Optimization

$$\nabla L(\hat{y}, y)$$

- Timeline
- Optimization
  - Momentum
  - Adaptive Learning Rates (Adagrad, RMSProp, Adam)

Timeline

Foundations
- Lin Alg
- Coding
- CLI

Basics
1. Introductory
3. Ethics
   - Competing technologies
4. Single
5. NN

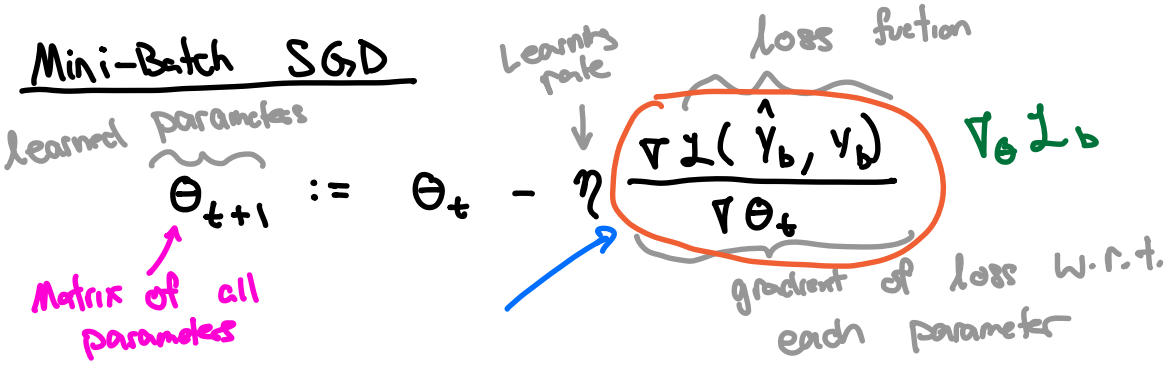
Intermediate
6. MB SGD
   - Optimization
   - Regularization
   - CNN
   - RNN

Advanced
- GANs
- Hyper tuning
- Attention

2. Deployed Apps

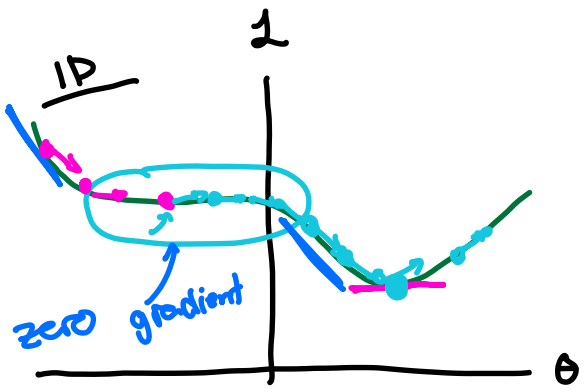
Don't wait until we cover something in class to use it.

Mini-Batch SGD

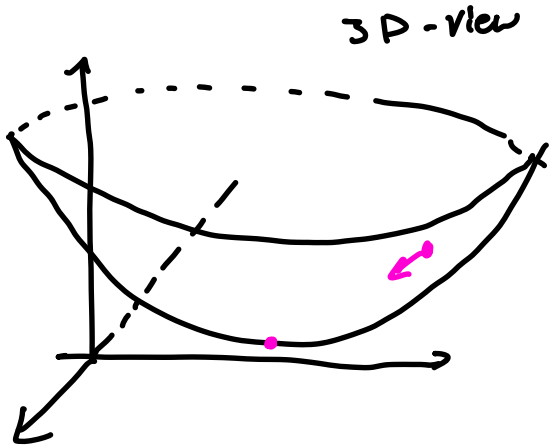


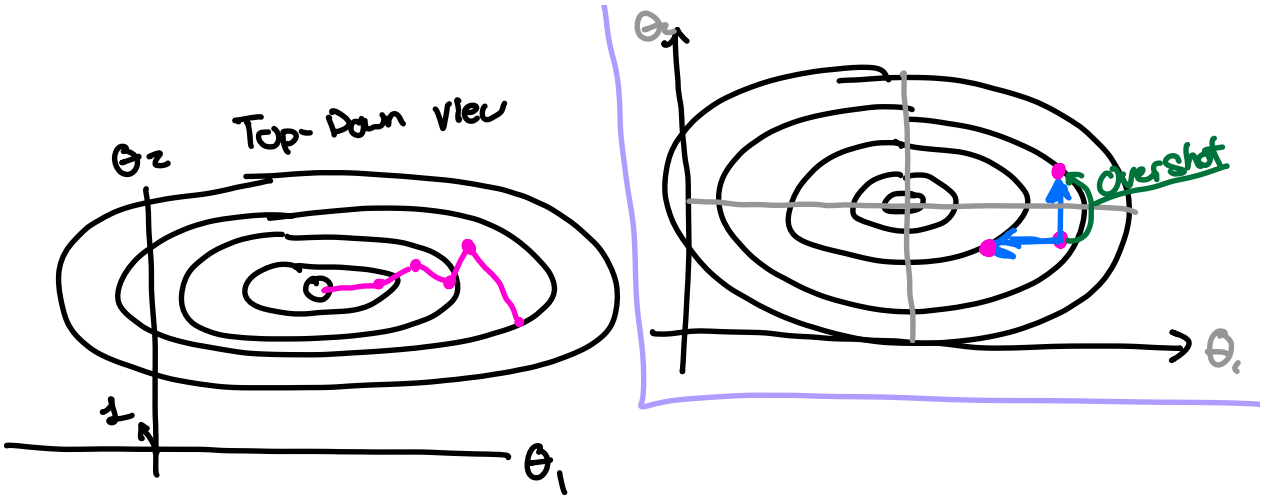
- How do we pick η
- Should it be the same for all parameters.
- How do we escape saddle points?

