Advanced gradient descent

ISTD 50.035

Computer Vision

Acknowledgement: Some images are from various sources: UCF, Stanford cs231n, etc.

Gradient descent variants

- Batch gradient descent
  - Compute gradient using the entire training dataset
  - One update per epoch
  - Very slow

$$W' = W - \gamma \nabla L$$

$$w_l' = w_l - \gamma \frac{\partial L}{\partial w_l}$$

- Stochastic gradient descent
  - Compute gradient using only one training example
  - One update per training example
  - Unstable

- Minibatch gradient descent
  - Performs an update for every mini-batch of n training examples (sometimes also called SGD)

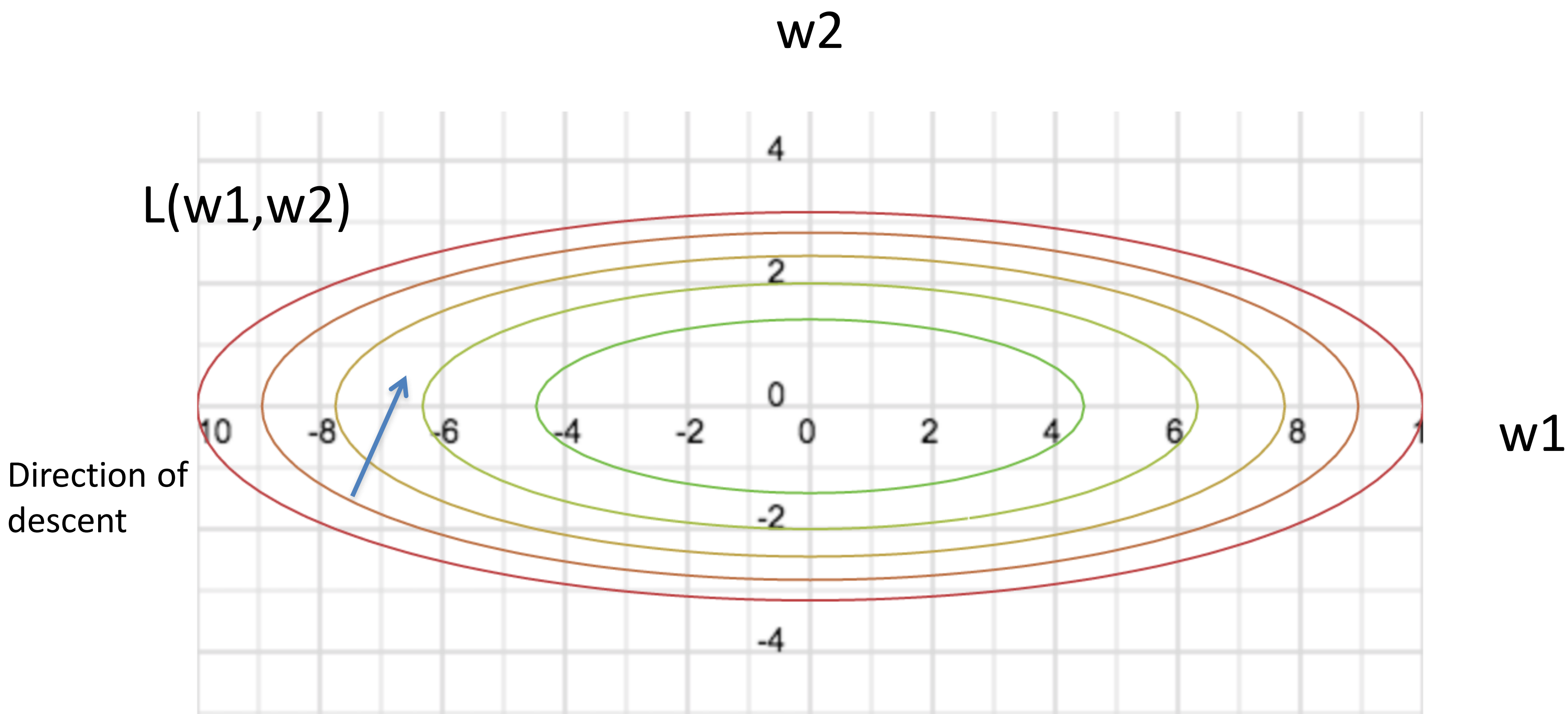
Issues with GD

- High dimensional, non-convex error function L(W)

