Tuesday, February 10

- Thank you for coming to GEMS/my talk!
- CCMS Applied Math seminar Mondays: 4:15 pm Estella 1021
- CCMS Colloquium Fridays: 11am Davidson Lecture Hall
- No OH Thursday, DM to meet Friday!

Plan

- Code demo
- Empirical Risk Minimization
- Gradient Descent

Linear Regression

$$\mathcal{L}(w) = \|Xw - y\|_2^2$$

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

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

$$\begin{bmatrix} \vdots \\ — x^{(i)T} — \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} \vdots \\ y^{(i)} \\ \vdots \end{bmatrix}$$

Time to compute?

$$\operatorname*{argmin}_w \|Xw - y\|_2^2 = (X^TX)^{-1}X^Ty$$

Question: How do we deal with intercept?

Empirical Risk Minimization

Answer to why squared-error

Suppose we believe

$$y^{(i)} = \langle w^*, x^{(i)} \rangle + \eta^{(i)}$$

same as $w^{*T} x^{(i)}$ but pretty :)

for noise $\eta \sim \mathcal{N}(0, \sigma^2)$

↑ normal dist

Then $y^{(i)} \sim \mathcal{N}(\langle w^*, x^{(i)} \rangle, \sigma^2)$

$$Pr(y^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y^{(i)} - \langle w, x^{(i)} \rangle)^2}{2\sigma^2} \right)$$

Idea: Find w that maximizes likelihood of observing data

$$\operatorname*{argmax}_w Pr(y^{(1)}, \dots, y^{(n)})$$

$$= \operatorname*{argmax}_w \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(- \frac{(y^{(i)} - \langle w, x^{(i)} \rangle)^2}{2\sigma^2} \right)$$

$$= \operatorname*{argmax}_w \exp\left(\sum_{i=1}^{n} - \frac{(y^{(i)} - \langle w, x^{(i)} \rangle)^2}{2\sigma^2} \right)$$

$$= \operatorname*{argmin}_w - \log\left(\exp\left(\sum_{i=1}^{n} - \frac{(y^{(i)} - \langle w, x^{(i)} \rangle)^2}{2\sigma^2} \right) \right)$$

$$= \operatorname*{argmin}_w - \sum_{i=1}^{n} - \frac{(y^{(i)} - \langle w, x^{(i)} \rangle)^2}{2\sigma^2}$$

$$= \operatorname*{argmin}_w \sum_{i=1}^{n} (y^{(i)} - \langle w, x^{(i)} \rangle)^2$$

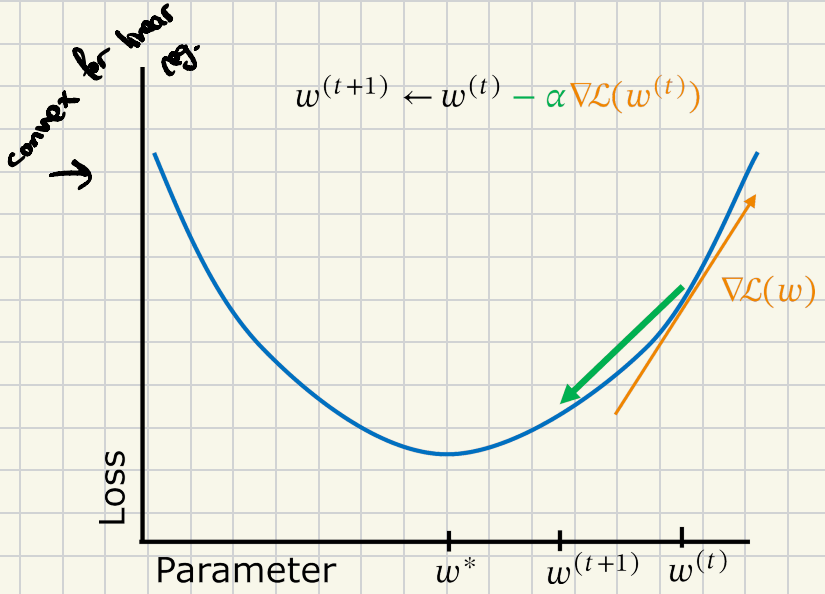
Thursday, February 12

Plan

Gradient Descent

Gradient Descent

Computing w^* for linear regression takes $O(nd^2 + d^3)$ time, can we improve?



Initialize $w^{(0)}$ randomly

Choose learning rate α

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

Time to take T steps for linear regression?

Why Gradient descent?

