Gradient Descent and Optimisation

ATML Track1

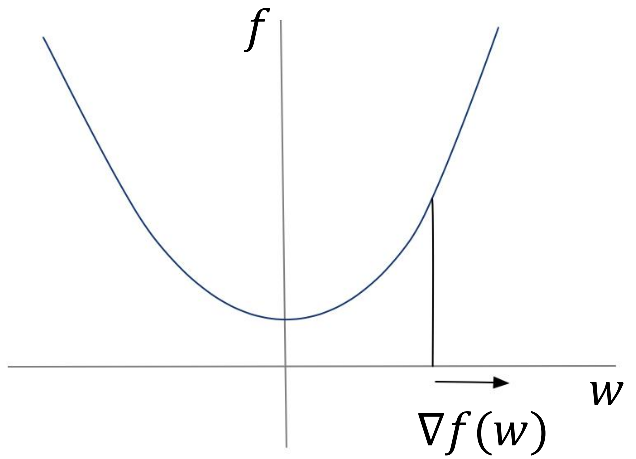
Rik Sarkar

Today

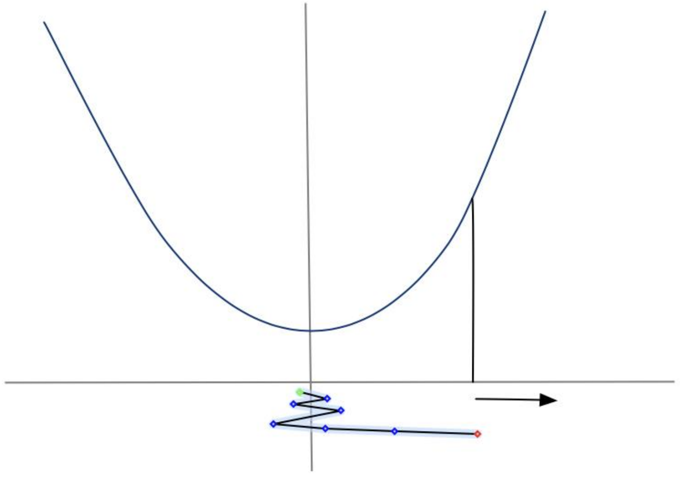
- Gradient descent
- Convexity
- Convergence of gradient descent
- Strong convexity
- Regularisation
- Stability

Gradient descent

- Gradient in 1d
  - $\nabla f = \frac{df(w)}{dw}$
  - The derivative as a vector
    - The direction and speed of increase of f

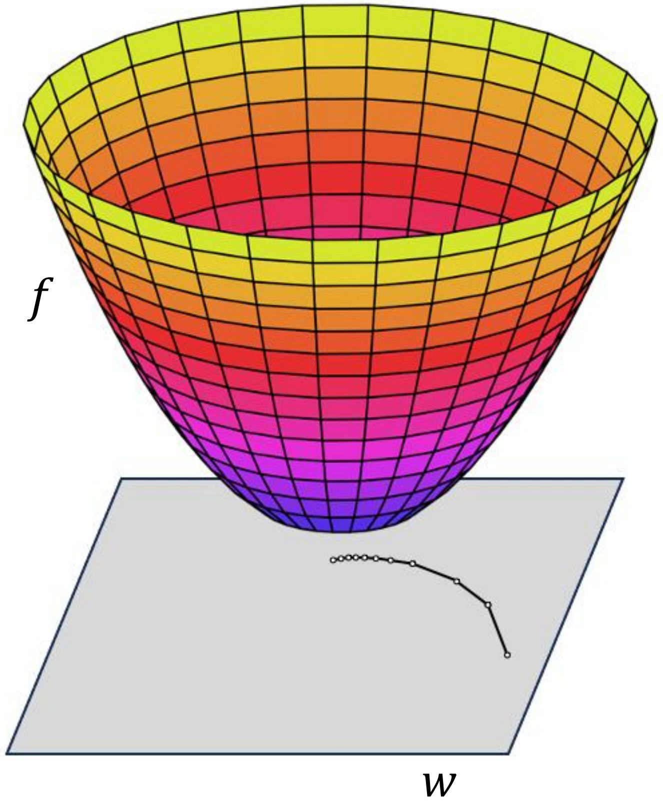


- We move opposite to the gradient $-\nabla f$
  - Step sizes proportional to the gradient
  - We might overshoot the min
  - But eventually converge to it



