Session 1

Analytic vs Numerical Optimization – Multiple Choice Questions

1. An analytic solution is generally preferred in mathematics and machine learning because it is:

- a) An approximation
- b) Slower to calculate
- c) Only available for large datasets
- **d) Faster and provides an exact solution**

---

2. When is an approximate solution typically necessary in machine learning?

- a) When the data matrix is square
- b) When an analytic solution generalizes well
- **c) When the data matrix is not square or when an exact solution does not generalize well**
- d) Only for single variable linear regression

---

3. For an overdetermined system (more equations than unknowns) in linear models, an approximate solution can be found algebraically using the:

- a) Least norm method
- b) Matrix inversion directly
- **c) Least square method**
- d) Newton's method

---

4. Which of the following is a key characteristic of a numerical solution for finding optimal model parameters?

- a) It directly calculates the exact parameter values

- b) It is only applicable to linear models
- **c) It involves iterative steps of prediction, error evaluation, and parameter updates**
- d) It guarantees finding the global minimum in one step

---

5. The L^1 norm is directly related to which loss function?

- a) Mean Squared Error (MSE)
- **b) Mean Absolute Error (MAE)**
- c) Euclidean Distance (related to L^2 norm)
- d) Sum of squared residuals (related to MSE)

---

6. What is a notable property of the Mean Absolute Error (MAE) loss function?

- a) It heavily penalizes outliers
- **b) It is robust to outliers**
- c) It has a changing gradient that enables better learning
- d) Its gradient is always zero at the minimum

---

7. The Mean Squared Error (MSE) loss function is characterized by:

- a) Being highly robust to outliers
- b) Having a constant gradient throughout the parameter space
- **c) Having a changing gradient that facilitates better learning**
- d) Its gradient being undefined at the minimum

---

8. The gradient of a function represents:

- a) The area under the function's curve
- b) The difference between the predicted and actual values

- **c) The derivative, indicating the sensitivity of the output to changes in the input**
- d) The second derivative of the function

---

9. In the context of optimization, the gradient provides information about the:

- a) Overall value of the loss function
- b) Number of data points
- **c) Direction and magnitude of the steepest ascent of the function**
- d) Learning rate to be used

---

10. Gradient Descent (GD) is an algorithm used to:

- a) Find the analytic solution to a linear system
- b) Directly calculate the inverse of a matrix
- **c) Minimize a function by iteratively moving in the direction of the negative gradient**
- d) Determine if a function is convex

---

11. The primary goal of numerical optimization in machine learning is to find:

- a) The exact solution to any problem
- b) A solution that perfectly fits the training data
- **c) The optimal model parameters that minimize the prediction error**
- d) The fastest way to compute matrix inverses

---

12. What happens to the model parameters (θ) during the update step in Gradient Descent if the gradient is positive?

- a) θ will remain unchanged
- **b) θ will decrease**
- c) θ will increase

- d) The learning rate will be adjusted

---

13. The learning rate in Gradient Descent controls the:

- a) Direction of movement in the parameter space
- **b) Size of the steps taken towards the minimum of the loss function**
- c) Number of iterations the algorithm will run
- d) Method used to calculate the gradient

---

14. Choosing a very large learning rate in Gradient Descent can lead to:

- a) Very slow convergence
- b) Guaranteed convergence to the global minimum
- **c) Overshooting the minimum and potentially divergence**
- d) Faster convergence without any risks

---

15. A local minimum in the context of a loss function is a point where:

- a) The function value is the smallest among all possible points
- b) The gradient is non-zero
- **c) The function value is smaller than at nearby points, but possibly greater than at a distant point**
- d) The analytic solution can be easily calculated

---

16. A global minimum of a loss function is:

- a) Any point where the gradient is zero
- b) A minimum that is only valid for the training data
- **c) A point where the function value is smaller than at all other feasible points**
- d) Always easily found by Gradient Descent

---

17. A convex function has the property that:

- a) It can have multiple local minima
- b) Gradient Descent is guaranteed to diverge on it
- **c) Any local minimum is also its global minimum**
- d) Its Hessian matrix is always negative definite

---

18. In linear regression, the goal is to find the optimal values of:

- a) The input features
- b) The prediction error
- **c) The model parameters (weights and biases)**
- d) The learning rate

---

19. The steps involved in a numerical solution typically include:

- a) Analytical derivation of the solution
- b) Solving a system of linear equations directly
- **c) Initialization of parameters, prediction, error evaluation, and parameter updates**
- d) Calculating the exact inverse of the data matrix