Saddle Points



Momentum





"Exponential Moving Average" of ∇

$$v_{t+1} := \beta_m v_t + (1-\beta_m)\nabla_\theta J_b$$

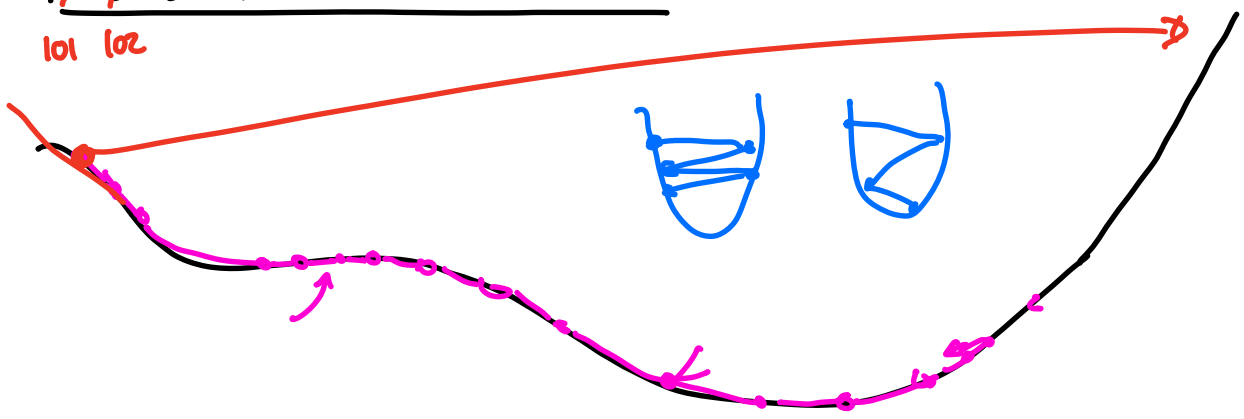
$$\theta_{t+1} := \theta_t - \eta v_{t+1}$$

velocity, friction, velocity, acceleration; averaging; rolling average gradients

100 time step average

1	2	3	4	...	100

101 102



Adaptive Learning Rates

RMSProp "Root-Mean-Square Backprop"

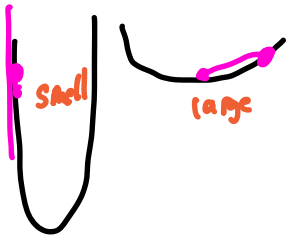
Average magnitude of gradient.

$$g^2_{t+1} := \beta_R g^2_t + (1-\beta_R)\nabla_\theta L_b^2$$

$$\theta_{t+1} := \theta_t - \eta \frac{\nabla_\theta L_b}{\sqrt{g^2_{t+1}} + \epsilon}$$

Effective Learning Rate

↑ very small value



- What if g^2 is large? → smaller steps
- What if g^2 is small? → larger steps

Adam "Adaptive Moment Estimation"

- A good first choice
- Combines momentum w/ RMSProp

$$v_{t+1} = \beta_m v_t + (1 - \beta_m) \nabla_\theta \mathcal{L}_b$$

$$\hat{v}_{t+1} = \frac{v_{t+1}}{1 - \beta_m^t}$$

$0.9^{(t)}$ $\quad \frac{1}{1-0.9} = \frac{1}{0.1}, \quad \frac{1}{1-0.9^{100}} = 1$

Bias correction for EMA

$$g^2_{t+1} = \beta_R g^2_t + (1 - \beta_R) \nabla_\theta \mathcal{L}_b^2$$

$$\hat{g}^2_{t+1} = \frac{g^2_{t+1}}{1 - \beta_m^t}$$

$$\theta_{t+1} := \theta_t - \eta \frac{\hat{v}_{t+1}}{\sqrt{\hat{g}^2_{t+1}} + \epsilon}$$

↙ momentum

↖ adaptive learning rate