- Difficult to choose learning rate (step size)
  - Too small: slow convergence
  - Too large: fluctuate around the minimum

- Same learning rate to all parameter

Gradient Descent

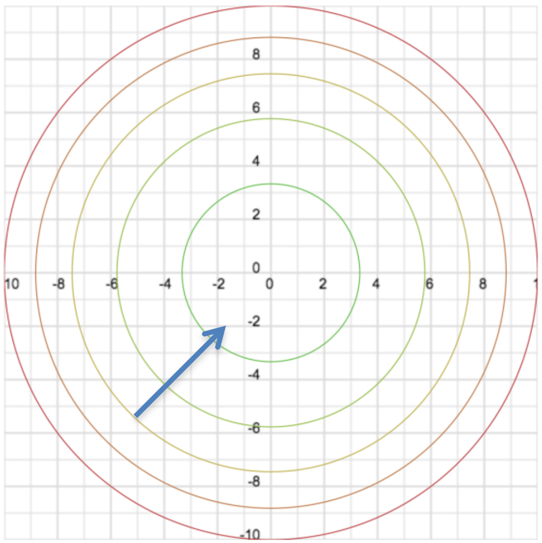


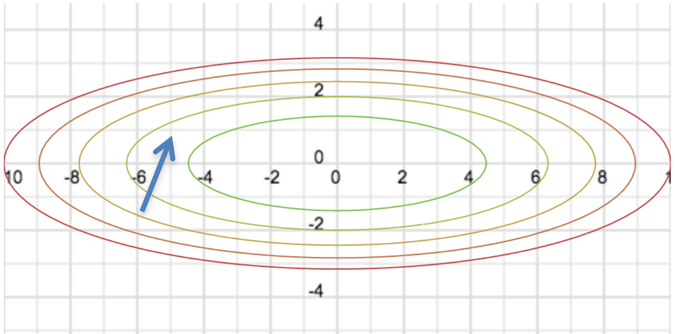
Level curve/surface: L(w1, w2, …) = c for some constant c

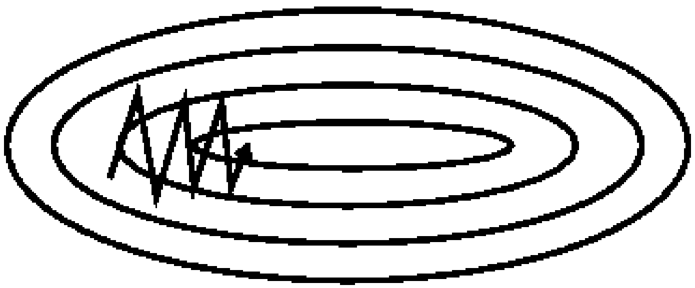
∇L Normal to level curve

Direction of maximal increase of L(.)