High dimensional gradients

- Our model parameter sets are vectors
  - $\boldsymbol{w} = [w_1, w_2, \dots, w_n]$
  - Each model is a point in high dimension
- High dimension Gradient
  - Take the gradient or derivative independently in each dimension and put them in an array
    - $\nabla f(\boldsymbol{w}) = [\frac{\partial f(\boldsymbol{w})}{\partial w_1}, \frac{\partial f(\boldsymbol{w})}{\partial w_2}, \dots, \frac{\partial f(\boldsymbol{w})}{\partial w_n}]$
  - The vector direction is the direction f increases the fastest
  - Length of the vector represents rate of increase



Gradient Descent algorithm

- Start with $\boldsymbol{w}^0$ initialised randomly
- At every step t :
  - $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \nabla f(\boldsymbol{w}^t)$
    - (Move in the direction that f decreases fastest With a step factor of η)

- After T steps, output the average vector $\bar{\boldsymbol{w}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{w}^t$
- Other version: output final vector $\boldsymbol{w}_{\boldsymbol{T}}$
- For us, f is the average loss L

Stopping gradient descent

- Stop after T steps
- Where we know what T is reasonable

- Convergence
  - What is a good T so that GD is close to the best model?

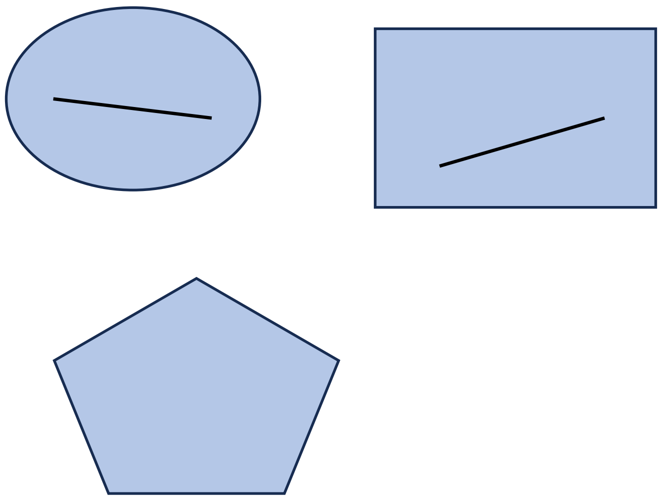
- We first need a few definitions

Convex sets

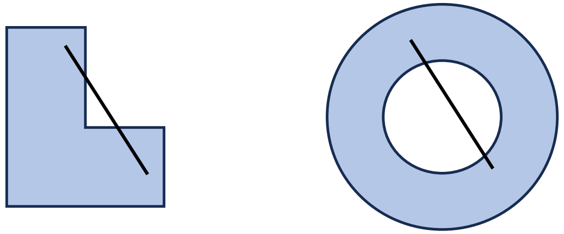
- A set C is convex if for any $\boldsymbol{u}, \boldsymbol{v} \in C$, the line segment connecting $\boldsymbol{u}, \boldsymbol{v}$ is in C.
  - Can be written formally as:
  - For any $\alpha \in [0,1]$, it is true that $\alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v} \in C$

- Observe that for points $\boldsymbol{u}, \boldsymbol{v}$
  - $\alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v}$, with $\alpha \in [0,1]$
  - Are points on the line segment connecting $\boldsymbol{u}, \boldsymbol{v}$

convex



Non convex

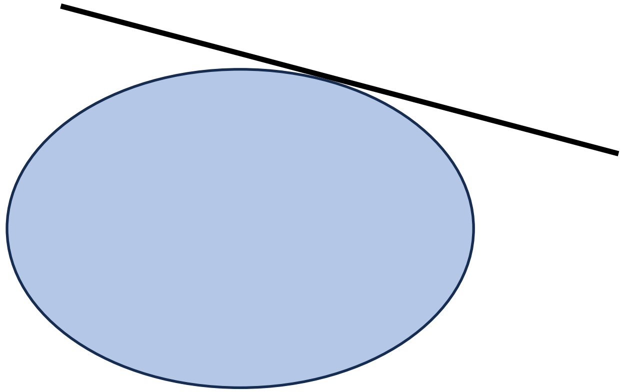


Convex sets

- Questions:
- Is $\mathbb{R}$ convex?
- Is $\mathbb{R}^n$ convex?

