Advanced Topics in ML

Soufiane Hayou

Saturday 13th May, 2023

- Name: Soufiane Hayou
- Email: hayou@nus.edu.sg
- Office: S17-05-08
- For any Lectures/Material related questions, please use the Discussion feature on Canvas. For other question, you can send me an email. In case I didn't respond after 2 days, send me again.
- For SSG funded students, attendance will be recorded through Zoom.

- Lectures will be recorded and posted on Canvas.
- Lecture slides and notebooks will also be on Canvas.

- Lectures will be recorded and posted on Canvas.
- Lecture slides and notebooks will also be on Canvas.

$\rightarrow$ DSA5202 is a 'natural' continuation of DSA5105/DSA5102!

- Lectures will be recorded and posted on Canvas.
- Lecture slides and notebooks will also be on Canvas.

→ DSA5202 is a 'natural' continuation of DSA5105/DSA5102!

→ DSA5202 is NOT a gentle introduction to ML (should be familiar with basics from DSA5105/DSA5102)

Teaching material

- Lecture slides and notebooks.
- Lecture notes of DSA5105 (will be uploaded on Canvas).

Teaching material

- Lecture slides and notebooks.
- Lecture notes of DSA5105 (will be uploaded on Canvas).

Further reading:

- "The Elements of Statistical Learning Data Mining, Inference, and Prediction", by Hastie, ,Tibshirani and Friedman.
- "Machine Learning, A Bayesian and Optimization Perspective", by Theodoridis
- "Pattern recognition and machine learning", by Bishop
- "Understanding machine learning, from theory to algorithms", by Shalev-Shwartz and Ben-David.

CA Components

- 3 Homeworks (online Quiz) (25%)
- Final project (25%)
- Final Test (Online live Quizz, week of June 12th) (50%)

In DSA5105, we learnt

- Different machine learning models
- How to use algorithms/packages to learn models
- How to validate the learning outcomes
- Use different techniques to improve results

Everything is working!

We will dig deeper in some topics

- How does optimization work?
- How can we avoid trainability issues in Deep Learning?
- How do we quantify the risk of a model?

Part I: Optimization Theory

- Understand Gradient Descent
- Variants of Gradient Descent

Part II: Deep Learning methods

- (Deep) Neural Networks
- Role of initialization, activation function etc

Part III: Quantifying uncertainty in ML

- Bayesian learning
- Monte Carlo sampling

Optimization (I)

Optimization task

Ideally, we want to find the model with the least expected prediction error

$$\min_{\mathbf{w}} f(\mathbf{w}), \quad f(\mathbf{w}) := \mathbb{E}_{\mathcal{D}}[\ell(Y, h_{\mathbf{w}}(X))],$$

where $\mathcal{D}$ is the data distribution.

Optimization task

Ideally, we want to find the model with the least expected prediction error

$$\min_{\mathbf{w}} f(\mathbf{w}), \quad f(\mathbf{w}) := \mathbb{E}_{\mathcal{D}}[\ell(Y, h_{\mathbf{w}}(X))],$$

where $\mathcal{D}$ is the data distribution.

$\rightarrow$ When fitting data, we consider empirical loss

$$\min_{\mathbf{w}} f(\mathbf{w}), \quad f(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h_{\mathbf{w}}(x_i))$$

Optimization task

Ideally, we want to find the model with the least expected prediction error

$$\min_{\mathbf{w}} f(\mathbf{w}), \quad f(\mathbf{w}) := \mathbb{E}_{\mathcal{D}}[\ell(Y, h_{\mathbf{w}}(X))],$$

where $\mathcal{D}$ is the data distribution.

$\rightarrow$ When fitting data, we consider empirical loss

$$\min_{\mathbf{w}} f(\mathbf{w}), \quad f(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h_{\mathbf{w}}(x_i))$$

$\rightarrow$ One general idea: find $\mathbf{w}$ so that $\nabla f(\mathbf{w}) = 0$.

Reference for today's lecture: "Convex Optimization", Boyd and Vandenberghe.

