20 Sample Questions for an Applied Optimization Course in Data Science

1. **Optimization in Data Science: Real-World Context**

   Question:

   - Define what an *optimization problem* is, and list two examples from data science (e.g., parameter tuning, resource allocation).
   - Describe why optimization problems are important in applications such as recommendation systems or supply chain management.
   - In your own words, explain one challenge (e.g., scalability, non-convexity, interpretability) that might arise in real-world optimization scenarios.

2. **Convex vs. Non-Convex Functions in Practice**

   Question:

   - Illustrate the concept of a *convex function* with a simple example (e.g., a parabola) and a *non-convex function* (e.g., a multi-modal function).
   - Discuss why convex problems are generally easier to solve and how this impacts data science tasks such as linear regression.
   - Give one practical example of a non-convex optimization scenario in machine learning (e.g., training a neural network). Why can it be difficult to find a global optimum in such cases?

3. **Basic Terminology: Objectives, Constraints, and Feasibility**

   Question:

   - Define the terms *objective function*, *constraint*, and *feasible region* in the context of optimization.
   - Provide a simple, real-life case (e.g., budgeting, dieting, scheduling) to illustrate how constraints limit the possible solutions.
   - Explain the difference between a *local optimum* and a *global optimum*. Why might a local optimum be acceptable in some data science problems?

4. **Lagrange Multipliers (Single Constraint)**

   Question:

   - Given an objective function $f(x, y)$ subject to one constraint $g(x, y) = 0$, write the *Lagrangian* $\mathcal{L}(x, y, \lambda)$.
   - Derive the necessary conditions for a stationary point (i.e., set up the system of equations that must be satisfied).

- Briefly discuss a practical scenario where Lagrange multipliers might help (e.g., maximizing a likelihood under a budget constraint).

5. **Geometric Interpretation of Lagrange Multipliers**

   Question:

   - In geometric terms, how do ∇f (the gradient of the objective) and ∇g (the gradient of the constraint) relate at a solution?
   - Why does this geometric insight help in understanding constrained optimization problems?
   - Provide an example where understanding this geometry clarifies *why* a certain point is an optimum (e.g., a circle constraint and a function to be maximized).

6. **Machine Learning as Optimization: Regression Example**

   Question:

   - Formulate linear regression as an optimization problem. Specify the decision variables, objective function (least-squares), and any constraints if applicable.
   - Discuss how a typical machine learning training process (finding best-fit parameters) aligns with the idea of minimizing a loss function.
   - In practice, what are some challenges (computational or statistical) in solving this optimization problem for very large datasets?

7. **Regularization in ML: Balancing Fit and Complexity**

   Question:

   - Explain the role of a regularization term (e.g., $\lambda\|\mathbf{w}\|^2$) in ridge regression. How does it change the optimization objective?
   - Describe how regularization helps mitigate overfitting. Give a real-world example of when overfitting might be especially problematic (e.g., medical diagnostics).
   - Compare ℓ_1 (Lasso) and ℓ_2 (Ridge) regularization in terms of their effect on model parameters.

8. **Overfitting and Optimization Diagnostics**

   Question:

   - You are optimizing a model and achieve near-zero training error but poor test performance. In optimization terms, what could be going on?
   - List at least three ways to adjust your optimization framework (or the model itself) to combat overfitting (e.g., early stopping, cross-validation, adding constraints, etc.).

- How would you assess whether you have struck the *right* balance between training accuracy and generalization?

9. **Heuristics in Optimization: Concept and Use Cases**

   Question:

   - Define *heuristic* methods and explain how they differ from exact or gradient-based methods in terms of solution guarantees and computational effort.
   - Provide two examples of heuristic algorithms (e.g., Genetic Algorithms, Simulated Annealing) and describe a data science task where each might be useful.
   - What are the main trade-offs when deciding between a heuristic and a more classical approach like gradient descent or the simplex method?

10. **Heuristic Algorithm Example: Simulated Annealing**

    Question (Elaborative):

    - Summarize the basic idea of *Simulated Annealing* (SA) using terms like *temperature*, *cooling schedule*, and *random perturbation*.
    - Propose a simple optimization problem (e.g., a traveling salesman subset) and outline how SA would explore solutions and (hopefully) converge to a good one.
    - Discuss one advantage and one drawback of using SA in a typical data science pipeline (e.g., hyperparameter tuning).

11. **Hybrid Approaches in Applied Problems**

    Question:

    - Explain what is meant by combining a heuristic (e.g., Genetic Algorithm) with a local search approach (e.g., gradient descent) in a *hybrid* method.
    - Why might such an approach outperform using either method alone in a high-dimensional data science task (e.g., feature selection with a nonlinear model)?
    - Suggest one potential drawback of this approach and a way to mitigate it (e.g., increased complexity of tuning).

12. **Gradient Descent Basics**

    Question:

    - Derive the update rule for *gradient descent* on a scalar function $f(\mathbf{x})$.
    - Explain how the choice of learning rate (α) affects convergence speed and stability.
    - Give a small numerical example (or a rough sketch) illustrating how gradient descent iterates move toward a minimum.

13. **Variants of Gradient Descent: SGD and Momentum**

    Question:

    - Differentiate between *batch* gradient descent, *stochastic* gradient descent (SGD), and *mini-batch* gradient descent.
    - Explain how *momentum* modifies the gradient descent update to potentially speed up convergence and reduce oscillations.
    - In the context of large-scale machine learning (e.g., training neural networks), why might mini-batch SGD be more practical than batch gradient descent?

14. **Common Optimization Pitfalls in Gradient Methods**

    Question:

    - Describe three issues that may arise when applying gradient-based methods in practice (e.g., exploding gradients, vanishing gradients, ill-conditioned objectives).
    - Propose strategies (e.g., learning rate schedules, normalization, careful initialization) to mitigate these issues.
    - Give a brief real-world example (e.g., training a deep network for image classification) where at least one of these pitfalls was encountered and resolved.

15. **Debugging a Gradient-Based Training Process**

    Question:

    - Suppose you are training a regression model via gradient descent and notice the training loss is *not* decreasing after several iterations. List potential causes (e.g., incorrect gradient computation, too large a learning rate).
    - Explain how you would diagnose each cause. Are there specific tests or plots you would generate?
    - Suggest practical steps to fix these issues (e.g., gradient checks, adjusting hyperparameters, verifying data preprocessing).

16. **Introduction to the Simplex Method for Linear Programs**

    Question:

    - Write a linear program in standard form:

    $$\max \quad \mathbf{c}^\top \mathbf{x} \quad \text{subject to} \quad A\mathbf{x} \le \mathbf{b}, \quad \mathbf{x} \ge 0.$$

    - Summarize (in plain language) the key idea behind the simplex method: moving from one *corner* of the feasible region to another to improve the objective.
    - Give a simple real-world example (e.g., product mix selection) where a linear program can be formulated and solved using this approach.

17. **Formulating a Small Linear Program**

    Question:

    - Propose a small linear programming problem with 2–3 variables and 2–3 constraints (e.g., a simplified diet problem, portfolio allocation, or resource scheduling).
    - Explicitly write the objective function and constraints. Justify why this is a linear problem (i.e., objective and constraints are linear).
    - Suggest how you might solve it: by hand (simplex table) or using a software library (e.g., Excel Solver, Python `pulp` or `scipy`).

18. **Interpretation of Simplex Solutions in Data Science Contexts**

    Question:

    - Once the simplex method finds an optimal corner point (solution), how can you interpret the final values of decision variables in a data-driven context (e.g., resource allocations, budget allocations)?
    - Discuss how you might deal with *integer constraints* (if you needed integer solutions) in an applied setting, even though the basic simplex method is for real-valued variables.
    - Reflect on a practical scenario where linear solutions might not be exactly implementable (e.g., discrete items), and how analysts often approximate or adapt solutions in real life.

19. **Comparing Gradient-Based Methods and Simplex in Practice**

    Question:

    - In what situations would you prefer gradient-based methods (e.g., for training a neural network) over the simplex method (e.g., for a linear optimization model)?
    - Provide a brief case study example where both a linear approach (simplex) and a nonlinear approach (gradient-based) might appear. How would you decide which method is more suitable?
    - Highlight one limitation of each method in practical data science contexts (e.g., high dimensionality for simplex, non-convex objectives for gradient descent).

20. **Practical Integration: From Formulation to Deployment**

    Question:

    - Outline the key steps in going from a real-world problem statement (e.g., "optimize marketing budget allocation") to deploying an optimization solution (formulation, selection of method, software implementation, results interpretation).

- Identify at least two pitfalls that might happen in the process (e.g., mis-specified objective, overlooked constraints, data quality issues).
- Discuss how you would involve domain experts or stakeholders in iterating on the optimization model to ensure it is both accurate and implementable.