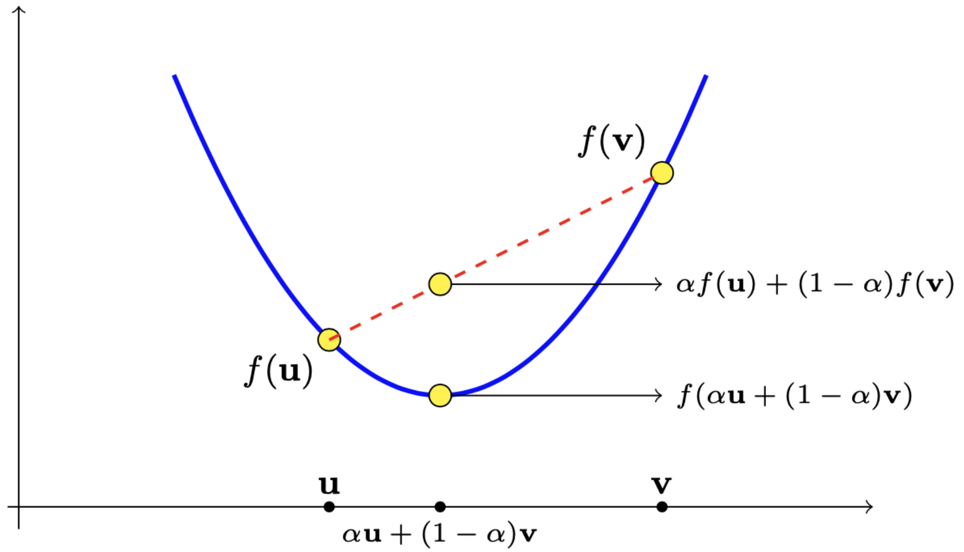
Convex sets: additional observation

- The set lies entirely on one side of a tangent to the boundary



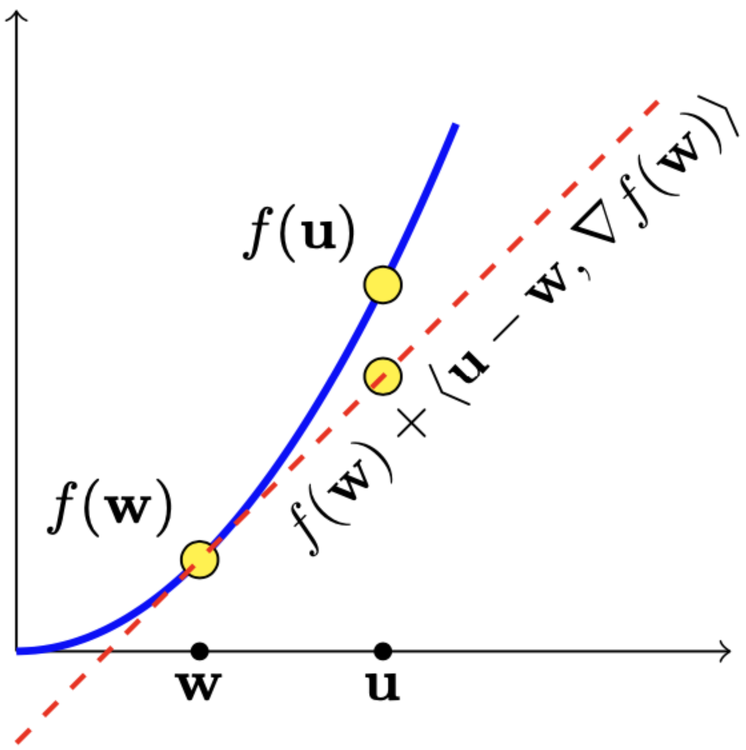
Convex function

- For a convex C, a function $f: C \rightarrow \mathbb{R}$ is convex if
- $f(\alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v}) \leq \alpha f(\boldsymbol{u}) + (1-\alpha) f(\boldsymbol{v})$
- The graph of f lies below the straight line connecting u and v
- A way to formalize the shape we have been drawing