---

20. The choice between MAE and MSE as a loss function often depends on:

- a) The size of the dataset
- b) The complexity of the model
- **c) The sensitivity to outliers**
- d) The availability of an analytic solution

Session 2

Gradient Descent & Optimization – Multiple Choice Questions

1. A partial derivative of a multivariable function measures:

- a) The total rate of change of the function
- b) The integral of the function with respect to one variable
- **c) The rate of change with respect to one variable while holding others constant**
- d) The second derivative of the function

---

2. The gradient of a multivariable function is a:

- a) Scalar value representing the overall slope
- **b) Vector of partial derivatives, each with respect to one of the variables**
- c) Matrix of second-order partial derivatives
- d) Single value indicating the direction of the minimum

---

3. A contour plot is a useful visualization tool in optimization as it shows:

- a) The path taken by the Gradient Descent algorithm
- b) The magnitude of the gradient at different points
- **c) The level sets (curves of constant function value) of a multi-variable function**
- d) The learning rate at each iteration

---

4. When applying Gradient Descent to Linear Regression, the model parameters (e.g., θ_0 and θ_1) should be updated:

- a) Sequentially, updating one parameter at a time
- **b) Simultaneously, using the gradients calculated at the same point**
- c) Only after the cost function starts to increase
- d) With different learning rates for each parameter initially

---

5. In the context of Gradient Descent for Single Variable Linear Regression, as you get closer to the minimum of the cost function:

- a) The learning rate should always be increased
- b) The gradient remains constant
- **c) The learning step (change in parameter value) will naturally decrease even with a fixed learning rate because the gradient becomes smaller**
- d) The parameters start to oscillate wildly

---

6. Using a very large learning rate in Gradient Descent can lead to:

- a) Slow and steady convergence
- **b) Overshooting the minimum and potentially diverging**
- c) Guaranteed convergence to the global minimum
- d) More accurate gradient calculations

---

7. Using a very small learning rate in Gradient Descent can lead to:

- a) Overshooting the minimum frequently
- **b) Very slow convergence, taking a long time to reach the minimum**
- c) A higher chance of escaping local minima
- d) Unstable parameter updates

---

8. Batch/Vanilla Gradient Descent calculates the gradient of the cost function:

- a) With respect to a single randomly chosen data point
- b) With respect to a small subset of the training data
- **c) With respect to the entire training dataset**
- d) Only for misclassified examples

---

9. A key advantage of Batch/Vanilla Gradient Descent is that:

- a) It is very fast for large datasets
- b) It can easily escape local minima due to noisy updates
- **c) It has a straight trajectory towards the minimum and is guaranteed to converge to the global minimum for a convex loss function (in theory)**
- d) It efficiently handles redundant data points

---

10. A significant disadvantage of Batch/Vanilla Gradient Descent is that:

- a) It often gets stuck in oscillations near the minimum
- b) It performs too many updates per epoch
- **c) It can be very slow for large datasets as each update requires processing the entire dataset**
- d) It is prone to the vanishing gradient problem

---

11. Feature scaling is an important preprocessing step for gradient-based optimization algorithms because:

- a) Always improves the accuracy of the model
- b) Is only necessary for linear regression
- **c) Helps ensure that all features contribute more equally to the gradient and can speed up convergence**
- d) Prevents overfitting by reducing the magnitude of the weights

---

12. Min-Max Normalization scales features to a range between:

- a) -1 and 1
- b) Around 0 with a standard deviation of 1
- c) Based on the interquartile range
- **d) 0 and 1**

---

13. Mean Normalization (standardization) scales features so that they have a:

- a) Range between 0 and 140
- **b) Mean of 0 and a standard deviation of 1**
- c) Range based on the minimum and maximum values
- d) Distribution that is skewed

---

14. For machine learning algorithms that are not distance-based, feature scaling is:

- a) Always required for optimal performance
- b) Essential for interpretability of coefficients
- **c) Not strictly needed in all cases**
- d) Only beneficial for high-dimensional data

---

15. The choice of learning rate in Gradient Descent is crucial because:

- a) A fixed learning rate always guarantees convergence
- b) The learning rate determines the final accuracy of the model
- **c) An inappropriate learning rate can lead to slow convergence, oscillations, or divergence**
- d) The optimal learning rate can be calculated analytically

---

16. In Multivariate Linear Regression, the hypothesis function involves:

- a) Only one input feature
- b) No intercept term
- **c) Multiple input features, each with its own weight**
- d) A non-linear combination of input features

---

17. The gradient in Multivariate Linear Regression is a vector containing:

- a) The cost function value for each data point
- b) The predicted output for each input feature
- **c) The partial derivative of the cost function with respect to each model parameter**
- d) The learning rate and the number of iterations

---

18. Updating model parameters simultaneously in Gradient Descent means that:

- a) Each parameter is updated with a different learning rate
- b) The updates for all parameters are calculated based on the new values of other parameters in the same iteration
- **c) The new value for each parameter is calculated based on the parameter values from the previous iteration, and then all parameters are updated at once**
- d) The data is processed in parallel to speed up the update

---

19. A local minimum can be problematic for Gradient Descent because:

- a) The learning rate will always become too small there
- b) The gradient at a local minimum is non-zero
- **c) The algorithm might converge to it even if a better (global) minimum exists elsewhere**
- d) It only occurs in convex optimization problems

---

20. The process of repeating the gradient descent update step until a certain condition is met is called:

- a) Feature engineering
- b) Hypothesis testing
- **c) Iteration until convergence**
- d) Gradient calculation

Session 3

---

1. Batch Gradient Descent Limitation

Q: A major problem with Batch Gradient Descent for very large datasets is:

a) Frequent parameter updates lead to instability

b) It always finds the global minimum quickly

c) Each parameter update requires processing the entire dataset, making it computationally expensive and slow

d) It is less likely to get stuck in local minima

---

2. Parameter Updates in SGD

Q: Stochastic Gradient Descent (SGD) updates model parameters:

a) After processing the entire training dataset

b) After processing a small batch of data

c) For each individual training observation

d) Only at the end of each epoch

---

3. Advantage of SGD

Q: One advantage of Stochastic Gradient Descent (SGD) compared to Batch GD is:

a) Smoother convergence towards the minimum

b) More efficient utilization of vectorization

c) It can be faster per epoch and might escape local minima more easily due to noisy updates

d) Guaranteed convergence to the global minimum for convex functions

---

4. Disadvantage of SGD

Q: A key disadvantage of Stochastic Gradient Descent (SGD) is:

a) It requires a very small learning rate

b) It converges directly to the exact minimum without oscillations

c) Its updates can be very noisy, leading to oscillations around the minimum and potentially slow convergence overall

d) It is not suitable for non-convex loss functions

---

5. Parameter Updates in Mini-batch GD

Q: Mini-batch Gradient Descent updates parameters based on:

a) A single randomly chosen data point

b) A small, randomly chosen subset of the training data

c) The entire training dataset

d) All misclassified data points

---

6. Advantage of Mini-batch GD

Q: Mini-batch Gradient Descent offers a compromise between Batch GD and SGD by:

a) Having the same computational cost as Batch GD per update

b) Introducing more noise than SGD in the updates

c) Having more stable convergence than SGD and being more computationally efficient than Batch GD per update

d) Not requiring shuffling of the data

---

7. Role of Learning Rate

Q: The learning rate in Gradient Descent determines:

a) The overall shape of the loss function

b) Whether the algorithm will find a local or global minimum

c) The step size taken in the direction of the negative gradient during parameter updates

d) The number of epochs required for convergence

---

8. Choosing a Good Learning Rate

Q: Finding a good value for the learning rate typically involves:

a) Calculating it analytically based on the dataset size

b) Using a fixed value throughout training without any adjustments

c) Experimenting with different values and observing the behavior of the cost function

d) Starting with a very small value and gradually increasing it

---

9. Oscillating or Increasing Cost Function

Q: If the cost function oscillates or increases during Gradient Descent, it might indicate that the learning rate is:

a) Too small

b) Too large, causing overshooting

c) Optimally set for convergence

d) Not affecting the learning process

---

10. Local Minima Challenge

Q: Local minima can be a challenge for Gradient Descent because:

a) The gradient is non-zero at a local minimum

b) They only occur in high-dimensional parameter spaces

c) The algorithm might get stuck in a local minimum and not reach a better global minimum

d) The learning rate always becomes zero at a local minimum

---

11. Strategy for Avoiding Local Minima

Q: One strategy to mitigate the problem of getting stuck in local minima is to:

a) Always use a very small learning rate

b) Ensure that the cost function is always convex

c) Run the optimization algorithm multiple times with different random initializations

d) Use Batch Gradient Descent on small datasets

---

12. Vanishing Gradient Problem

Q: The vanishing gradient problem in deep learning occurs when:

a) The learning rate is set to a very large value

b) Gradients increase exponentially during backpropagation

c) Gradients become extremely small as they are backpropagated through many layers, hindering learning in earlier layers

d) The optimization algorithm gets stuck in a saddle point

---

13. Exploding Gradient Problem

Q: The exploding gradient problem in deep learning is similar to having:

a) A learning rate that is too small

b) A learning rate that is too large, causing unstable learning and divergence

c) A perfectly flat loss surface

d) A well-regularized model

---

14. Purpose of Momentum-based GD

Q: Momentum-based Gradient Descent aims to accelerate learning by:

a) Decreasing the learning rate over time

b) Adding a fraction of the previous update vector to the current update vector, helping to move faster in consistent directions and dampening oscillations

c) Updating parameters based on individual data points

d) Using an adaptive learning rate for each parameter

---

15. Range of Momentum Term (γ)

Q: The momentum term (γ) in Momentum-based GD typically takes values between:

a) 0 and infinity

b) 0 and 1

c) -1 and 1

d) 1 and infinity

---

16. Effect of Increasing Momentum

Q: Increasing the value of the momentum term (γ) generally leads to:

a) Slower convergence and fewer oscillations

b) More precise convergence to the minimum

c) Faster movement in consistent directions but potentially increased oscillations and overshooting

d) A complete avoidance of local minima

---

17. Nesterov Accelerated Gradient (NAG)

Q: Nesterov Accelerated Gradient (NAG) differs from standard momentum by:

a) Not using any form of momentum

b) Calculating the gradient at the current parameter values

c) First making a "look-ahead" step in the direction of the previous momentum and then calculating the gradient at that new position to make a correction

d) Using a fixed learning rate for all parameters

---

18. Benefit of the "Look-ahead" Step in NAG

Q: The "look-ahead" step in Nesterov Accelerated Gradient (NAG) helps to:

a) Increase the learning rate adaptively

b) Completely eliminate oscillations near the minimum

c) Make a more informed gradient calculation and often leads to faster convergence and reduced oscillations compared to standard momentum

d) Solve the vanishing gradient problem in deep networks

---

19. Shuffling in SGD and Mini-batch GD

Q: In both SGD and Mini-batch GD, it is generally preferable to shuffle the training data before each epoch to:

a) Reduce the computational cost per update

b) Ensure that the same subset of data is used for each update

c) Introduce more randomness in the parameter updates, which can help escape local minima and improve generalization

d) Make the gradient calculations more accurate

---

20. Choosing Mini-batch Size

Q: The choice of batch size in Mini-batch Gradient Descent is a hyperparameter that can be tuned. Common values often are:

a) Prime numbers greater than 100

b) Very large numbers close to the dataset size

c) Powers of 2, such as 32, 64, 128, etc.

d) Small odd numbers less than 10

Session 4

---

1. Adaptive learning rate methods like Adagrad, RMSProp, and Adam primarily aim to:

a) Solve the problem of exploding gradients in deep networks

b) Accelerate learning in gently sloped loss functions using momentum

c) Adjust the learning rate for each parameter individually based on the history of its gradients

d) Guarantee convergence to the global minimum for non-convex functions

---

2. Adagrad adapts the learning rate for each parameter by:

a) Using the same decaying learning rate for all parameters

b) Increasing the learning rate for parameters with frequently large gradients

c) Decreasing the effective learning rate for parameters that have received large, frequent updates in the past (sparse features get larger effective learning rates)

d) Using momentum to adjust the learning rate based on the direction of movement

---

3. A key advantage of Adagrad is that:

a) It prevents the learning rate from decaying too quickly

b) It works very well with dense features that are updated frequently

c) Parameters associated with sparse features (infrequently updated) receive larger learning rates, leading to better updates

d) It incorporates momentum to accelerate learning in flat regions

---

4. A significant drawback of Adagrad is that:

a) It increases the learning rate indefinitely, leading to divergence

b) It does not handle sparse features effectively

c) The accumulated sum of squared gradients in the denominator continuously increases, causing the learning rate to become very small and potentially stop learning too early

d) It is computationally very expensive due to the accumulation of gradients

---

5. RMSProp addresses the aggressive learning rate decay of Adagrad by:

a) Using a fixed learning rate for all parameters

b) Accumulating the squared gradients without any decay

c) Using an exponentially decaying average of past squared gradients in the denominator, preventing the learning rate from becoming excessively small

d) Incorporating a momentum term to counteract the learning rate decay

---

6. The exponentially weighted moving average (EWMA) used in RMSProp helps to:

a) Give equal weight to all past squared gradients

b) Completely ignore older squared gradients

c) Give more weight to recent squared gradients and less weight to older ones, providing a more adaptive learning rate

d) Eliminate the need for a learning rate hyperparameter

---

7. Adam (Adaptive Moment Estimation) combines ideas from:

a) Only Adagrad and Nesterov Accelerated Gradient (NAG)

b) Only RMSProp and Batch Gradient Descent

c) Momentum-based Gradient Descent and RMSProp

d) Only Stochastic Gradient Descent (SGD) and Adagrad

---

8. Adam computes:

a) Only an adaptive learning rate for each parameter based on the second moment of the gradients

b) Only an exponentially decaying average of past gradients (momentum) for each parameter

c) Both an exponentially decaying average of past gradients (first moment - like momentum) and an exponentially decaying average of past squared gradients (second moment - like RMSProp) for each parameter to adapt the learning rate

d) Neither adaptive learning rates nor adaptive momentum

---

9. The parameters traditionally used in the Adam optimization algorithm are:

a) A single learning rate η and a decay factor β

b) A momentum term γ and a learning rate η

c) Learning rate η, first moment decay rate β_1, second moment decay rate β_2, and a small constant ε

d) Only decay rates β_1 and β_2

---

10. Bias correction is applied in Adam to:

a) Prevent overfitting by penalizing large weights

b) Accelerate the convergence speed in later iterations

c) Make the estimates of the first and second moments more accurate during the initial iterations when the averages are biased towards zero

d) Ensure that the learning rate for all parameters remains within a certain range

---

11. Compared to standard Momentum-based GD, Adam typically:

a) Requires more manual tuning of the learning rate

b) Converges slower on sparse datasets

c) Adapts the learning rate for each parameter, often leading to faster and more stable convergence across a wider range of problems

d) Is less effective at escaping saddle points in the loss landscape

---

12. The small constant ε (epsilon) in the denominator of the update rules for Adagrad, RMSProp, and Adam is used to:

a) Increase the learning rate for small gradients

b) Decrease the learning rate for large gradients

c) Prevent division by zero, ensuring numerical stability

d) Introduce more randomness into the optimization process

---

13. The update rule in Adagrad involves dividing the learning rate by:

a) The exponentially decaying average of squared gradients

b) The current gradient value

c) The square root of the accumulated sum of squared gradients

d) A constant momentum term

---

14. The update rule in RMSProp involves dividing the learning rate by:

a) The accumulated sum of squared gradients

b) The current gradient value

c) The square root of the exponentially decaying average of squared gradients

d) A constant momentum term

---

15. The first moment estimate (m(t)) in Adam is analogous to:

a) The adaptive learning rate in Adagrad

b) The squared gradient term in RMSProp

c) The velocity term in Momentum-based GD

d) The bias correction factor

---

16. The second moment estimate (v(t)) in Adam is related to:

a) The momentum term

b) The first moment estimate

c) The uncentered variance of the gradients, used for adaptive learning rates (similar to RMSProp)

d) The bias in the initial steps

---

17. For training deep neural networks, Adam with mini-batch gradient descent is generally preferred due to its:

a) Guaranteed convergence to the global minimum for non-convex functions

b) Simplicity and need for minimal hyperparameter tuning

c) Adaptive learning rates and momentum, which often lead to faster and more stable training

d) Ability to handle exploding gradients without any modifications

---

18. The concept of using different learning rates for different parameters is a key aspect of:

a) Standard Momentum-based GD

b) Vanilla Batch Gradient Descent

c) Adaptive learning rate methods like Adagrad, RMSProp, and Adam

d) Stochastic Gradient Descent

---

19. Decaying the learning rate over time is a strategy that can help:

a) Escape local minima more easily

b) Accelerate the initial stages of learning

c) Improve convergence to the minimum by taking larger steps initially and smaller steps as the minimum is approached

d) Prevent the learning rate from becoming too small in Adagrad

---

20. The motivation behind accelerated gradient descent methods (Momentum and NAG) was primarily to address the issue of:

a) Aggressive learning rate decay

b) Slow progress in flat regions of the loss function

c) Sensitivity to outliers in the data

d) High computational cost per iteration

Session 5

---

1. Gradient Descent is classified as a:

a) Zeroth-order optimization method

b) First-order optimization method

c) Second-order optimization method

d) Higher-order optimization method

---

2. First-order optimization methods like Gradient Descent primarily use which information to find the minimum of a function?

a) The value of the function itself

b) The first derivative (gradient) of the function

c) The second derivative (Hessian) of the function

d) Approximations of higher-order derivatives

---

3. Newton's method is an example of a:

a) First-order optimization method

b) Second-order optimization method

c) Stochastic optimization method

d) Gradient-free optimization method

---

4. Newton's method approximates the objective function locally using a:

a) Linear function

b) Plane

c) Paraboloid (quadratic function)

d) Series of straight line segments

---

5. A key advantage of Newton's method over standard Gradient Descent is its:

a) Lower computational cost per iteration

b) Simpler implementation and fewer hyperparameters

c) Typically faster convergence rate (quadratic convergence) when the initial guess is close to the minimum

d) Guarantee of finding the global minimum for non-convex functions

---

6. A major drawback of Newton's method, especially for high-dimensional problems in machine learning, is the:

a) Need for careful tuning of the learning rate

b) Tendency to get stuck in local minima more often

c) High computational cost of calculating and inverting the Hessian matrix, which scales as $O(n^3)$

d) Inability to handle non-convex objective functions

---

7. The Hessian matrix contains:

a) The first-order partial derivatives of the function

b) The second-order partial derivatives of the function

c) The gradient of the function at different points

d) Information about the step size in Gradient Descent

---

8. If the Hessian matrix at a stationary point is positive definite, then that point is a:

a) Local maximum

b) Saddle point

c) Local minimum

d) Global maximum

---

9. If the Hessian matrix at a stationary point is negative definite, then that point is a:

a) Local maximum

b) Local minimum

c) Saddle point

d) Global minimum

---

10. If the Hessian matrix at a stationary point is indefinite, then that point is a:

a) Local maximum

b) Local minimum

c) Saddle point

d) Global optimum

---

11. Quasi-Newton methods aim to:

a) Calculate the exact inverse Hessian efficiently

b) Avoid using gradient information altogether

c) Approximate the Hessian matrix or its inverse to reduce computational cost while still leveraging second-order information

d) Perform optimization only on convex functions

---

12. The secant method is a one-dimensional quasi-Newton method that approximates the:

a) Function value

b) Gradient using finite differences

c) Second derivative using finite differences of the first derivative

d) Optimal learning rate

---

13. In multi-dimensional quasi-Newton methods, the approximate Hessian (B) is typically updated using:

a) Only the gradient at the current iteration

b) Randomly generated values

c) Information from previous iterations, such as the change in parameters (Δx) and the change in gradients (y)

d) Only the current learning rate

---

14. The quasi-Newton condition (or secant equation) relates the approximate Hessian at the next step (B_{k+1}) to the:

a) Learning rate and the current gradient

b) Previous Hessian and a random matrix

c) Change in parameters (Δx) and the change in gradients (y)

d) Function value at the current and previous steps

---

15. One of the main advantages of quasi-Newton methods over Newton's method is that they:

a) Always find the global minimum

b) Do not require the objective function to be differentiable twice

c) Usually generate an estimate of the inverse Hessian directly, avoiding the costly matrix inversion step

d) Have a guaranteed quadratic convergence rate for all types of functions

---

16. BFGS (Broyden–Fletcher–Goldfarb–Shanno) is a popular:

a) First-order optimization algorithm

b) Stochastic optimization technique

c) Quasi-Newton method

d) Gradient-free optimization method

---

17. In the BFGS method, the approximate inverse Hessian is updated by:

a) Directly calculating the inverse of the true Hessian

b) Using only the gradient at the current step

c) Adding rank-one or rank-two update matrices based on the changes in parameters

and gradients

d) Multiplying the previous inverse Hessian by a constant learning rate

---

18. The Taylor series provides a way to:

a) Find the roots of a function exactly

b) Calculate the gradient of a complex function in one step

c) Approximate a function around a specific point using its derivatives at that point

d) Determine if a function is convex or non-convex

---

19. First-order Taylor series approximation is used in:

a) Newton's method

b) Gradient Descent

c) BFGS

d) Secant method

---

20. Second-order Taylor series approximation is utilized in the derivation of:

a) Stochastic Gradient Descent

b) Momentum-based Gradient Descent

c) Newton's method

d) Adagrad