1 Dimension $w \in \mathbb{R}$

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

For small Δ,

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

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

We want $\mathcal{L}(w+\Delta) - \mathcal{L}(w) < 0$

$\Rightarrow$ Choose $\Delta = -\alpha \frac{\partial \mathcal{L}(w)}{\partial w}$ so

$$\mathcal{L}(w+\Delta) - \mathcal{L}(w) \approx \frac{\partial \mathcal{L}(w)}{\partial w} \cdot -\frac{\partial \mathcal{L}(w)}{\partial w}\alpha < 0$$

$$w^{(t+1)} \leftarrow w^{(t)} + \Delta = w^{(t)} - \alpha \frac{\partial \mathcal{L}}{\partial w}(w^{(t)})$$

Higher Dimensions $w \in \mathbb{R}^d$

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

$$\mathcal{L}(w+\Delta e_i) - \mathcal{L}(w) \approx \frac{\partial \mathcal{L}(w)}{\partial w_i}\Delta$$

For $v \in \mathbb{R}^d$

$$\mathcal{L}(w+\Delta v) - \mathcal{L}(w) \approx \sum_{i=1}^{d} \text{change along component } i$$

$$\approx \sum_{i=1}^{d} \frac{\partial \mathcal{L}(w)}{\partial w_i}\Delta v_i = \Delta \langle \nabla \mathcal{L}_w(w), v \rangle$$

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

Choose $v = -\nabla_w \mathcal{L}(w)$ to align direction

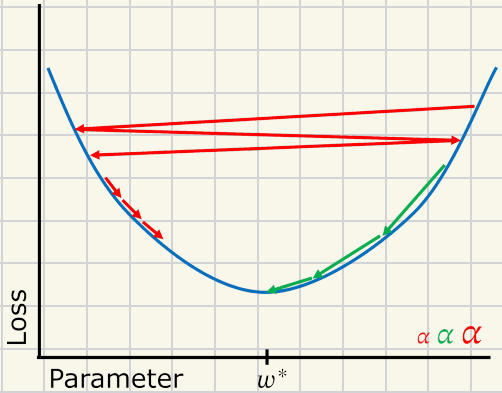
Stochastic Gradient Descent

Choose batch $S \subseteq \{1, \dots, n\} = [n]$

$$\mathcal{L}_S(w) = \frac{1}{|S|} \sum_{i \in S} \left(f(x^{(i)}) - y^{(i)} \right)^2$$

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

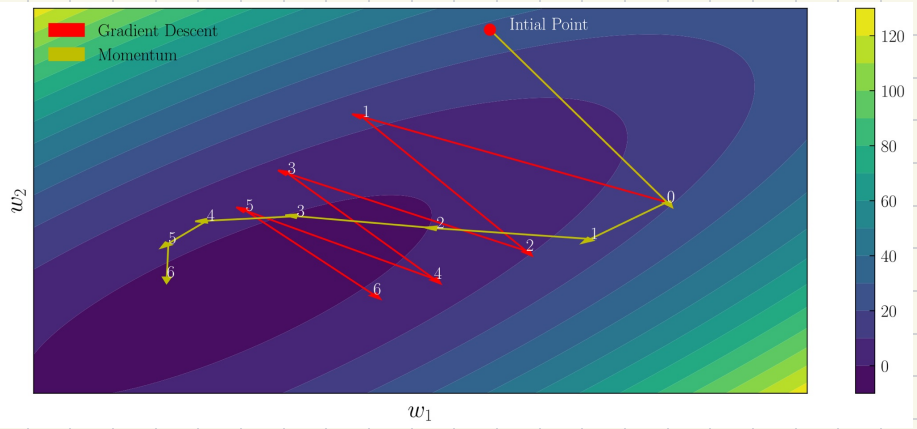
Step Sizes



Choosing α ...

- **Manual:**
  If loss decreases, try $\uparrow \alpha$
  If loss jumps around, try $\downarrow \alpha$
  } by factors of 1/2
- **Decay:**
  Decrease as we "get closer"
- **Adaptive:**
  Choose proportional to inverse grad sum
  ↳ ↑ grad, risk of overshooting
  ↳ use sum of past for stability

Momentum



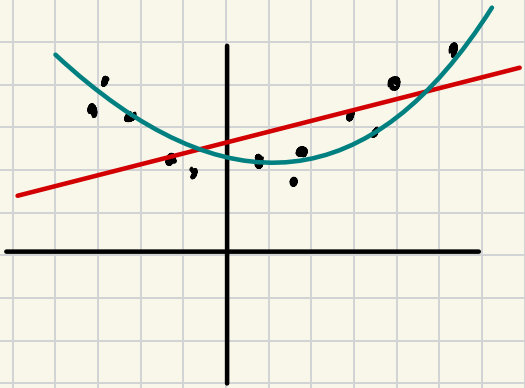
$$v^{(t+1)} \leftarrow \beta v^{(t)} + \nabla_w \mathcal{L}_S(w^{(t)})$$