Properties of convex functions

- For every $\boldsymbol{w}$ the tangent at $f(\boldsymbol{w})$lies below f:
  - $\forall \boldsymbol{u}, f(\boldsymbol{u}) \geq f(\boldsymbol{w}) + \langle \nabla f(\boldsymbol{w}), \boldsymbol{u} - \boldsymbol{w} \rangle$
- If $f: \mathbb{R} \to \mathbb{R}$ is twice differentiable, then
  - f is convex
  - f' is monotone nondecreasing
  - f'' is nonnegative
- Are equivalent



Combining convex functions

- If f_i are convex functions
- $g(x) = \max_i f_i(x)$ is convex
- $g(x) = \sum_i a_i f_i(x)$ is convex
  - What is the consequence for loss functions?

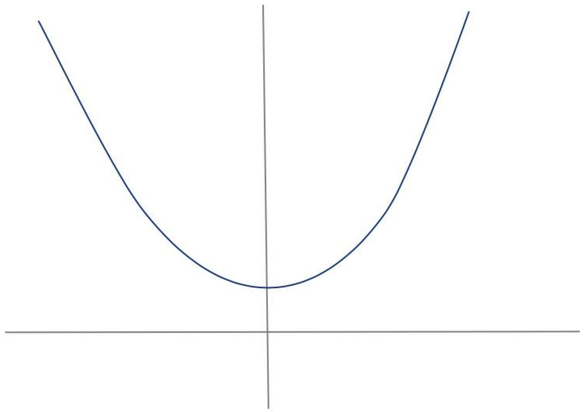
Combination of loss functions

- If $\ell(\cdot, x)$is convex for each $x \in S$
- Then the average empirical loss $L_S = \frac{1}{m}\sum_{x \in \mathrm{S}} \ell(\cdot, x)$ is convex

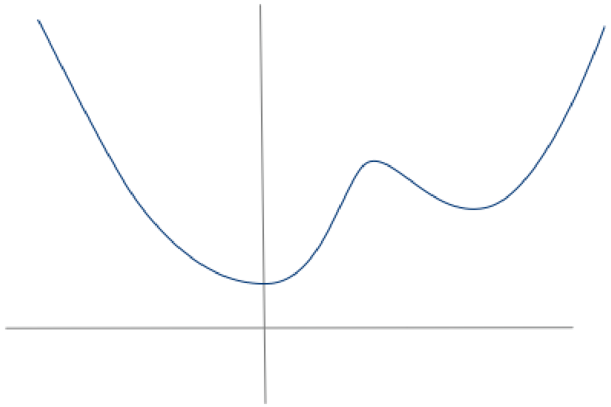
- Check that logistic loss is convex

Properties of convex functions

- If u is a local minimum, then it is a global minimum
  - No other point has a lower value



- Non-convex functions
  - There can be more than one local minima



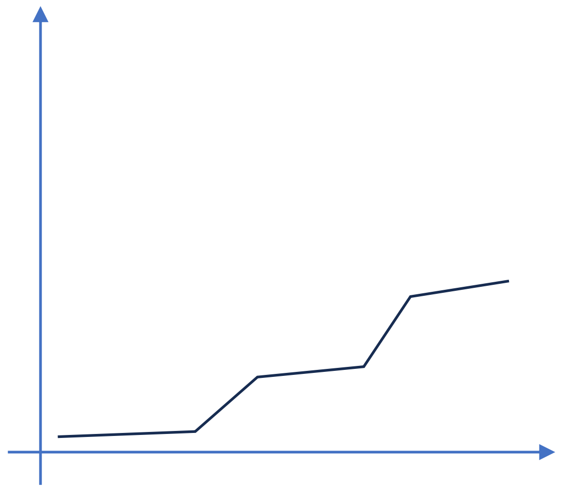
Convex function Question

- Is the the global minimum unique for convex functions?
- Can there be more than one point with the global min value?