Gradient Descent

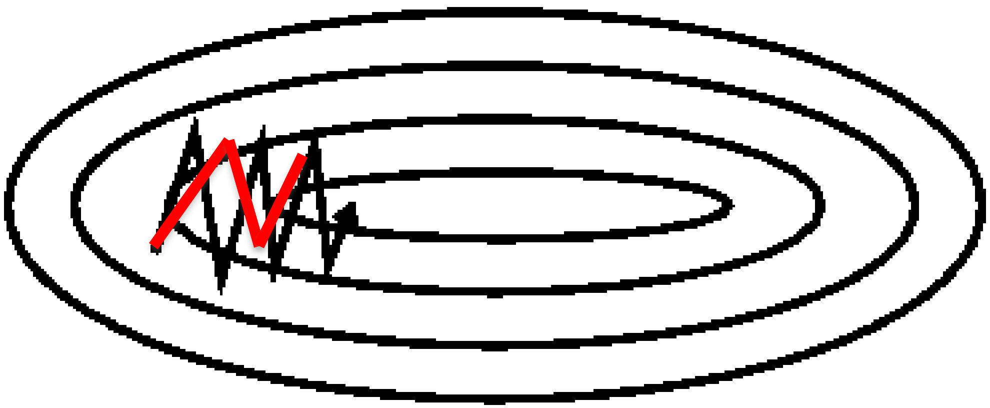






GD: more trouble if the loss function is much more steeply in some dimensions than in other: slow converges
Ideas to improve GD: make more update in the 'right' dimension (w1 here)

RMSprop



w1: smaller gradient (dL/dw -> same dL but larger dw)
w2: larger gradient (steeper)

-> make larger update in dim with small grad (w1), smaller update in dim with large grad (w2)

Moving average:

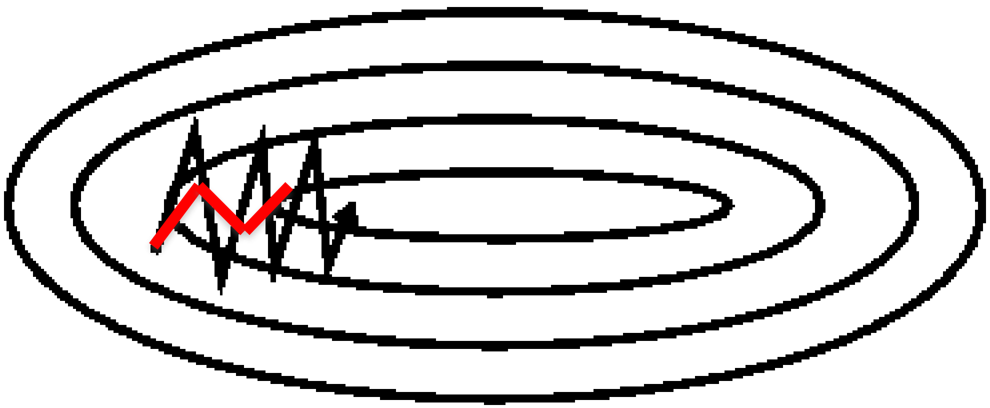
$$s_{dw_i} = \beta \cdot s_{dw_i} + (1-\beta)(dw_i)^2$$

Std. GD:

$$w_i' = w_i - \gamma dw_i$$

$$w_i' = w_i - \gamma \frac{dw_i}{\sqrt{s_{dw_i}}}$$

GD with Momentum



Reduce update for dimensions where grad change direction

v_{dw_i} (result) is small if opp. signs

Std. GD:

$$w_i' = w_i - \gamma dw_i$$

$$v_{dw_i} = \beta \cdot v_{dw_i} + (1 - \beta)(dw_i)$$

$$w_i' = w_i - \gamma v_{dw_i}$$

Adam (Adaptive moment estimation)

Std. GD:

$$w_i' = w_i - \gamma dw_i$$

Adam: Combine momentum and RMSprop

During the t-th parameter update:

$$v_{dw_i} := \beta \cdot v_{dw_i} + (1-\beta)(dw_i)$$

$$v_{dw_i} := \frac{v_{dw_i}}{1-\beta^t}$$

$$s_{dw_i} := \alpha \cdot s_{dw_i} + (1-\alpha)(dw_i)^2$$

$$s_{dw_i} := \frac{s_{dw_i}}{1-\alpha^t}$$

$$w_i' = w_i - \gamma \frac{v_{dw_i}}{\sqrt{s_{dw_i}}}$$