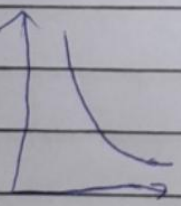
Variation of Gradient Descent optimization

Batch Gradient Descent vs stochastic gradient descent vs Mini Batch GD

Batch GD

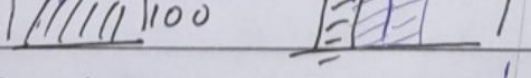
→ Gradient is computed over entire dataset before making an update to model parameters



→ loss decreases smoothly

Stochastic GD

→ gradient is computed over 1 training example before making update



$\theta = \theta - \eta \text{ grad}$

1 epoch

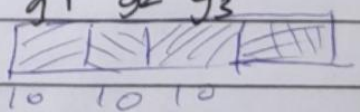
1 update

→ faster conversion but noisy updates

→ unable to use vectorization

Mini Batch GD

→ subset of training data is created before making update



1 epoch = 10 mini batch

$\theta = \theta - \eta g^1$

$\theta = \theta - \eta g^2$

:

more frequent updates

faster convergence

Page No.
Date :

Batch Gradient Descent

Pseudocode

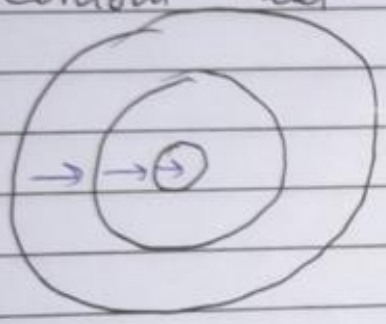
1 epoch | | | x, y |

data

1 epoch iterate over entire data.

for e in range (MAX-ITRS):
    grad = gradient(x, y)
    θ = θ − η · grad

Contour Plot



Loss Function

Mini Batch Gradient Descent

Pseudocode

1 epoch | b1 | b2 | b3 | [...]
x' y'

batch no = m
batch size

for e in range (max-itr):
    shuffle(data)
    for batch in all_batches:
        x', y' = load_batch()
        grad = calc_grad(x, y)
        θ = θ − η grad

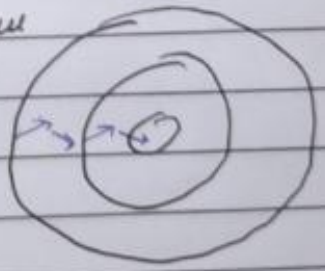
freq of updates

↓

- faster more computation convergence

→ updates can be noisy
→ achieve local minima much earlier

Contour plot



loss function

noisy loss curve