine into one inner product, and $\|\Delta\|_2^2$ appears twice, leaving

$$0 \le \langle \nabla F(\theta) - \nabla F(\eta), \theta - \eta \rangle - \frac{1}{L}\|\Delta\|_2^2. \quad \blacksquare$$

The iterates come closer to the minimizer

Corollary. If F is convex and L-smooth and $\theta_\star$ minimizes F, the iterates of gradient descent with $\gamma_t = 1/L$ satisfy

$$\|\theta_t - \theta_\star\|_2^2 \le \|\theta_{t-1} - \theta_\star\|_2^2 - \frac{1}{L}\langle \nabla F(\theta_{t-1}), \theta_{t-1} - \theta_\star \rangle.$$

By convexity the inner product is at least $F(\theta_{t-1}) - F_\star \ge 0$, so the distance to $\theta_\star$ is non-increasing, and the amount by which it decreases is the excess value. The corollary converts progress in value into progress in distance, which is what the telescoping of the next slide needs.

Proof of the corollary

Write $v = \nabla F(\theta_{t-1})$ and recall $\nabla F(\theta_\star) = 0$. Then

$$
\begin{aligned}
\|\theta_t - \theta_\star\|_2^2 &= \left\|\theta_{t-1} - \theta_\star - \frac{v}{L}\right\|_2^2 = \|\theta_{t-1} - \theta_\star\|_2^2 - \frac{2}{L}\langle v, \theta_{t-1} - \theta_\star\rangle + \frac{\|v\|_2^2}{L^2} && \text{expanding} \\
&= \|\theta_{t-1} - \theta_\star\|_2^2 - \frac{2}{L}\langle v, \theta_{t-1} - \theta_\star\rangle + \frac{\|v - \nabla F(\theta_\star)\|_2^2}{L^2} && \nabla F(\theta_\star) = 0 \\
&\le \|\theta_{t-1} - \theta_\star\|_2^2 - \frac{2}{L}\langle v, \theta_{t-1} - \theta_\star\rangle + \frac{1}{L}\langle v - \nabla F(\theta_\star), \theta_{t-1} - \theta_\star\rangle && \text{co-coercivity} \\
&= \|\theta_{t-1} - \theta_\star\|_2^2 - \frac{1}{L}\langle v, \theta_{t-1} - \theta_\star\rangle. && \blacksquare
\end{aligned}
$$

The convex rate

Theorem. If F is convex and L-smooth and $\theta_\star$ minimizes F, the iterates of gradient descent with $\gamma_t = 1/L$ satisfy

$$F(\theta_t) - F_\star \le \frac{L}{t+1}\|\theta_0 - \theta_\star\|_2^2.$$

Proof. By the descent lemma the sequence $(F(\theta_i))_i$ is non-increasing, so its last term is below its average, and

$$\begin{aligned} F(\theta_t) - F_\star &\le \frac{1}{t+1}\sum_{i=0}^{t}(F(\theta_i) - F_\star) \;\le\; \frac{1}{t+1}\sum_{i=0}^{t}\langle \nabla F(\theta_i), \theta_i - \theta_\star\rangle && \text{convexity} \\ &\le \frac{L}{t+1}\sum_{i=0}^{t}\left(\|\theta_i - \theta_\star\|_2^2 - \|\theta_{i+1} - \theta_\star\|_2^2\right) \;\le\; \frac{L}{t+1}\|\theta_0 - \theta_\star\|_2^2. && \text{corollary, telescoping} \end{aligned}$$

■

4) Strongly convex functions, the heuristic

Return to the gradient flow, where $\frac{d}{dt} F(\theta(t)) = -\|\nabla F(\theta(t))\|_2^2$. Suppose the gradient is large whenever the value is far from optimal,

$$\|\nabla F(\theta)\|_2^2 \geq 2\mu \left(F(\theta) - F_\star\right) \quad \text{for every } \theta.$$

Then $h(t) = F(\theta(t)) - F_\star$ satisfies $h'(t) \leq -2\mu h(t)$, and Grönwall's lemma gives

$$F(\theta(t)) - F_\star \leq e^{-2\mu t} \left(F(\theta_0) - F_\star\right).$$

The convergence is exponential in time, not polynomial. The rest of this part identifies a hypothesis under which the displayed inequality holds, and transfers the conclusion to the discrete algorithm.

Strong convexity

Definition. F is **μ-strongly convex** if for every θ and every η,

$$F(\eta) \geq F(\theta) + \langle \nabla F(\theta), \eta - \theta \rangle + \frac{\mu}{2} \|\eta - \theta\|_2^2.$$

If F is C^2 this is equivalent to $\nabla^2 F(\theta) \succeq \mu I$ for every θ. Taking $\mu = 0$ recovers convexity.

A μ-strongly convex function has a unique minimizer: it is convex, it is coercive because the right-hand side above tends to infinity, and it is strictly convex.

