numerical computation

- typically refers to algorithms that solve mathematical problems by methods that update estimates of the solution via an iterative process, rather than analytically deriving a formula for the correct solution.
- because these are estimates, the concept of measuring how bad our estimates are becomes relevant.
- The fundamental difficulty in performing continuous math on a digital computer is that we need to represent infinitely many real numbers with a finite number of bit patterns.
- Rounding error is problematic, especially when it compounds across many operations, and can cause algorithms that work in theory to fail in practice if they are not designed to minimize the accumulation of rounding error.
- Underflow occurs when numbers near zero are rounded to zero. Many functions behave qualitatively differently when their argument is zero rather than a small positive number.
- Another highly damaging form of numerical error is overflow. Overflow occurs when numbers with large magnitude are approximated as ∞ or −∞. Further arithmetic will usually change these infinite values into not-a-number values
- All this stuff is especially relevant for developers of low level libraries.
- Conditioning refers to how rapidly a function changes with respect to small changes in its inputs. Functions that change rapidly when their inputs are perturbed slightly can be problematic for scientific computation because rounding errors in the inputs can result in large changes in the output.
- Objective, loss, cost all usually mean the same thing. objective is usually used when we are trying to maximize something, while other two are used when we minimize something. The optimal solution is represented with a $*$ superscript. Ascent problems are sometimes called hill climbing.
- points where $f(x) = 0$ are called critical or stationary points. critical points that are neither maxima nor minima are called saddle points.
- in very high dimension functions, there are tons of saddle points and relatively very few maxima/minima.
- gradient points in the direction of steepest ascent/descent.
- The strategy of gradient updates where you look for $f(\mathbf{x} - \epsilon \nabla_x f(\mathbf{x}))$ for varying values of ϵ until we find the best update is called line search.
- When the second partial derivatives are continuous, the differential operators are commutative. $\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}$. in this case, the hessian is symmetric (at the points where this is true)
- hessian is a matrix of second order partial derivatives (all possible ones). We can write multivariate taylor expansion using hessian like we do using derivatives in the univariate case.
- if hessian is positive semidefinite at a point, that point is a local minima.
- Naive gradient descent (first order optimization method) fails to exploit information in the curvature of the loss function. (the Hessian contains this information)
- Newton’s method is based on using a second order taylor series expansion to approximate $f(x)$ near some point $x = x_0$.
- A Lipschitz continuous function is a function whose rate of change is bounded by a Lipshitz constant for all x, y: $|f(x) - f(y)| = ||x - y||_2$
- deep learning doesn’t usually give convex optimization problems so its hard to find global minima.
- constrained optimization is a problem of optimizing a function with some constraints on the input. the feasible region is the set of all points that satisfy all constraints.

- The KKT (Karash Kuhn Tucker) approach is a way to convert a constrained optimization problem to an unconstrained optimization problem, using something called a generalized Lagrangian.
- The KKT approach is to write the optimization problem of $f(x)$ given $g_i(x) = 0$ and $h_i(x) \leq 0$ by defining the Lagrangian.

  $$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_i \lambda_i \, g^{(i)}(\boldsymbol{x}) + \sum_j \alpha_j h^{(j)}(\boldsymbol{x}).$$

  Then, if we solve

  $$\min \max \max_{, \;\geq 0} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\alpha})$$

  it is the same as solving

  $$\min_{\in \mathbb{S}} f(\boldsymbol{x}).$$

  because for any x that the constraint is violated,

  $$\max \max_{, \;\geq 0} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = \infty.$$

  and for x's where the constraint is not violated,

  $$\max \max_{, \;\geq 0} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = f(\boldsymbol{x}),$$

  Since the overall problem is a minimization problem, it will pick out the x's where the constraint is not violated so as to minimize the whole thing.
- We say that a constraint $h_i(x)$ is active if $h_i(x^*) = 0$.
- At the optimal solution,
  - the gradient of the generalized lagrangian is 0
  - all constraints on both x and the KKT multipliers are satisfied
  - the inequality constraints exhibit “complementary slackness”. That is $\alpha \odot \mathbf{h}(\mathbf{x}) = \mathbf{0}$. That is, for each inequality constraint, either $h_i(x) = 0$ or α_i is 0. To see why, notice that this will help maximize the last term.

  The above are called the **KKT conditions.**