Stationary points and Lipschitz gradients

Definition

We say $\mathbf{w}$ is a stationary point of f if $\nabla f(\mathbf{w}) = 0$. We say it is an ϵ-stationary point if $\|\nabla f(\mathbf{w})\| \leq \epsilon$

Definition

We say ∇f is L-Lipschitz if

$$\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{w} - \mathbf{y}\|.$$

Convexity and Optimum

- A convex set D is a set that satisfies: if $x, y \in D$, then for all $t \in [0,1], tx + (1-t)y \in D$.
- A function f is convex, if for any $t \in [0,1]$,

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

Differentiable convex function

- A convex set D is a set that satisfies: if $x, y \in D$, then for all $t \in [0,1], tx + (1-t)y \in D$.
- A function f is convex, if for any $t \in [0,1]$,

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

- If f is $\mathcal{C}^1$ (differentiable with continuous derivative), then it is convex if and only if

$$f(x) + \nabla f(x)^T (y - x) \leq f(y).$$

Differentiable convex function

- A convex set D is a set that satisfies: if $x, y \in D$, then for all $t \in [0,1], tx + (1-t)y \in D$.
- A function f is convex, if for any $t \in [0,1]$,

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

- If f is $\mathcal{C}^1$ (differentiable with continuous derivative), then it is convex if and only if

$$f(x) + \nabla f(x)^T(y-x) \leq f(y).$$

- Consequence: if $\nabla f(x) = 0$ then x is a minimizer of f.

- A $\mathcal{C}^1$ function f is c-strongly convex if for all x, y

$$f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} c \|y - x\|^2 \leq f(y)$$

Differentiable convex function

- A $\mathcal{C}^1$ function f is c-strongly convex if for all x, y

$$f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} c \|y - x\|^2 \leq f(y)$$

- Equivalently

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq c \|y - x\|^2.$$

Differentiable convex function

- A $\mathcal{C}^1$ function f is c-strongly convex if for all x, y

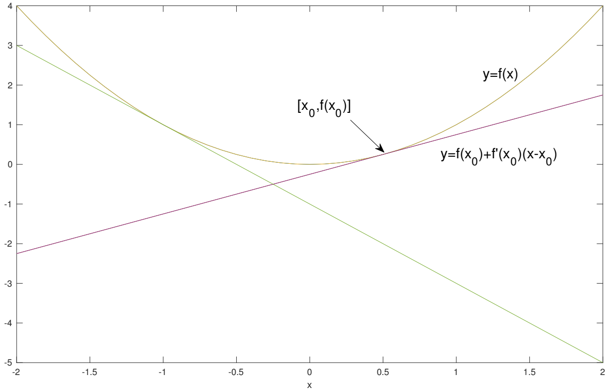
$$f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} c \|y - x\|^2 \leq f(y)$$

- Equivalently

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq c\|y - x\|^2.$$

- Consequence: if $\nabla f(x) = 0$ then x is the **unique** minimizer of f.

The graph of a convex function is above all tangent lines.



Definition

Let $\mathbf{A}$ be a real $n \times n$ symmetric matrix. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be its eigenvalues.

(a) $\mathbf{A}$ is said to be **positive semidefinite** (PSD) if $x^T \mathbf{A} x \geq 0, \ \forall \ x \in \mathcal{R}^n$. This is equivalent to $\lambda_n \geq 0$.

(b) $\mathbf{A}$ is said to be **positive definite** (PD) if $x^T \mathbf{A} x > 0, \ \forall \ x \neq 0$. This is equivalent to $\lambda_n > 0$.

(c) we write $\mathbf{A} \succeq cI$ if $x^T \mathbf{A} x > c\|x\|^2, \ \forall \ x \in \mathcal{R}^n \backslash \{0\}$. This is equivalent to $\lambda_n \geq c$.

Hessian and Convexity

Theorem

Suppose that $f(x)$ *is* $\mathcal{C}^2$ *on an* **open** *convex set* D *in* $\mathcal{R}^n$.

- *The function* f *is convex on* D **if and only if** *the Hessian matrix* $H_f(x)$ *is PSD at each* $x \in D$.
- *If the Hessian matrix* $H_f(x) \succeq cI$ *at each* $x \in D$, *then* f *is* c**-strongly convex** *on* D.