Lipschitz and smooth functions

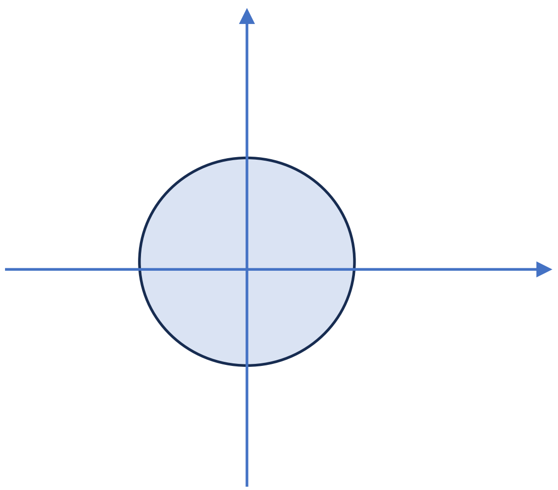
- A function f is ρ-Lipschitz if
  - $\left|\left|f(\boldsymbol{w}_1) - f(\boldsymbol{w}_2)\right|\right| \leq \rho||\boldsymbol{w}_1 - \boldsymbol{w}_2||$

- A function that does not change too fast
  - If the derivative ∇f is bounded by ρ,
    - Then the function is also ρ-Lipschitz
    - But lipschitzness can be defined/computed even when the derivative does not exist



- Smooth functions
  - f is β-smooth if ∇f is β-Lipschitz:
  - $\left|\left|\nabla f(\boldsymbol{v}) - \nabla f(\boldsymbol{w})\right|\right| \leq \beta||\boldsymbol{v} - \boldsymbol{w}||$

Boundedness



- A hypothesis class is bounded if
- $\forall w \in \mathcal{H}, \left|\left|\boldsymbol{w}\right|\right| \leq B$
- For some constant B
- That is, we are considering models only within a restricted ball of radius B

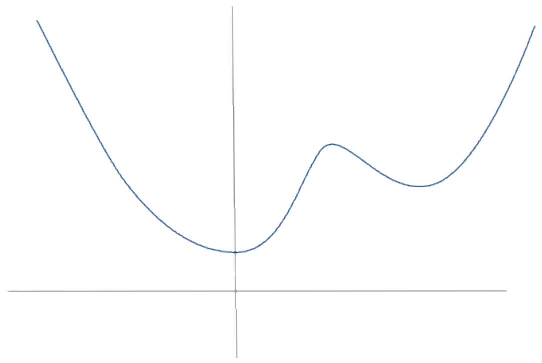
GD Convergence theorem

- For convex lipschitz bounded learning
- Setting $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$
- We can get $f(\overline{w}) - f(w^*) \leq \frac{B\rho}{\sqrt{T}}$

- Alternatively, to achieve $f(\overline{\boldsymbol{w}}) - f(\boldsymbol{w}^*) \leq \epsilon$ the number of rounds is:
- $T \geq \frac{B^2\rho^2}{\epsilon^2}$

Discussion: why do we need these properties

- For non-convex functions, there is no guarantee of getting close to the optimum value



- Lipschitz bound
  - Ensures that the steps are not so large that they take us very far from the action

- Boundedness
  - If we start unbounded distance from the min, it can take unbounded number of steps to get there
  - Sometimes it is assumed that everything occurs within radius $B = 1$ (after scaling) and is omitted from the discussion.