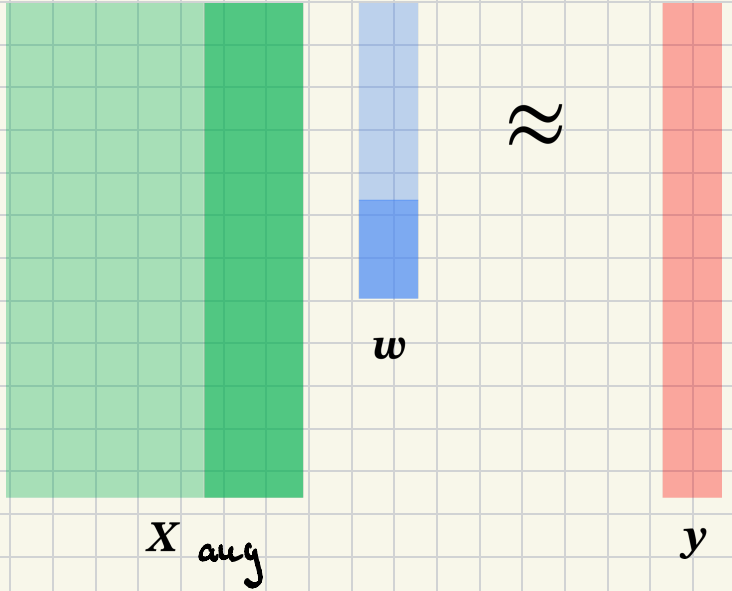
$$w^{(t+1)} \leftarrow w^{(t)} - \alpha v^{(t+1)}$$

$\beta \in [0, 1)$ controls "friction"

Polynomial Regression

Can we do better than a linear fit?





$$f(x) = w_0 + w_1 x_1 + \dots + w_d x_d \quad \text{(linear)}$$

$$f(x) = w_0 + w_1 x_1 + w_2 x_1 x_2 + w_3 x_2^2 \quad \text{(polynomial)}$$

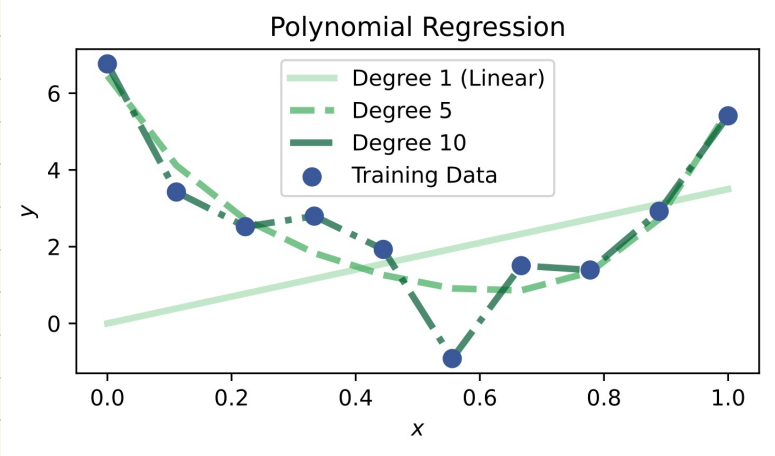
Claim:

$$\min_{w \in \mathbb{R}^{d_{aug}}} \|y - X_{aug} w\|_2^2 \leq \min_{w \in \mathbb{R}^{d}} \|y - Xw\|_2^2$$

Downsides of more features?

1. Time complexity
   (mitigated by kernel methods)

2. Overfitting



Generalization Error

Train ↑ Train model

Validation ↑ Tune hyperparams

Test ↑ use once

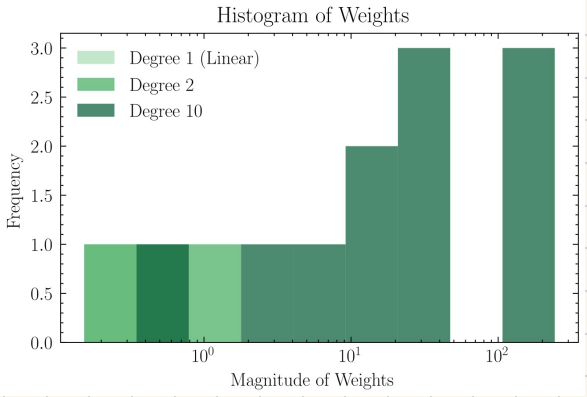
Overfit with:

1. Too many parameters
2. Too much training

Regularization

Occam's razor: Simplest explanation is usually right

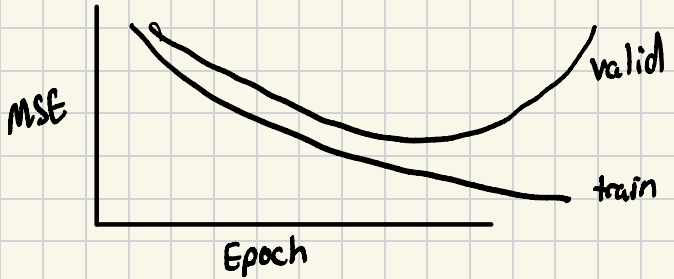
Complicated ≈ big changes
≈ big weights



$$\underset{w}{\operatorname{argmin}} \quad \mathcal{L}(w) + \lambda \|w\|_2^2$$

Traditional View

Tradeoff between generalization and training



Double Descent