Example

Consider the following functions. Are they convex or strongly convex?

- $f(w) = e^w$

Example

Consider the following functions. Are they convex or strongly convex?

- $f(w) = e^w$
- $f(\mathbf{w}) = \|\mathbf{w}\|^2$

Example

Consider the following functions. Are they convex or strongly convex?

- $f(w) = e^w$
- $f(\mathbf{w}) = \|\mathbf{w}\|^2$
- $f(\mathbf{w}) = \|A\mathbf{w} - b\|^2$

Iterative algorithms

Question: find stationary point $\mathbf{w}^*$: $\nabla f(\mathbf{w}^*) = \mathbf{0}$.

Question: find stationary point $\mathbf{w}^*$: $\nabla f(\mathbf{w}^*) = \mathbf{0}$.

$\rightarrow$ generate iterates

$$\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n \to \mathbf{w}^*.$$

Update rules:

How to get $\mathbf{w}_{k+1}$ from $\mathbf{w}_k$?

Update rules:

$$\text{How to get } \mathbf{w}_{k+1} \text{ from } \mathbf{w}_k?$$

When do we stop the iterations?

Design question

Update rules:

$$\text{How to get } \mathbf{w}_{k+1} \text{ from } \mathbf{w}_k?$$

When do we stop the iterations?

- Budget: stop at $k = T$, T is given.

Update rules:

How to get $\mathbf{w}_{k+1}$ from $\mathbf{w}_k$?

When do we stop the iterations?

- Budget: stop at $k = T$, T is given.
- Tolerance: stop at $\|\nabla f(\mathbf{w}_k)\| \leq$Tol. Tol is a small precision requirement.

Update rules:

How to get $\mathbf{w}_{k+1}$ from $\mathbf{w}_k$?

When do we stop the iterations?

- Budget: stop at $k = T$, T is given.
- Tolerance: stop at $\|\nabla f(\mathbf{w}_k)\| \leq$Tol. Tol is a small precision requirement.
- Improvement: stop if $\|\mathbf{w}_k - \mathbf{w}_{k+1}\| \leq$ Tol.

- Complexity: how many iterations for the algorithm to produce $\mathbf{w}_k$ so that $\|\mathbf{w}_k - \mathbf{w}^*\| \leq \epsilon$?

Assessment of algorithms

- Complexity: how many iterations for the algorithm to produce $\mathbf{w}_k$ so that $\|\mathbf{w}_k - \mathbf{w}^*\| \leq \epsilon$?
- Implementation: is the algorithm easy to implement? What ingredients do we need?

- Complexity: how many iterations for the algorithm to produce $\mathbf{w}_k$ so that $\|\mathbf{w}_k - \mathbf{w}^*\| \leq \epsilon$?
- Implementation: is the algorithm easy to implement? What ingredients do we need?
- Update cost: what is the cost of running one step of the algorithm?

- Complexity: how many iterations for the algorithm to produce $\mathbf{w}_k$ so that $\|\mathbf{w}_k - \mathbf{w}^*\| \leq \epsilon$?
- Implementation: is the algorithm easy to implement? What ingredients do we need?
- Update cost: what is the cost of running one step of the algorithm?
- Scalability: how does the computational and storage cost scale with the problem size?

Assessment of algorithms

- Complexity: how many iterations for the algorithm to produce $\mathbf{w}_k$ so that $\|\mathbf{w}_k - \mathbf{w}^*\| \leq \epsilon$?
- Implementation: is the algorithm easy to implement? What ingredients do we need?
- Update cost: what is the cost of running one step of the algorithm?
- Scalability: how does the computational and storage cost scale with the problem size?
- Parallelization: if I have multiple CPUs/GPUs, can I run the algorithm faster?

Newton's method

Try to solve

$$\nabla f(\mathbf{w}) = \mathbf{0}.$$

Try to solve

$$\nabla f(\mathbf{w}) = \mathbf{0}.$$

- Suppose ∇f and H_f are both available.