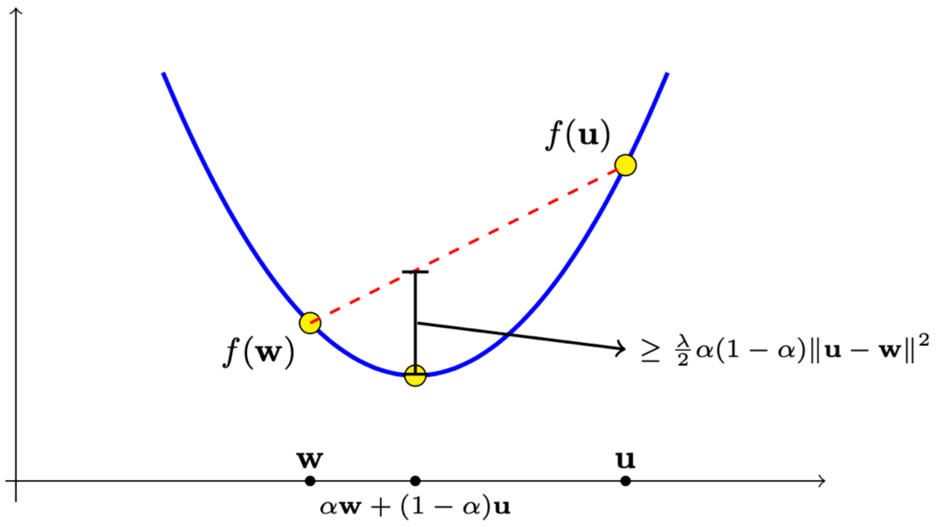
Strong Convexity

- Function f is λ-strongly convex if

$$f(\alpha\mathbf{w} + (1-\alpha)\mathbf{u}) \leq \alpha f(\mathbf{w}) + (1-\alpha) f(\mathbf{u}) - \frac{\lambda}{2}\alpha(1-\alpha)\|\mathbf{w}-\mathbf{u}\|^2$$

- Alternative definition:
  - $f(x)$ is λ-strongly convex
  - Iff $f(x) = g(x) + \frac{\lambda}{2}||x||^2$, where $g(x)$ is convex

- Strongly convex functions have unique global minimum
  - (If a minimum exists. There are some technicalities around mathematical existence of minimum that we don't need to worry about.)



Regularization

- Instead of the pure loss, minimize loss with a regularization term:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \left(L_S(\mathbf{w}) + R(\mathbf{w}) \right)$$

- Commonly used: $R(\boldsymbol{w}) = \lambda \left|\left|\boldsymbol{w}\right|\right|^2$
  - Called Tikhonov regularization

Try yourself:

Go to wolfram alpha and plot a polynomial: $y = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$

- With numbers of your choice in place of coefficients a_i
- Now scale the coefficients: multiply all the coefficients with the same number (may be fractions too). What do you see?

- $R(\boldsymbol{w}) = \lambda||\boldsymbol{w}||^2$
  - Is 2-strongly convex
- If $L_S(w)$ is convex, then $L_S(w) + R(W)$ is 2-strongly convex
- Strong convexity implies stability

Stability

- Intuitively: A learning algorithm is stable if
  - A small change to training set does not cause a big change to the output (model or hypothesis)

- This is a desirable property because…

Stability

- Intuitively: A learning algorithm is stable if
  - A small change to training set does not cause a big change to the output (model or hypothesis)

- This is a desirable property because
  - It implies that it is not too sensitive to specific S. does not overfit
  - If we continue to use it, it will not abruptly change behavior as new data comes in

- Suppose in S, we replace z_i with $z' \sim \mathcal{D}$
- Let us write this as S^i

- A good algorithm A should have small value for
  - $|\ell(A(S^i), z_i) - \ell(A(S), z_i)|$

- The loss at z_i does not depend too much on it being in the sample

Stability definition and result

- Algorithm A is on-average-replace-one-stable with rate $\epsilon(m)$
- If
  - $\mathbb{E}\left[\ell\left(A\left(S^i\right), z_i\right) - \ell(A(S), z_i)\right] \leq \epsilon(m)$

Stability definition and result

- Algorithm A is on-average-replace-one-stable with rate $\epsilon(m)$
- If
  - $\mathbb{E}\left[\ell\left(A\left(S^i\right), z_i\right) - \ell(A(S), z_i)\right] \leq \epsilon(m)$

- Theorem:
  - $\mathbb{E}\left[L_{\mathcal{D}}\left(A(S)\right) - L_S\left(A(S)\right)\right] = \mathbb{E}\left[\ell\left(A\left(S^i\right), z_i\right) - \ell(A(S), z_i)\right]$

  - The generalization gap is bounded by the stability