Compare with L-smoothness, which gives the same expression with $\leq$ and L in place of $\geq$ and μ. A function that is both is trapped between two paraboloids, and necessarily $\mu \leq L$.

Strong convexity implies the Łojasiewicz inequality

Proposition. If F is differentiable and μ-strongly convex with minimizer $\theta_\star$, then for every θ,

$$\|\nabla F(\theta)\|_2^2 \geq 2\mu\,(F(\theta) - F_\star)\,.$$

Proof. The right-hand side of the definition is a quadratic in η, minimized at $\eta = \theta - \nabla F(\theta)/\mu$. Taking the infimum over η on both sides of the definition,

$$F_\star = \inf_\eta F(\eta) \;\geq\; \inf_\eta \left(F(\theta) + \langle \nabla F(\theta), \eta - \theta \rangle + \frac{\mu}{2}\|\eta - \theta\|_2^2 \right) = F(\theta) - \frac{1}{2\mu}\|\nabla F(\theta)\|_2^2. \qquad \blacksquare$$

This is exactly the hypothesis the heuristic needed, and it is the only consequence of strong convexity used below.

Linear convergence

Theorem. If F is L-smooth and μ-strongly convex, gradient descent with $\gamma_t = 1/L$ satisfies

$$F(\theta_t) - F_\star \leq \left(1 - \frac{1}{\kappa}\right)^t (F(\theta_0) - F_\star) \leq e^{-t/\kappa} (F(\theta_0) - F_\star), \quad \kappa = \frac{L}{\mu}.$$

Proof. Put the Łojasiewicz inequality into the descent lemma,

$$\begin{aligned} F(\theta_t) - F_\star &\leq F(\theta_{t-1}) - F_\star - \frac{1}{2L}\|\nabla F(\theta_{t-1})\|_2^2 && \text{descent lemma} \\ &\leq F(\theta_{t-1}) - F_\star - \frac{\mu}{L}(F(\theta_{t-1}) - F_\star) = \left(1 - \frac{1}{\kappa}\right)(F(\theta_{t-1}) - F_\star), && \text{Łojasiewicz} \end{aligned}$$

and iterate. The second inequality of the statement is $1 - u \leq e^{-u}$. ■

The condition number

Reaching $F(\theta_t) - F_\star \le \epsilon$ takes $\kappa \log((F(\theta_0) - F_\star)/\epsilon)$ iterations. The dependence on ϵ is logarithmic, so accuracy is cheap, and the dependence on the **condition number** $\kappa = L/\mu$ is linear, so conditioning is what the bill is made of.

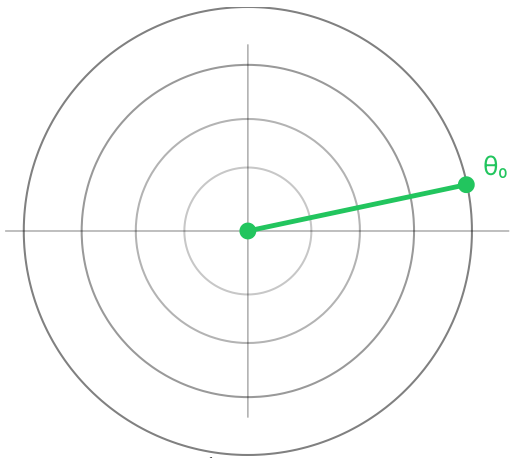
What κ measures. On a quadratic $F(\theta) = \frac{1}{2}\langle A\theta, \theta\rangle$ with $A \succ 0$, L and μ are the largest and smallest eigenvalues of A, and κ is the eccentricity of the level ellipsoids. At $\gamma = 1/L$ gradient descent kills the steep direction in one step and then crawls along the flat one, at rate $1 - 1/\kappa$ per step.

Rescaling changes κ. Multiplying one coordinate of θ by 10^3, which is what happens when one feature is in metres and another in kilometres, multiplies κ by 10^6. Standardizing the features is not cosmetic, it is a change of κ.

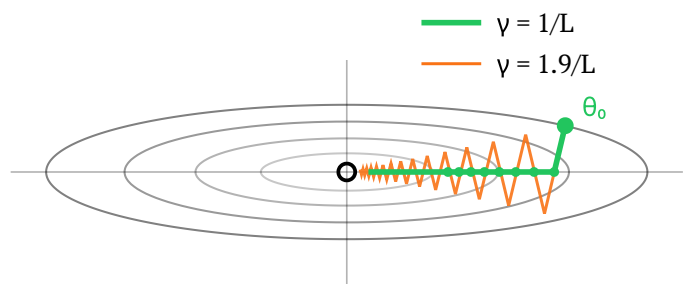
Newton removes κ. Newton's flow decayed at rate 1 regardless of F. Every practical method between the two, and Adam in Part IV, is an attempt to buy part of that for less than $O(d^3)$.