Try to solve

$$\nabla f(\mathbf{w}) = \mathbf{0}.$$

- Suppose ∇f and H_f are both available.
- Generate a sequence $\mathbf{w}_1, \ldots, \mathbf{w}_k \to \mathbf{w}^*$.

Try to solve

$$\nabla f(\mathbf{w}) = \mathbf{0}.$$

- Suppose ∇f and H_f are both available.
- Generate a sequence $\mathbf{w}_1, \ldots, \mathbf{w}_k \to \mathbf{w}^*$.
- 1st order expansion $\nabla f(\mathbf{w}) \approx \nabla f(\mathbf{w}_k) + H_f(\mathbf{w}_k)(\mathbf{w} - \mathbf{w}_k)$

Try to solve

$$\nabla f(\mathbf{w}) = \mathbf{0}.$$

- Suppose ∇f and H_f are both available.
- Generate a sequence $\mathbf{w}_1, \ldots, \mathbf{w}_k \to \mathbf{w}^*$.
- 1st order expansion $\nabla f(\mathbf{w}) \approx \nabla f(\mathbf{w}_k) + H_f(\mathbf{w}_k)(\mathbf{w} - \mathbf{w}_k)$
- want $\nabla f(\mathbf{w}_{k+1}) = \mathbf{0} \Rightarrow \mathbf{w}_{k+1} - \mathbf{w}_k \approx -[H_f(\mathbf{w}_k)]^{-1} \nabla f(\mathbf{w}_k)$

Try to solve

$$\nabla f(\mathbf{w}) = \mathbf{0}.$$

- Suppose ∇f and H_f are both available.
- Generate a sequence $\mathbf{w}_1, \ldots, \mathbf{w}_k \to \mathbf{w}^*$.
- 1st order expansion $\nabla f(\mathbf{w}) \approx \nabla f(\mathbf{w}_k) + H_f(\mathbf{w}_k)(\mathbf{w} - \mathbf{w}_k)$
- want $\nabla f(\mathbf{w}_{k+1}) = \mathbf{0} \Rightarrow \mathbf{w}_{k+1} - \mathbf{w}_k \approx -[H_f(\mathbf{w}_k)]^{-1} \nabla f(\mathbf{w}_k)$
- Iterate: $\mathbf{w}_{k+1} = \mathbf{w}_k - [H_f(\mathbf{w}_k)]^{-1} \nabla f(\mathbf{w}_k)$.

Pseudo code

Algorithm 1: Newton's method

Input: $\mathbf{w}_0, \nabla f, [H_f]^{-1}$, Tol

Output: $\mathbf{w}_T$ so $\|\nabla f(\mathbf{w}_T)\| \leq$ Tol

$k = 0$;

while $\|\nabla f(\mathbf{w}_k)\| \geq$ *Tol* **do**

  $\mathbf{w}_{k+1} = \mathbf{w}_k - [H_f(\mathbf{w}_k)]^{-1}\nabla f(\mathbf{w}_k)$;

  $k = k + 1$;

Theorem

Suppose $\|\mathbf{w}_0 - \mathbf{w}^*\|$ is sufficiently small, $H_f(\mathbf{w})^{-1}$ and ∇f are L-Lipschitz, then for a constant M

$$\|\mathbf{w}_{k+1} - \mathbf{w}^*\| \leq M\|\mathbf{w}_k - \mathbf{w}^*\|^2, \quad \forall k.$$

Quadratic convergence: doubling the digit of accuracy every iteration.

