

SPPU-BE-COMP-CONTENT - KSKA Git

ML

CLASSMATE

Date :

Page :

ASSIGNMENT-3

Q1

F

FEATURE

BATCH GRADIENT

STOCHASTIC

MINI BATCH

DESCENT

GRADIENT DESCENT

GRADIENT DESCENT

1) Data used per update

Entire training dataset

Single data point

Small subset (mini batch)

2) Speed

Slow for large datasets

Fast updates

Balanced

3) convergence

Smooth, stable

Noisy, fluctuates

Smooth with some noise

4) Memory usage

High

Low

Moderate

5) When to use

Small datasets

Very large datasets

large datasets
GPU training

Q2

Goal: Minimise the cost function (Mean Squared Error)

$$J(\theta) = \frac{1}{2m} \sum (y - \hat{y})^2$$

Process:

1. Initialize weights randomly.

SPPU-BE-COMP-CONTENT - KSKA Git

2. Compute gradient of the cost function wrt each weight

$$\frac{\partial J}{\partial \theta} = -\frac{1}{n} \sum (y - \hat{y})x$$

3. update weight:

$$\theta = \theta - \alpha \cdot \text{gradient}$$

4. Repeat until convergence

- Effect: Iteratively reduces error by moving weights in the direction of steepest descent.

Q3

Gradient descent doesn't always converge to an optimum, it depends on

- Learning rate (α): Too large \rightarrow overshooting, too small \rightarrow very slow
- cost function shape: Convex \rightarrow guaranteed global minimum; Non convex \rightarrow may get stuck in local minima or saddle points.
- Initialization of weights // poor initializⁿ can slow or prevent convergence.