The condition number, in a picture

Level sets of $F(\theta) = \frac{1}{2}(\mu\theta_1^2 + L\theta_2^2)$, drawn to scale so that the eccentricity is the real one, with gradient descent started at the same point.



κ = 1, one step



κ = 20, forty-five steps and not there

At $\gamma = 1/L$ the steep direction is killed in a single step, and the whole cost is the crawl along the flat one, at rate $1 - 1/\kappa$ per step. The zigzag of the folklore picture is what a step larger than $1/L$ does: it is faster across the valley and no faster along it.

The four regimes

hypotheses	what converges	rate	iterations for ϵ
L-smooth	the squared gradients, in Cesàro mean	$O(1/t)$	$O(\epsilon^{-2})$ to reach an ϵ-critical point
L-smooth, convex	$F(\theta_t) - F_\star$	$O(1/t)$	$O(\epsilon^{-1})$
L non-smooth, convex	$F(\theta_t) - F_\star$	$O(1/\sqrt{t})$	$O(\epsilon^{-2})$
L-smooth, μ-strongly convex	$F(\theta_t) - F_\star$	$O(e^{-t/\kappa})$	$O(\kappa \log(1/\epsilon))$
L-smooth, convex, with momentum	$F(\theta_t) - F_\star$	$O(1/t^2)$	$O(\epsilon^{-1/2})$

The first three rows are proved above, with $\gamma_t = 1/L$ throughout. The fourth is Nesterov's accelerated gradient, stated in Part IV and not proved here. Convex non-smooth is a consequence of the future SGD

Part III. Stochastic gradient descent

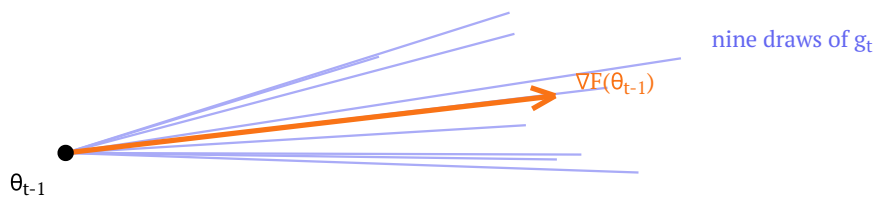
The idea

Replace $\nabla F(\theta_{t-1})$ by an **unbiased estimator** $g_t(\theta_{t-1})$ of it, meaning

$$\mathbb{E}_{t-1}\left[g_t(\theta_{t-1})\right] = \nabla F(\theta_{t-1}),$$

where $\mathbb{E}_{t-1}$ conditions on everything known at time $t-1$, in particular on θ_{t-1}.

Why. Because $g_t(\theta_{t-1})$ can be very much faster to compute than $\nabla F(\theta_{t-1})$, and the previous part showed that what a step needs is a descent direction, not the exact gradient.



individually wrong, on average exactly right

Where the estimator comes from

For $F(\theta) = \frac{1}{n}\sum_{i \le n} \ell(y_i, f_\theta(x_i))$, draw $B_t \subseteq \{1, \dots, n\}$ uniformly at random with $|B_t| = b$, and set

$$g_t(\theta) = \frac{1}{b} \sum_{i \in B_t} \nabla_\theta \ell(y_i, f_\theta(x_i)).$$

Each index is in B_t with probability b/n, so $\mathbb{E}_{t-1}[g_t(\theta)] = \nabla F(\theta)$ at every fixed θ: the **mini-batch** gradient is unbiased.

Complexity. One step costs $O(bc)$ instead of $O(nc)$, where c is the cost of one gradient of one term. With $n = 10^6$ and $b = 128$, that is four orders of magnitude per step.

The trade. Each step is n/b times cheaper but each step is worse. Part III is about this tradeoff.

The algorithm

Stochastic gradient descent.

- input: a starting point θ_0
- recursion: $\theta_t = \theta_{t-1} - \gamma_t g_t(\theta_{t-1})$, where $(\gamma_t)_{t \geq 1}$ is the sequence of steps

Two hypotheses carry the whole part.

(H1) Unbiasedness. $\mathbb{E}_{t-1}[g_t(\theta_{t-1})] = \nabla F(\theta_{t-1})$.

(H2) Controlled second moment. $\|g_t(\theta_{t-1})\|_2^2 \leq G^2$ almost surely.

Under (H1) and (H2), $\|\nabla F(\theta)\|_2 = \|\mathbb{E}_{t-1}[g_t(\theta)]\|_2 \leq G$ by Jensen, so F is automatically G-Lipschitz. The same G appears in both roles below for that reason.

1) The general smooth case

Lemma (descent lemma for SGD). If F is L-smooth and (H1) holds, then for every $t \geq 1$,

$$\mathbb{E}_{t-1}\left[F(\theta_t)\right] \leq F(\theta_{t-1}) - \gamma_t \|\nabla F(\theta_{t-1})\|_2^2 + \frac{L}{2}\gamma_t^2 \mathbb{E}_{t-1}\left[\|g_t(\theta_{t-1})\|_2^2\right].$$

Proof. L-smoothness at θ_{t-1}, with $\theta_t - \theta_{t-1} = -\gamma_t g_t(\theta_{t-1})$, gives

$$F(\theta_t) \leq F(\theta_{t-1}) - \gamma_t \langle \nabla F(\theta_{t-1}), g_t(\theta_{t-1})\rangle + \frac{L}{2}\gamma_t^2 \|g_t(\theta_{t-1})\|_2^2.$$

Take $\mathbb{E}_{t-1}$. The point θ_{t-1} is known at time $t-1$, so it comes out of the conditional expectation, and (H1) turns the inner product into $\|\nabla F(\theta_{t-1})\|_2^2$. ■

The rate in the smooth case

Take $\gamma_t = \gamma$ constant. Under (H2) the last term is at most $\frac{L}{2}\gamma^2 G^2$. Taking full expectations, summing over $i = 1, \dots, t$ and telescoping,

$$\frac{1}{t}\sum_{i=0}^{t-1} \mathbb{E}_{B_{1:t}}\left[\|\nabla F(\theta_i)\|_2^2\right] \leq \frac{F(\theta_0) - F_\star}{\gamma t} + \frac{L}{2}\gamma G^2.$$

With $\gamma = 1/\sqrt{t}$ for a horizon of t steps, the right-hand side is

$$\frac{1}{\sqrt{t}}\left(F(\theta_0) - F_\star + \frac{LG^2}{2}\right) = O\left(\frac{1}{\sqrt{t}}\right).$$

The two terms pull against each other

$$\underbrace{\frac{F(\theta_0) - F_\star}{\gamma t}}_{\text{forgetting } \theta_0} + \underbrace{\frac{L}{2}\gamma G^2}_{\text{noise floor}}$$

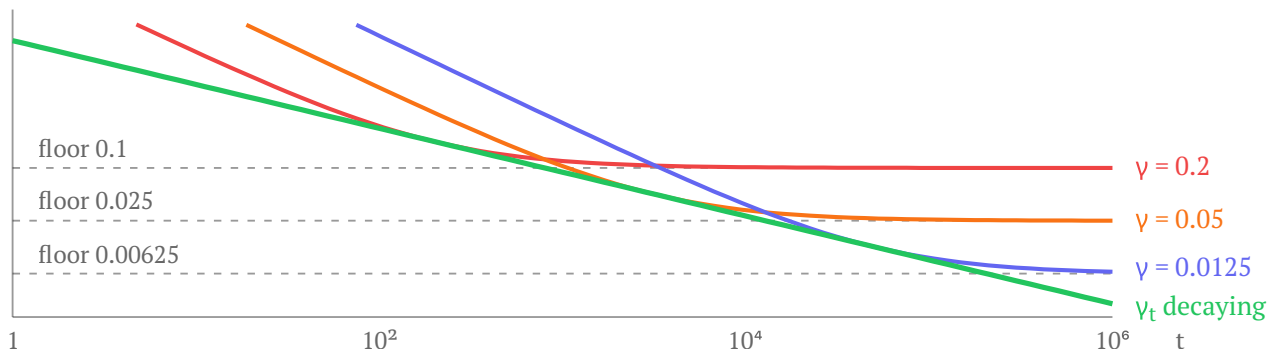
A large step forgets the starting point quickly and raises the floor. **A small step** lowers the floor and never leaves θ_0. The optimum of the sum is $\gamma = \sqrt{2(F(\theta_0) - F_\star)/(LG^2 t)}$, of order $1/\sqrt{t}$.

With γ constant, SGD does not converge. The second term does not depend on t: the iterates reach a neighbourhood of a critical point of size $O(\gamma)$ in the objective, and then orbit inside it forever.

This is the theoretical reason for decaying the step, and it is what is being observed when a training loss stops moving, the learning rate is divided by ten, and it starts moving again.

The noise floor, in a picture

The bound $\dfrac{F(\theta_0) - F_\star}{\gamma t} + \dfrac{L}{2}\gamma G^2$, drawn against t on logarithmic axes, for three constant steps and for $\gamma_t \propto 1/\sqrt{t}$.



Each constant step flattens onto its own dashed floor. Dividing γ by four divides the floor by four and multiplies by sixteen the number of steps needed to reach it, so the axis has to run to 10^6 before the smallest step has finished. Only the decaying step keeps descending, and it passes under every floor in turn.

2) The convex case

Theorem. Let F be convex, admitting a minimizer $\theta_\star$ with $\|\theta_0 - \theta_\star\|_2 \leq D$, and let (H1) and (H2) hold. With $\gamma_t = \frac{D}{G\sqrt{t}}$, the iterates of SGD satisfy

$$\mathbb{E}_{B_{1:t}}\left[F(\bar{\theta}_t) - F_\star\right] \leq DG\frac{2+\log t}{2\sqrt{t}}, \quad \text{where} \quad \bar{\theta}_t = \frac{\sum_{i=1}^t \gamma_i \theta_{i-1}}{\sum_{i=1}^t \gamma_i}.$$

No smoothness is assumed. What is bounded is the value at the weighted average of the iterates, not at the last one.

Proof, one step

Expanding the square, with $\mathbb{E} = \mathbb{E}_{B_{1:t}}$,

$$\begin{aligned}\mathbb{E}\left[\|\theta_t - \theta_\star\|_2^2\right] &= \mathbb{E}\left[\|\theta_{t-1} - \theta_\star - \gamma_t g_t(\theta_{t-1})\|_2^2\right] \\ &= \mathbb{E}\left[\|\theta_{t-1} - \theta_\star\|_2^2\right] - 2\gamma_t \mathbb{E}\left[\langle g_t(\theta_{t-1}), \theta_{t-1} - \theta_\star\rangle\right] + \gamma_t^2 \mathbb{E}\left[\|g_t(\theta_{t-1})\|_2^2\right].\end{aligned}$$

The last term is at most $\gamma_t^2 G^2$ by (H2). The middle one is handled by conditioning on the past.

Proof, the cross term

$$\begin{aligned}
\mathbb{E}\left[\langle g_t(\theta_{t-1}), \theta_{t-1} - \theta_\star \rangle\right] &= \mathbb{E}\left[\langle \mathbb{E}_{t-1}\left[g_t(\theta_{t-1})\right], \theta_{t-1} - \theta_\star \rangle\right] && \text{tower property, } \theta_{t-1} \text{ known at } t-1 \\
&= \mathbb{E}\left[\langle \nabla F(\theta_{t-1}), \theta_{t-1} - \theta_\star \rangle\right] && \text{(H1)} \\
&\geq \mathbb{E}\left[F(\theta_{t-1}) - F_\star\right]. && \text{convexity}
\end{aligned}$$

Substituting into the previous display and rearranging,

$$\gamma_t \mathbb{E}\left[F(\theta_{t-1}) - F_\star\right] \leq \frac{1}{2}\left(\mathbb{E}\left[\|\theta_{t-1} - \theta_\star\|_2^2\right] - \mathbb{E}\left[\|\theta_t - \theta_\star\|_2^2\right]\right) + \frac{\gamma_t^2 G^2}{2}.$$

Proof, summing

Summing over $i = 1, \dots, t$, telescoping the distances and dividing by $\sum_{i \le t} \gamma_i$,

$$\frac{\sum_{i \le t} \gamma_i \mathbb{E}\left[F(\theta_{i-1}) - F_\star\right]}{\sum_{i \le t} \gamma_i} \le \frac{\|\theta_0 - \theta_\star\|_2^2}{2 \sum_{i \le t} \gamma_i} + \frac{G^2}{2} \frac{\sum_{i \le t} \gamma_i^2}{\sum_{i \le t} \gamma_i}.$$

The left-hand side is a convex combination of the $\mathbb{E}[F(\theta_{i-1})]$ with weights $\gamma_i / \sum_{j \le t} \gamma_j$, so by Jensen it is at least $\mathbb{E}[F(\bar{\theta}_t)] - F_\star$, which is what the theorem bounds.

Proof, the two sums

With $\gamma_i = \frac{D}{G\sqrt{i}}$, and using $i \le t$ on the one hand and comparing with an integral on the other,

$$\sum_{i \le t} \gamma_i \ge \frac{D}{G} \sum_{i \le t} \frac{1}{\sqrt{t}} = \frac{D\sqrt{t}}{G}, \quad \sum_{i \le t} \gamma_i^2 = \frac{D^2}{G^2} \sum_{i \le t} \frac{1}{i} \le \frac{D^2}{G^2} \left(1 + \log t\right).$$

Substituting both, together with $\|\theta_0 - \theta_\star\|_2 \le D$,

$$\mathbb{E}\left[F(\bar{\theta}_t) - F_\star\right] \le \frac{G}{D\sqrt{t}} \left(\frac{D^2}{2} + \frac{G^2}{2} \cdot \frac{D^2(1 + \log t)}{G^2} \right) = DG \frac{2 + \log t}{2\sqrt{t}}. \quad \blacksquare$$

Reading the theorem

The rate is $1/\sqrt{t}$, against $1/t$ for gradient descent on the same class of functions. The noise costs a square root, and no choice of step recovers it: $1/\sqrt{t}$ is optimal for this class.

Averaging is not cosmetic. The bound is on $\bar{\theta}_t$. The last iterate θ_t has just absorbed a full noisy step, and controlling it requires more work and more hypotheses.

Nothing depends on d**.** D and G are Euclidean quantities, and the dimension enters only through them. This is what makes the method usable with $d = 10^9$.

Nothing depends on n **either.** The sample size does not appear in the statement, and it does not appear in the cost of one step. That is the next slide.

What it costs to reach accuracy ϵ

Multiply the number of iterations by the cost of one iteration, with c the cost of one gradient of one term.

method	iterations	per iteration	total
GD, convex	$O(LD^2/\epsilon)$	$O(nc)$	$O(ncLD^2/\epsilon)$
GD, μ-strongly convex	$O(\kappa \log(1/\epsilon))$	$O(nc)$	$O(nc\kappa \log(1/\epsilon))$
SGD, convex	$O(D^2G^2/\epsilon^2)$, up to log	$O(bc)$	$O(bcD^2G^2/\epsilon^2)$

> The total cost of SGD does not contain n.

How accurate does the optimization need to be?

The excess risk of the first part of the lecture is a sum: the generalization gap, of order $n^{-1/2}$ in every bound of the previous lectures, plus the optimization error ϵ. Taking ϵ much below $n^{-1/2}$ leaves the sum unchanged.

So set $\epsilon = n^{-1/2}$ and read off the totals, at fixed b and c:

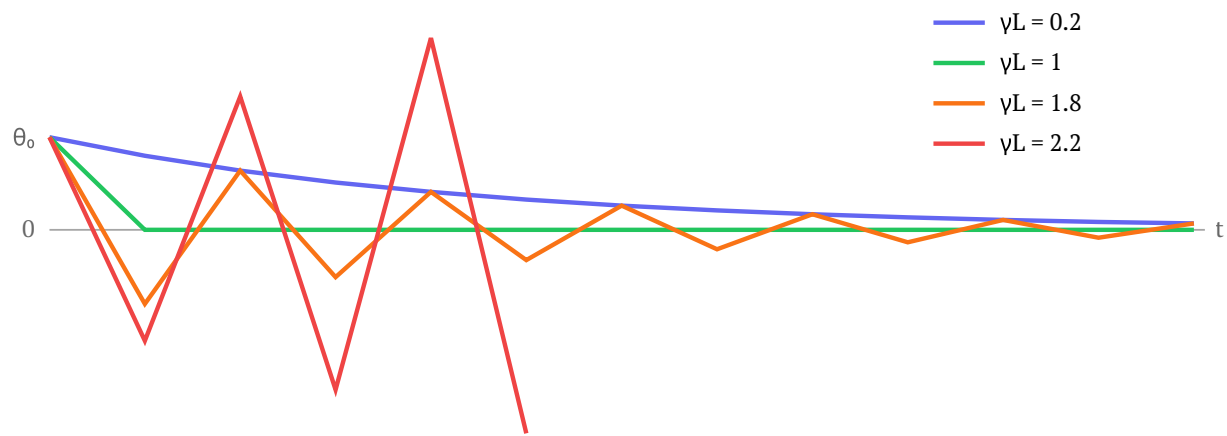
$$\text{GD, convex} : O(n^{3/2}), \quad \text{GD, strongly convex} : O(n\kappa \log n), \quad \text{SGD} : O(n).$$

For a fixed computing budget, the question is not which algorithm minimizes F best, but which reaches the smaller risk. SGD wins because it is the one whose bill is linear in the amount of data it is allowed to look at.

Part IV. And in practice ?

What the learning rate does

On $F(\theta) = \frac{L}{2}\theta^2$ in one dimension, gradient descent reads $\theta_t = (1 - \gamma L)\theta_{t-1}$, hence $\theta_t = (1 - \gamma L)^t \theta_0$: convergence exactly when $\gamma < 2/L$, and the fastest possible choice is $\gamma = 1/L$, which lands on the minimum in one step.



Choosing the learning rate in practice

The theory prescribes $\gamma = 1/L$. In practice L is unknown, and a global L is a large overestimate of the curvature almost everywhere, so $1/L$ would be far too small even if it were available.

The grid. Try $\gamma \in \{10^{-5}, 3 \cdot 10^{-5}, 10^{-4}, \dots\}$, a few hundred steps each. Keep the largest γ whose loss still goes down, then divide it by two or three.

The range test. One single run in which γ is multiplied by a constant factor every few steps, plotting the loss against γ. The curve falls, flattens, then explodes. Take γ an order of magnitude below the explosion.

What to watch. Divergence, not the final loss. A hundred steps are enough to see whether $\gamma > 2/L$, and a comparison of final losses needs full runs.

What not to do. A line search, as in classical optimization, needs several evaluations of F per step, and one evaluation of F is a full pass over the data. It costs more than it saves.