Note that for some $\mathbf{y}_k$ between $\mathbf{w}^*$ and $\mathbf{w}_k$,

$$
\begin{aligned}
0 = \nabla f(\mathbf{w}_*) &= \nabla f(\mathbf{w}_k) + H_f(\mathbf{y}_k)(\mathbf{w}_* - \mathbf{w}_k) \\
\mathbf{w}^* =& \mathbf{w}_k - [H_f(\mathbf{y}_k)]^{-1} \nabla f(\mathbf{w}_k) \\
\mathbf{w}_{k+1} &= \mathbf{w}_k - [H_f(\mathbf{w}_k)]^{-1} \nabla f(\mathbf{w}_k) \\
\|\mathbf{w}_* - \mathbf{w}_{k+1}\| &= \|[H_f(\mathbf{y}_k)]^{-1} - [H_f(\mathbf{w}_k)]^{-1}\| \nabla f(\mathbf{w}_k) \\
&\leq L\|\mathbf{y}_k - \mathbf{w}_k\| L \|\mathbf{w}^* - \mathbf{w}_k\| \\
&\leq L^2 \|\mathbf{w}^* - \mathbf{w}_k\|^2
\end{aligned}
$$

1. Choose $\mathbf{w}_1$

Multivariate Newton's method

1. Choose $\mathbf{w}_1$
2. Let $\mathbf{w}_{k+1} = \mathbf{w}_k - [H_f(\mathbf{w}_k)]^{-1} \nabla f(\mathbf{w}_k)$

Multivariate Newton's method

1. Choose $\mathbf{w}_1$
2. Let $\mathbf{w}_{k+1} = \mathbf{w}_k - [H_f(\mathbf{w}_k)]^{-1} \nabla f(\mathbf{w}_k)$
3. Repeat step 2 until $\|\nabla f(\mathbf{w}_k)\| < Tol$.

Multivariate Newton's method

1. Choose $\mathbf{w}_1$
2. Let $\mathbf{w}_{k+1} = \mathbf{w}_k - [H_f(\mathbf{w}_k)]^{-1}\nabla f(\mathbf{w}_k)$
3. Repeat step 2 until $\|\nabla f(\mathbf{w}_k)\| < Tol$.

- A sequence of points
- Quadratic convergence to $\mathbf{w}^*$
- In practice, Hessian can be hard to obtain.
- A numerical approximation version called BFGS is often used.

Example

Example

Implement the Newton's method for

$$\min f(x, y) = x^4 + y^2.$$

Use $[x_0; y_0] = [1, 0]$. Find $[x_1, y_1]$. How about $[x_2, y_2]$?

(2/3,0), (4/9,0)

Why do we want to learn the convergence theory of algorithms?
Why do we want to manually find the iterations?

Gradient Descent

Gradient descent

- Suppose we let $\mathbf{w}_{k+1} = \mathbf{w}_k + h\mathbf{v}_k$ with a small h.

- Suppose we let $\mathbf{w}_{k+1} = \mathbf{w}_k + h\mathbf{v}_k$ with a small h.
- What is the choice of directional $\mathbf{v}_k$ so $f(\mathbf{w}_{k+1})$ is minimized?

- Suppose we let $\mathbf{w}_{k+1} = \mathbf{w}_k + h\mathbf{v}_k$ with a small h.
- What is the choice of directional $\mathbf{v}_k$ so $f(\mathbf{w}_{k+1})$ is minimized?
- Use 1st order Taylor $f(\mathbf{w}_{k+1}) \approx f(\mathbf{w}_k) + h\langle \mathbf{v}_k, \nabla f(\mathbf{w}_k)\rangle$

Gradient descent

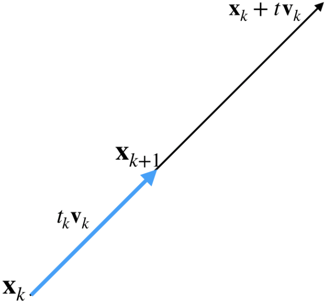
- Suppose we let $\mathbf{w}_{k+1} = \mathbf{w}_k + h\mathbf{v}_k$ with a small h.
- What is the choice of directional $\mathbf{v}_k$ so $f(\mathbf{w}_{k+1})$ is minimized?
- Use 1st order Taylor $f(\mathbf{w}_{k+1}) \approx f(\mathbf{w}_k) + h\langle \mathbf{v}_k, \nabla f(\mathbf{w}_k)\rangle$
- Unormalized $\mathbf{v}_k = -\nabla f(\mathbf{w}_k)$ is the **steepest descent direction**.

Multivariate:

$$\min f(\mathbf{w}), \quad \mathbf{w} \in \mathcal{R}^n$$

Univariate:

$$\min f(\mathbf{w}_k + h\mathbf{v}_k), \quad h \geq 0.$$



- We will let $\mathbf{w}_{k+1} = \mathbf{w}_k + h\mathbf{v}_k$.
- Turn Multivariate into Univariate.

Algorithm 2: Gradient Descent

Input: $\mathbf{w}_0, \nabla f, (t_k)$, Tol

Output: $\mathbf{w}_k$ so $\|\nabla f(\mathbf{w}_t)\| \leq$ Tol

```
k = 0;
while ||∇f(w_k)|| ≤ Tol do
    v_k = −∇f(w_k);
    (h_k = arg min_t f(w_k + h v_k));
    w_{k+1} = w_k + h_k v_k;
    k = k + 1;
```

$k = 0$;

while $\|\nabla f(\mathbf{w}_k)\| \leq$ *Tol* **do**

$\quad \mathbf{v}_k = -\nabla f(\mathbf{w}_k)$;

$\quad (h_k = \arg\min_t f(\mathbf{w}_k + h\mathbf{v}_k))$;

$\quad \mathbf{w}_{k+1} = \mathbf{w}_k + h_k\mathbf{v}_k$;

$\quad k = k + 1$;

How to find h_k for

$$\mathbf{w}_{k+1} = \mathbf{w}_k + h_k \mathbf{v}_k?$$

How to find h_k for

$$\mathbf{w}_{k+1} = \mathbf{w}_k + h_k \mathbf{v}_k?$$

Several choices

- Minimization rule: $\min f(\mathbf{w}_k + h_k \mathbf{v}_k)$, $h_k \in [0, \bar{t}]$

How to find h_k for

$$\mathbf{w}_{k+1} = \mathbf{w}_k + h_k \mathbf{v}_k?$$

Several choices

- Minimization rule: $\min f(\mathbf{w}_k + h_k \mathbf{v}_k)$, $h_k \in [0, \bar{t}]$
- (Exponential) Decreasing schedule: $h_k = \beta^k$ where $\beta \in (0, 1)$ is some tuning parameter.

How to find h_k for

$$\mathbf{w}_{k+1} = \mathbf{w}_k + h_k \mathbf{v}_k?$$

Several choices

- Minimization rule: $\min f(\mathbf{w}_k + h_k \mathbf{v}_k)$, $h_k \in [0, \bar{t}]$
- (Exponential) Decreasing schedule: $h_k = \beta^k$ where $\beta \in (0, 1)$ is some tuning parameter.
- Using some fixed h (most popular).

Convergence of gradient descent

Convergence of Gradient descent

Proposition

Suppose $f(\mathbf{w}) \geq 0$ and ∇f is L-Lipschitz. Then gradient descent with fixed step size $h \leq \frac{1}{L}$ satisfies

$$f(\mathbf{w}_n) - f(\mathbf{w}^*) \leq \frac{\|\mathbf{w}_0 - \mathbf{w}^*\|^2}{2hn}.$$

Theorem

Consider applying gradient descent with fixed step size

$$\mathbf{w}_{k+1} = \mathbf{w}_k - h\nabla f(\mathbf{w}_k)$$

If f is c-strongly convex, ∇f is L-Lipschitz and $h \leq \frac{c}{L^2}$

$$f(\mathbf{w}_n) - f(\mathbf{w}^*) \leq L(1-ch)^n \|\mathbf{w}_0 - \mathbf{w}^*\|^2$$

Remark: the condition can be improved to $h \leq O(c/L)$.

$$
\begin{aligned}
\|w_{k+1} - w^*\|^2 &= \|w_k - w^* - h\nabla f(w_k)\|^2 \\
&\leq \|w_k - w^*\|^2 - 2hc\|w_k - w^*\|^2 \\
&\quad + h^2L^2\|w_k - w^*\|^2 \\
&\leq (1 - ch)\|w_k - w^*\|^2.
\end{aligned}
$$

By induction we can show our claim.

Code: `https://github.com/lilipads/gradient_descent_viz`