9/9/2025

Reminders

↳ Quiz

↳ Pset +Self Grade!

↳ Office Hours

↳ Discord

↳ Events

- Conference Oct 3-5
- Redistricting Sep 16 Lunch @ Ath
- Trees Sep 16 7pm @ Mudd

Review

$\underline{x}^{(i)} \in \mathbb{R}^d$ $\quad y^{(i)} \in \mathbb{R}$

$\underline{X} \in \mathbb{R}^{n \times d}$ $\quad \underline{y} \in \mathbb{R}^n$

$$\underline{X} = \begin{bmatrix} \\ -\underline{x}^{(i)T}- \\ \\ \end{bmatrix} \qquad \underline{y} = \begin{bmatrix} \\ y^{(i)} \\ \\ \end{bmatrix}$$

Goal: $f(x^{(i)}) \approx y^{(i)}$

① Model — Linear! $f(x) = w^T x$

② Loss — MSE! $\mathcal{L}(w) = \sum_{i=1}^{n} \left[f(x^{(i)}) - y^{(i)} \right]^2 \frac{1}{n}$

③ Optimizer — Exact! $w^* = (X^T X)^+ X^T y$

Why pseudo inverse?

"Invert" non-invertible matrices

e.g. $\underline{X} \in \mathbb{R}^{n \times d}$

$$\underline{X} = \sum_{i=1}^{d} \sigma_i \, \underline{u}_i \, \underline{v}_i^T$$

$$\underline{X}^+ \underline{X} = \sum_{j=1}^{d} \frac{1}{\sigma_j} \underline{v}_j \, \underline{u}_j^T \sum_{i=1}^{d} \sigma_i \, \underline{u}_i \, \underline{v}_i^T$$

$$= \sum_{i=1}^{d} \underline{v}_i \, \underline{v}_i^T = I_d$$

Greatest advice: "Go sit on your rock"

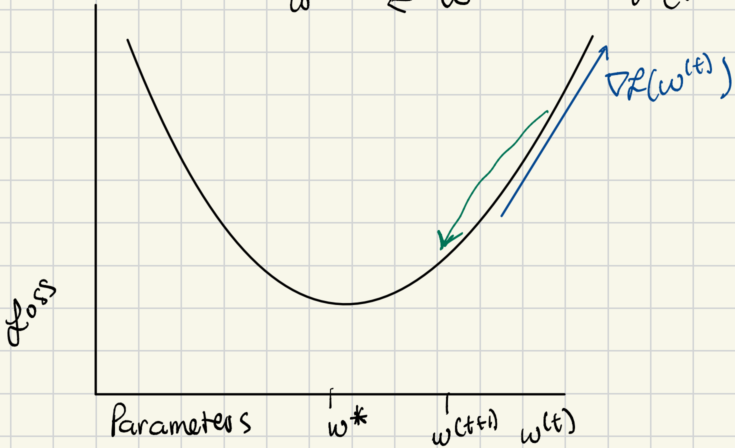
Two issues:

↳ Time to compute w^*

↳ What if data is not linear?

Gradient Descent

$$w^{(t+1)} \leftarrow w^{(t)} - \alpha \nabla \mathcal{L}(w^{(t)})$$



Intuition : Move away from steepest ascent

α = "step size" or "learning rate"

The Math (as promised)

$\boxed{w \in \mathbb{R}}$

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \lim_{\Delta \to 0} \frac{\mathcal{L}(w+\Delta) - \mathcal{L}(w)}{\Delta}$$

$$\Rightarrow \quad \mathcal{L}(w+\Delta) - \mathcal{L}(w) \approx \frac{\partial \mathcal{L}(w)}{\partial w} \Delta$$

Choose Δ so $\mathcal{L}(w+\Delta) - \mathcal{L}(w)$ is negative ....

$$\Delta = - \frac{\partial \mathcal{L}(w)}{\partial w}$$

$$\boxed{w \in \mathbb{R}^d}$$

$$\frac{\partial \mathcal{L}(w)}{\partial w_i} = \lim_{\Delta \to 0} \frac{\mathcal{L}(w + \Delta e_i) - \mathcal{L}(w)}{\Delta}$$

$$\Rightarrow \mathcal{L}(w + \Delta e_i) - \mathcal{L}(w) \approx \Delta \langle \nabla_w \mathcal{L}(w), e_i \rangle$$

$$\mathcal{L}(w + v) - \mathcal{L}(w) \approx \Delta \langle \nabla_w \mathcal{L}(w), v \rangle$$

Choose $\Delta v = -\alpha \nabla_w \mathcal{L}(w)$

$$\mathcal{L}(w + v) - \mathcal{L}(w) \approx -\alpha \, \|\nabla_w \mathcal{L}(w)\|_2^2$$
$$= -\alpha \, \|\nabla_w \mathcal{L}(w)\|_2 \|\nabla_w \mathcal{L}(w)\|_2$$

Recall $\langle a, b \rangle = \|a\|_2 \|b\|_2 \cos(\theta)$

$\max_\theta \cos \theta = 1 \Rightarrow$ achieve "best" update!