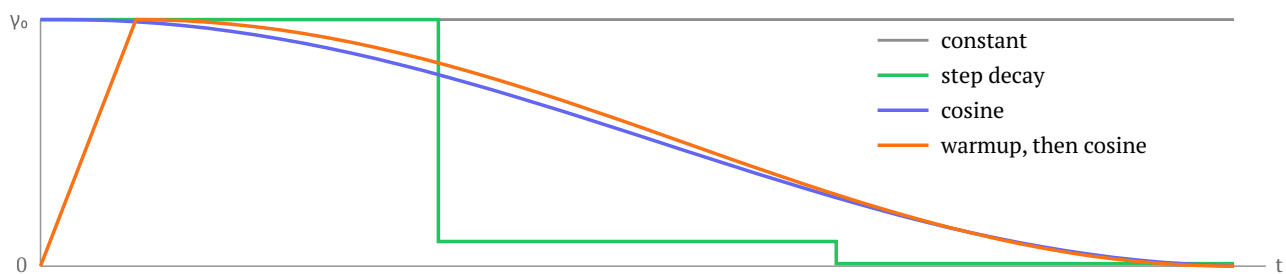
Schedules and warmup

Part III already requires a decaying step: with γ constant the noise floor $\frac{L}{2}\gamma G^2$ does not move.

What the theory gives. $\gamma_t \propto 1/\sqrt{t}$ in the convex case, $\gamma_t \propto 1/(\mu t)$ in the strongly convex case.

What is used. A constant step for most of training, then a decay to nearly zero: step decay, dividing γ by ten at fixed epochs, or cosine decay, $\gamma_t = \frac{\gamma_0}{2}\left(1 + \cos(\pi t/T)\right)$ over a horizon T fixed in advance.

Warmup. A linear increase from 0 to γ_0 over the first few thousand steps. At initialization the local curvature is much larger than it will be later, so the largest usable step is smaller at the start than in the middle.



Batch size, and the linear scaling rule

Compare k steps with batch b and step γ against one step with batch kb and step $k\gamma$, treating ∇F as constant over the k small steps. The displacements are

$$-\gamma \sum_{j=1}^{k} g_j^{(b)} \quad \text{and} \quad -k\gamma g^{(kb)},$$

with the same mean $-k\gamma \nabla F$ and the same variance $k\gamma^2 \sigma^2 / b$, where σ^2 is the variance of a single term.

Linear scaling rule. Multiplying the batch size by k and the learning rate by k leaves the first two moments of the trajectory unchanged. A large batch buys hardware efficiency, not fewer passes over the data, and the rule stops applying once $k\gamma$ approaches the ceiling $2/L$.

Momentum

Heavy ball. With $m_0 = 0$ and $\beta \in [0, 1)$,

$$m_t = \beta m_{t-1} + \nabla F(\theta_{t-1}), \quad \theta_t = \theta_{t-1} - \gamma m_t, \quad \text{so} \quad m_t = \sum_{i=1}^{t} \beta^{t-i} \nabla F(\theta_{i-1}).$$

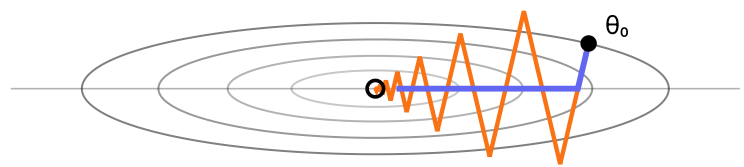
The step is an exponentially weighted average of the past gradients, over an effective horizon $1/(1-\beta)$, which is 10 for the usual $\beta = 0.9$.

In the continuous picture. The limit is $\ddot{\theta}(t) + a\dot{\theta}(t) + \nabla F(\theta(t)) = 0$, a ball of non-zero mass rolling in the landscape with friction a. The gradient flow of Part I is the massless limit. Inertia carries the ball along a narrow valley instead of letting it bounce between the walls.

What it buys. On a quadratic with condition number κ, tuned momentum needs $O(\sqrt{\kappa}\log(1/\epsilon))$ iterations instead of $O(\kappa \log(1/\epsilon))$. Nesterov's variant reaches $O(1/t^2)$ in the convex smooth case, and both are optimal among methods that only query gradients. Proofs omitted, see Nesterov.

Momentum, in a picture

The same quadratic as Part II, $\kappa = 20$, from the same starting point: gradient descent at $\gamma = 1/L$ against heavy ball at the optimal γ and β.



gradient descent, 45 steps, still short

heavy ball, at the minimum after about twenty

the rate per step is $1 - 1/\kappa = 0.95$ against $\sqrt{\beta} = 0.63$

Momentum overshoots the valley floor and crosses it repeatedly, which looks worse but is much faster: the inertia carries it along the flat direction instead of letting each step be scaled by μ alone.

Adam

All operations coordinatewise, with $m_0 = v_0 = 0$ and $g_t = g_t(\theta_{t-1})$,

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t \odot g_t, \quad \hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t},$$

$$\theta_t = \theta_{t-1} - \gamma \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \delta}, \qquad \beta_1 = 0.9, \quad \beta_2 = 0.999, \quad \delta = 10^{-8}.$$

$\hat{m}_t$ is momentum. Dividing by $\sqrt{\hat{v}_t}$ is a diagonal preconditioner: the cheap surrogate for the $\nabla^2 F^{-1}$ of Newton's method, at $O(d)$ time and $3d$ memory instead of $O(d^3)$ and d^2.

Where each term comes from

The update has three pieces and three separate origins.

Adagrad (Duchi, Hazan and Singer, 2011) divides by the accumulated squared gradient,

$$v_t = \sum_{i \le t} g_i \odot g_i, \qquad \theta_t = \theta_{t-1} - \gamma \frac{g_t}{\sqrt{v_t} + \delta}.$$

A coordinate that has seen large gradients gets a small step. The per-coordinate step decays like $1/\sqrt{t}$ whether or not the objective has stopped moving, which is right for sparse features and fatal for a deep network.

RMSProp (Tieleman and Hinton, 2012) replaces the sum by an exponential moving average, $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t \odot g_t$. The step no longer dies, and the horizon $1/(1 - \beta_2) = 1000$ becomes a choice rather than all of history.

Adam (Kingma and Ba, 2015) keeps that denominator, puts momentum in the numerator, and adds the bias correction that makes the first thousand steps usable.

Why Adam works, and why the bias correction

It is scale free per coordinate. $|\Delta\theta_j| \lesssim \gamma$ whatever the size of the gradient in coordinate j. The step is a trust region, not a gain, which is why $\gamma = 10^{-3}$ transfers between problems where an SGD step would have to be retuned by orders of magnitude.

The bias. Unrolling, $m_t = (1-\beta_1)\sum_{i\le t}\beta_1^{t-i} g_i$ and $(1-\beta_1)\sum_{i\le t}\beta_1^{t-i} = 1-\beta_1^t$. If the g_i have a common mean, m_t estimates $1-\beta_1^t$ times that mean, because $m_0 = 0$ drags the average towards zero. Dividing by $1-\beta_1^t$ removes the bias exactly, and likewise for v_t.

Why it cannot be skipped. The two biases do not cancel in the ratio: it is off by $(1-\beta_1^t)/\sqrt{1-\beta_2^t}$, which is 3.2 at $t = 1$ with the default constants. Without the correction the first steps are three times too long.

Adam on a constant gradient

Proposition. If $g_i = g$ for every $i \le t$, then $\hat{m}_t = g$ and $\hat{v}_t = g \odot g$ exactly, for every $t \ge 1$, and therefore

$$\theta_t - \theta_{t-1} = -\gamma \frac{g}{|g| + \delta},$$

coordinatewise, which is $-\gamma \operatorname{sign}(g)$ up to δ.

Proof. $m_t = (1-\beta_1)\sum_{i \le t} \beta_1^{t-i} g = g(1-\beta_1^t)$, so $\hat{m}_t = m_t/(1-\beta_1^t) = g$. The same computation with β_2 and $g \odot g$ gives $\hat{v}_t = g \odot g$, and $\sqrt{\hat{v}_t} = |g|$. ■

So a coordinate moves by γ per step, whatever its gradient is. That is a trust region claim: γ is a distance in parameter space, not a gain applied to a gradient. It is also why Adam needs a schedule: nothing in the update gets smaller as the minimum is approached.

Weight decay is not L^2 regularization

Warning. Put $\Omega(\theta) = \frac{\lambda}{2}\|\theta\|_2^2$ inside F. Its gradient $\lambda\theta$ is then divided by $\sqrt{\hat{v}_j} + \delta$ coordinatewise, so the effective penalty is $\lambda / \sqrt{\hat{v}_j}$: different in every coordinate, and depending on the history of the gradients. It is not the penalty that was intended.

AdamW. Apply the decay outside the adaptive scaling,

$$\theta_t = \theta_{t-1} - \gamma \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \delta} + \lambda \theta_{t-1} \right).$$

For plain SGD the two formulations give the same update, which is why the distinction went unnoticed for years. For Adam they differ.

What to remember

Both algorithms are one Taylor expansion. Order one gives gradient descent, order two gives Newton, which solves $\nabla F = 0$, reaches any critical point including maxima, and costs $O(d^3)$ a step.

Smoothness is what makes a step safe. The descent lemma buys $\|\nabla F(\theta_{t-1})\|_2^2/2L$ per step, and $\gamma = 1/L$ minimizes the quadratic upper bound exactly.

Each hypothesis buys one order. Smooth gives $O(1/t)$ on the gradients, convex $O(1/t)$ on the value, strongly convex $O(e^{-t/\kappa})$, momentum $O(1/t^2)$.

The noise costs a square root and creates a floor. SGD converges at $1/\sqrt{t}$, and at a constant step it stops at a noise floor of order γ. Decaying the step is what makes the limit exist.

SGD is the workhorse for a complexity reason. Its total cost to a fixed accuracy does not contain n, and since the optimization error only has to match $n^{-1/2}$, its bill is linear in n against $n^{3/2}$.