For linear regression,

$$\nabla_w \mathcal{L}(w) = \frac{2}{n} X^T (Xw - y)$$

Time to compute:

$\boxed{9/11/25}$

Reminders

↳ Notes!

↳ Self-grade due Friday

↳ Pset 3 due Monday

↳ Quiz Tuesday

↳ exercises + notes = fair game

↳ Events!

↳ Please ask specific questions, e.g.,

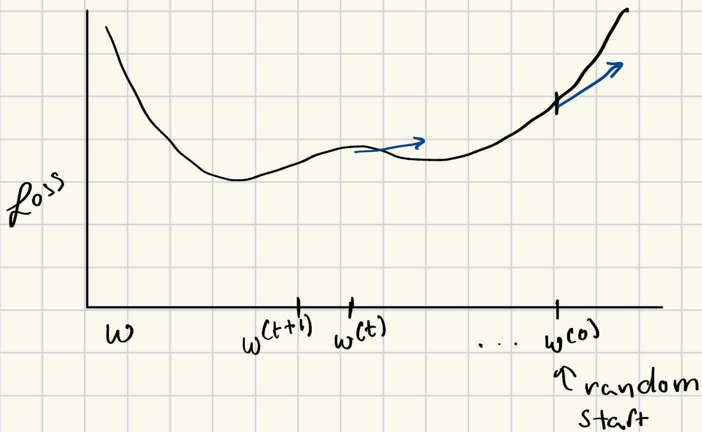
"Explain concept A"

"Why is B true"

↳ $\boxed{\text{Sit on your rock}}$

most important thing I can offer

Review: Gradient Desent



$$w^{(t+1)} \leftarrow w^{(t)} - \alpha \nabla \mathcal{L}(w^{(t)})$$

Linear regression:

$$\nabla \mathcal{L}(w) = \underset{d \times n}{X^T}(\underset{n \times d}{X}\underset{d \times 1}{w} - \underset{n \times 1}{y})$$

Time:

Even Faster

n and d can be prohibitively large

Enter: STOCHASTIC gradient descent

↑
"stokhos"

$$S \subseteq \{1, \dots, n\} \quad \text{s.t.} \quad |S| = m$$

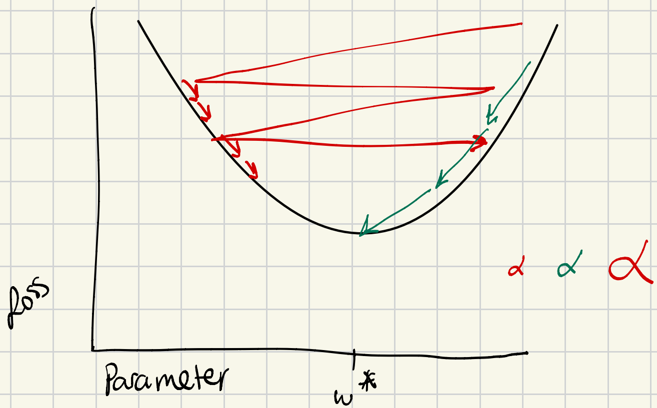
$$\mathcal{L}_S(w) = \frac{1}{m} \sum_{i \in S} [f(x^{(i)}) - y^{(i)}]^2$$

For linear regression,

$$\nabla_w \mathcal{L}_S(w) = \frac{2}{|S|} X_S^T (X_S w - y_S)$$

Time:

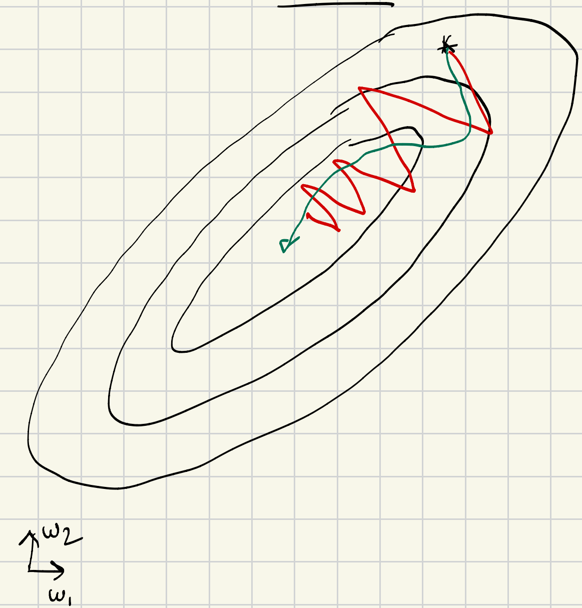
Learning Rate



heuristic strategies:

- If loss unstable, decrease $(\alpha/2)$
  If loss slowly decreasing, increase (2α)
- "scheduler" start big → small, cycle
- Adaptive $\alpha^{(t+1)} \leftarrow \dfrac{\alpha^{(t)}}{\left(\nabla_w \mathcal{L}(w^{(t)})\right)^2}$

Momentum



$$m^{(t+1)} \leftarrow \beta\, m^{(t)} + (1-\beta)\, \nabla_w \mathcal{L}_S(w^{(t)})$$

$$w^{(t+1)} \leftarrow w^{(t)} - \alpha\, m^{(t+1)}$$

Polynomial Regression

Q: We covered how to speed up optimization, how do we address model mis-specification?

A: Add more *expressive* features

Linear regression e.g.,

$$f(x) = w_1 x_1 + w_2 x_2$$

$$w^* = X^+ y \quad = (X^T X)^{-1} X^T y$$

Polynomial regression e.g.,

$$f(x) = w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$$

$$X_{aug} = [\, X \quad Z \,]$$

$$\underbrace{\begin{bmatrix} \; \\ \; \\ \; \end{bmatrix}}_{d}\underbrace{\begin{bmatrix} \; \\ \; \\ \; \end{bmatrix}}_{d'}$$

$$w^*_{aug} = X_{aug}^+ \, y$$

Q: What changes in 3 step recipe?

Expressivity

$w^* = X^+ y$ $\quad w^*_{aug} = X^+_{aug} y$

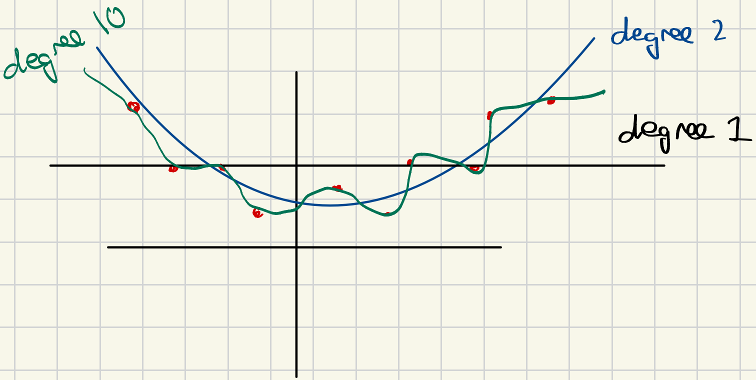
$\boxed{\text{Claim: More expressive}}$

$$\min_{w \in \mathbb{R}^{d+d'}} \| X_{aug} w - y \|_2^2 \le \min_{w \in \mathbb{R}^d} \| X w - y \|_2^2$$

Proof: choose $\hat{w} = \begin{bmatrix} w^* \\ 0 \\ \vdots \\ 0 \end{bmatrix}$. Then $\| X_{aug} \hat{w} - y \|_2^2 = \| X w^* - y \|_2^2$

Since $\hat{w}$ is always a choice, w^*_{aug} must be at least as good

Which degree to choose?



overfits (memorize)

x^1, x^2, x^{10}

not expressive enough

just right :)

Generalization Error

Training	Validation	Test

Data

repeat:

1. Train model
2. Check loss on validation set
3. Update hyperparameters (learning rate, momentum, etc)

Pernicious issue: model performs better on validation even though never "trained" on it

Solution: Use test loss <u>once</u> on final model

Regularization

Idea: Use simplest model

Weight magnitude $\approx$ complexity

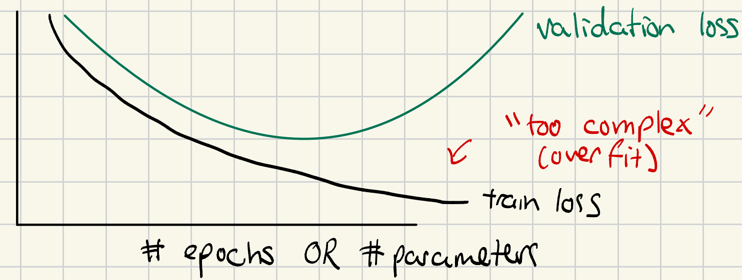
↳ More output variation per input change

New loss:

$$\mathcal{L} + \lambda \|w\|_2^2$$

Double Descent

Traditional View:



Modern View: