GMRIT Training Tomorrow's Engineers Today	**GMR Institute of Technology** **Rajam, AP** (An Autonomous Institution Affiliated to JNTUGV, AP)	**GMRIT/ADM/F-44** **REV.: 00**

Cohesive Teaching – Learning Practices (CTLP)

Class	6th Sem. - B. Tech.	Department: CSE(AI&DS)	
Course	Optimization techniques for machine learning	**Course Code**	21DS601
Prepared by	Dr.T.Daniya, Sr. Assistant Professor, CSE(AI&ML)		
Lecture Topic	Introduction to Optimization: Introduction to optimization problems and applications in machine learning, Convexity, convex functions, and convex optimization		
Course Outcome (s)	CO1	**Program Outcome (s)**	PO_1, PO_2, PO_4, PO_5, PO_{12}
Duration	5 Hours	**Lecture**	1-5	Unit I
Pre-requisite (s)	Machine Learning & Deep Learning Concepts		

1. Objective

- Develop a foundational understanding of optimization principles and their role in machine learning.
- Gain the ability to apply optimization techniques to solve practical machine learning problems.

2. Intended Learning Outcomes (ILOs)

At the end of this session the students will able to:

A. Analyze Optimization Problems.
B. Apply Optimization Algorithms
C. Evaluate Optimization Solutions in Machine Learning Models.

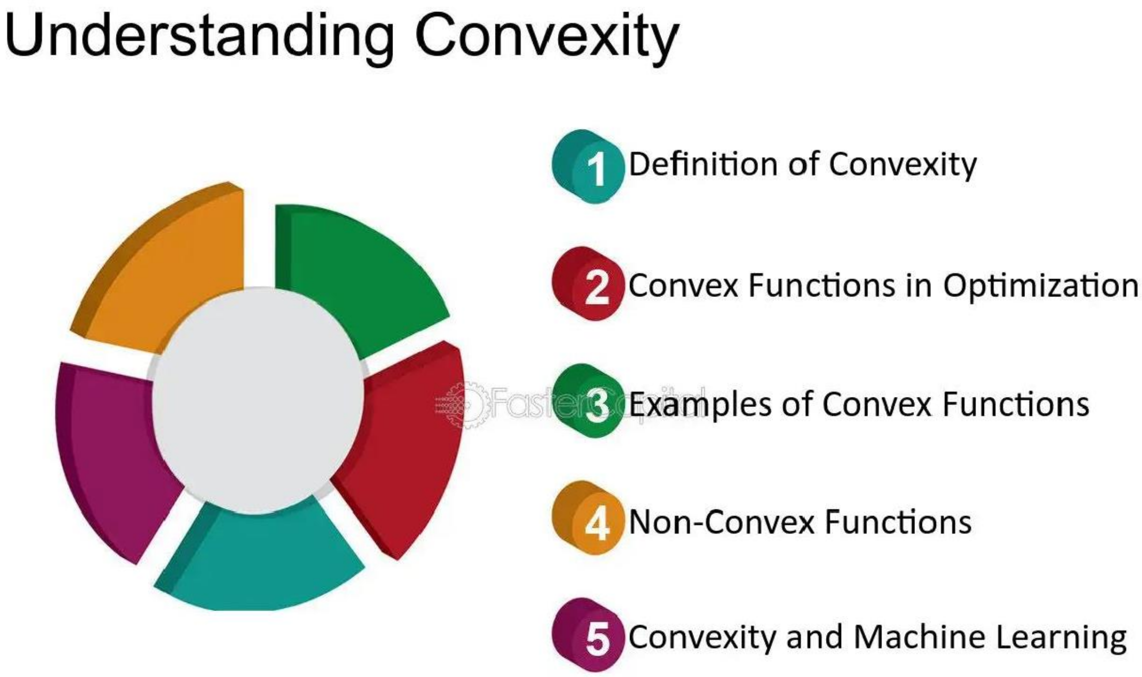
3. 2D Mapping of ILOs with Knowledge Dimension and Cognitive Learning Levels of RBT

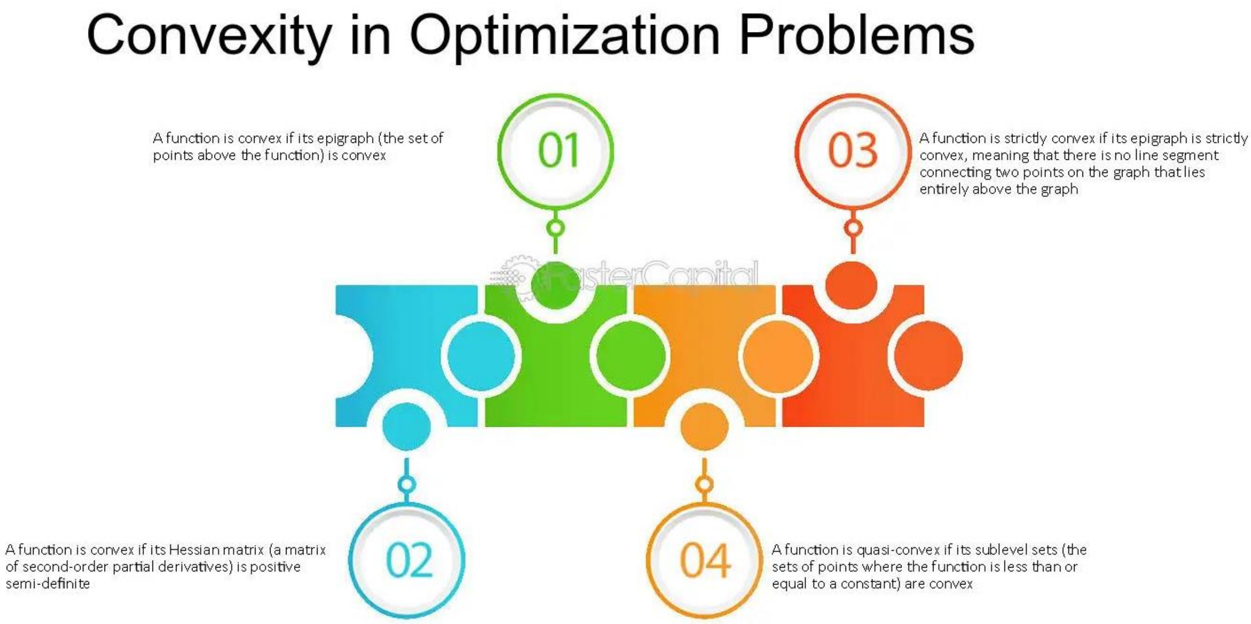
	Cognitive Learning Levels (2D)					
Knowledge Dimension (1D)	Remember	Understand	Apply	Analyze	Evaluate	Create
Factual		**A**	**B**		**C**	
Conceptual						
Procedural						
Meta Cognitive						

4. Teaching Methodology

- Chalk & Board, Visual Presentatio

5. Evocation


Understanding Convexity
1 Definition of Convexity
2 Convex Functions in Optimization
3 Examples of Convex Functions
4 Non-Convex Functions
5 Convexity and Machine Learning


Convexity in Optimization Problems
01
A function is convex if its epigraph (the set of points above the function) is convex
02
A function is convex if its Hessian matrix (a matrix of second-order partial derivatives) is positive semi-definite
03
A function is strictly convex if its epigraph is strictly convex, meaning that there is no line segment connecting two points on the graph that lies entirely above the graph
04
A function is quasi-convex if its sublevel sets (the sets of points where the function is less than or equal to a constant) are convex

6. Deliverables

Lecture – 1 – Introduction to optimization problems

Optimization in Machine Learning: Definition

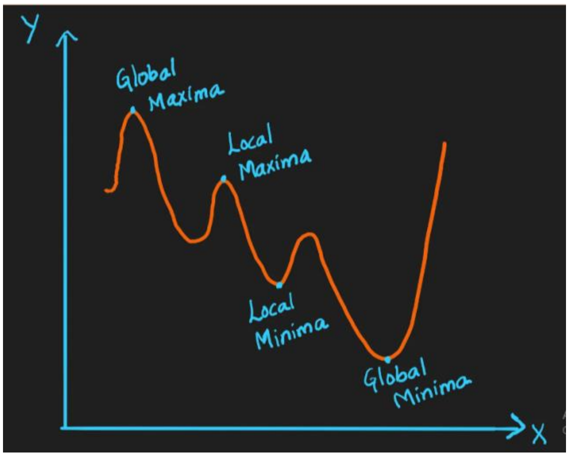
In machine learning, optimization problems are fundamental. They form the core of the training process, **where the goal is to find the best possible set of parameters (weights, coefficients) for your model to achieve the desired outcome**. This translates to mathematically finding the minimum or maximum value of a function, known as the **objective function**.

Why do we optimize our machine learning models?

We compare the results in every iteration by changing the hyperparameters in each step until we reach the optimum results. We create an accurate model with less error rate. There are different ways using which we can optimize a model.

MAXIMA AND MINIMA

Maxima is the largest and Minima is the smallest value of a function within a given range. We represent them as below:



Global Maxima and Minima: It is the maximum value and minimum value respectively on the entire domain of the function

Local Maxima and Minima: It is the maximum value and minimum value respectively of the function within a given range.

There can be only one global minima and maxima but there can be more than one local minima and maxima.

Key Components:

- **Objective Function (Loss Function or Cost Function):** This function quantifies how well your model is performing on a given task. Common examples include mean squared error (MSE) for regression problems and binary cross-entropy for classification problems. The objective function is what you're aiming to optimize (minimize or maximize).
- **Decision Variables:** These are the parameters within your model that you can adjust to influence the model's behavior. For instance, in linear regression, the decision variables are the weights assigned to each feature. By adjusting these weights, you can alter the model's fit to the data.

Types of Optimization Problems:

- **Constrained vs. Unconstrained:**
  Constrained: Here, there are restrictions on the values that the decision variables can take. For example, you might want a weight to be positive or within a specific range.
  Unconstrained: No limitations exist on the decision variables' values(can be positive or negative or fractional etc.). Linear regression is a common example of an unconstrained optimization problem.

- **Convex vs. Non-convex:**
  Convex: The objective function has a single, well-defined minimum (or maximum). This makes optimization algorithms more efficient in finding the optimal solution.
  Non-convex: The objective function may have multiple local minima (or maxima), making it more challenging to find the global optimum (the absolute best solution).

Some popular Optimization Algorithms

Machine learning employs a variety of optimization algorithms to navigate the objective function's landscape and find the optimal set of parameters. Some popular algorithms include:

- **Gradient Descent and its variants:** These algorithms iteratively adjust the decision variables in the direction that minimizes (or maximizes) the objective function. They work by calculating the gradient, which points in the direction of the steepest ascent or descent.

- **Stochastic Gradient Descent (SGD):** A common variant that processes data one example (or minibatch) at a time, making it efficient for large datasets.

- **Momentum and Adagrad:** These are more sophisticated gradient descent variants that address issues like slow convergence or getting stuck in local minima.

Choosing the Right Optimizer: A Balancing Act

The selection of an optimization algorithm depends on several factors, including:

- The type of optimization problem (constrained, unconstrained, convex, non-convex)
- The size and nature of your dataset
- Computational resources available
- The desired level of accuracy

Real-World Examples:

- **Image Recognition:** Optimizing the weights in a convolutional neural network to accurately classify images.
- **Speech Recognition:** Adjusting the parameters in a recurrent neural network to transcribe spoken language.
- **Recommendation Systems:** Optimizing the ranking of items for a user based on their preferences.

Lecture – 2 –Applications in machine learning

Applications of Optimization Problems in Machine Learning

Optimization problems lie at the heart of machine learning, driving the training process and enabling models to make accurate predictions or decisions. Here's a deeper look at how optimization is applied in various machine learning domains:

1. Supervised Learning:

- **Regression:** In linear regression, the objective function (e.g., mean squared error) measures the difference between the model's predictions and the actual target values. The optimization algorithm (like gradient descent) adjusts the weights of the linear model to minimize this error, leading to a better fit for the training data.

- **Classification:** Here, the objective function might be binary cross-entropy for binary classification or multi-class cross-entropy for problems with multiple categories. The optimization process tunes the model's parameters to maximize the likelihood of correctly classifying data points.

2. Unsupervised Learning:

- **Clustering:** K-means clustering aims to minimize the within-cluster variance. The optimization algorithm iteratively adjusts cluster centroids to group data points that are similar to each other, minimizing the overall distance between points and their assigned cluster centers.
- **Dimensionality Reduction:** Techniques like Principal Component Analysis (PCA) use optimization to find the most informative directions (principal components) in the data. This reduces the dimensionality while preserving maximum variance, leading to more efficient data storage and analysis.

3. Deep Learning:

- **Deep Neural Networks (DNNs):** Training a DNN involves optimizing millions of weights across its layers. The objective function (often the loss function) measures the model's performance on a training dataset. Optimization algorithms like stochastic gradient descent (SGD) repeatedly adjust these weights to minimize the loss and improve the model's ability to learn complex patterns.

- **Convolutional Neural Networks (CNNs):** In CNNs used for image recognition, the objective function might be categorical cross-entropy. The optimization process adjusts the filter weights and biases within the convolutional layers to extract relevant features from images and accurately classify them.

4. **Reinforcement Learning:**

- **Policy Learning:** Reinforcement learning agents interact with an environment, receiving rewards for desired actions. The objective function might be to maximize the expected cumulative reward over the long term. Optimization algorithms like Q-learning and policy gradient methods help the agent learn a policy (mapping from states to actions) that optimizes this objective function, leading to optimal decision-making within the environment.

Beyond Core Learning:

Optimization problems extend beyond the core learning tasks in machine learning. They are also crucial for:

- **Hyperparameter Tuning:** Finding the best hyperparameter values (e.g., learning rate, number of hidden layers) for a machine learning model can significantly impact its performance. Optimization techniques like grid search or random search can be used to explore different hyperparameter combinations and identify the set that yields the best results.
- **Model Selection:** In scenarios where you have multiple machine learning models, optimization can help choose the best model for a specific task. This might involve comparing the performance of different models on a validation dataset and selecting the one that minimizes the loss or maximizes accuracy.

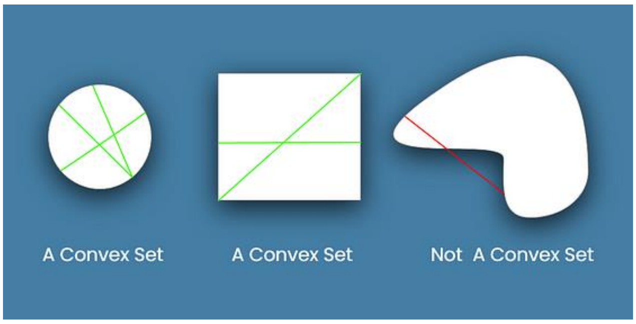
Lecture –3 – Convexity

What is Convexity?

In optimization, convexity refers to the **curvature** of the function you're trying to optimize (minimize or maximize). This function, often called the **objective function** or **loss function**, represents how well your machine learning model is performing.

Convex Sets

Think of convex sets as shapes where any line joining 2 points in this set is never outside the set. This is called a convex set.



It is evident that any line joining 2 points on a circle or say a square (the shapes on extreme left and middle), will have all the line segments within the shape. These are examples of convex sets.

On the other hand, the shape on the extreme right in the figure above has part of a line outside the shape. Thus, this is not a convex set.

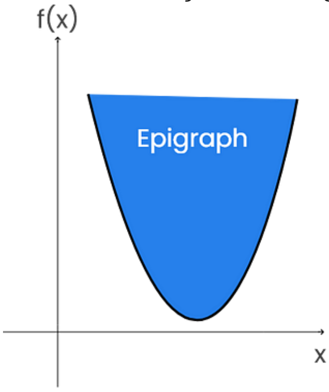
Imagine the objective function as a landscape with hills and valleys. In a convex optimization problem:

- The landscape has a single, well-defined **minimum** (if you're minimizing) or **maximum** (if you're maximizing).
- As you move in any direction from this minimum/maximum, the function's value increases (for minimization) or decreases (for maximization). There are no hidden dips or bumps that could trap you in a suboptimal solution.

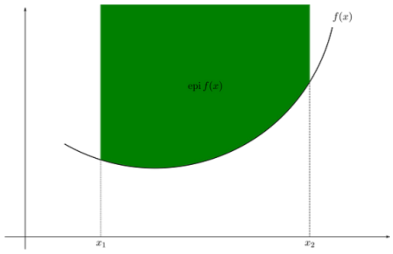
Epigraph

Consider the graph of a function f.

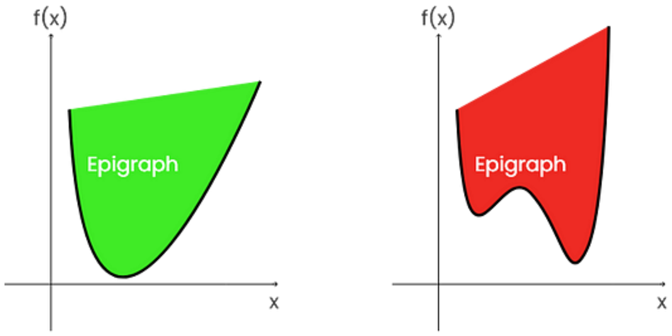
An epigraph is a set of points lying on or above the function's graph.



Lecture-4 Convex Function



A function f is said to be a convex function if its epigraph is a convex set (as seen in the green figure below on the left).
This means that every line segment drawn on this graph is always equal to or above the function graph. Pause a minute and check for yourself.



A convex function   Not a convex function

This means that a function f is not convex if there exist two points x, y such that the line segment joining f(x) and f(y), is below the curve of the function f. This causes the loss of convexity of the epigraph (as seen in the red-figure above on the right).
This means that every line segment drawn on this graph is not always equal to or above the function graph. The same can be proven by taking points on the bends.

Testing for convexity

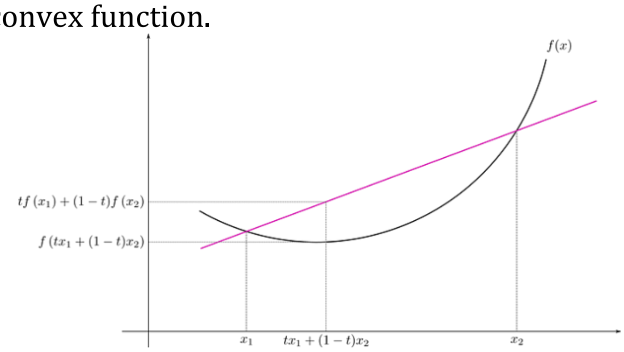
Most of the cost functions in the case of neural networks would be non-convex. Thus you must test a function for convexity.

A function f is said to be a convex function if the second-order derivative of that function is greater than or equal to 0.

Condition for convex functions.

$$f''(x) \geq 0$$

Examples of convex functions: $y=e^x$, $y=x^2$. Both of these functions are differentiable twice.
Below figure represents convex function.

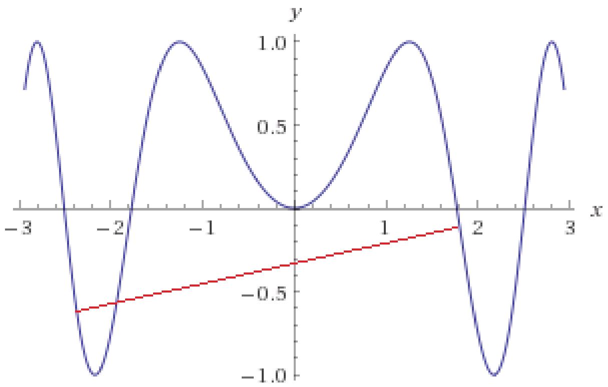


If -f(x) (minus f(x)) is a convex function, then the function is called a concave function.

Condition for concave functions.

$$-f''(x) \geq 0$$

Examples of concave functions: $y=-e^x$. The function is differentiable twice.
Below figure represents non-convex function.



Benefits of Convexity:

- **Easier Optimization:** Convex functions are generally much easier to optimize. Many efficient optimization algorithms are guaranteed to find the global minimum (or maximum) for a convex function. This makes the training process of your machine learning model more reliable and predictable.

- **Theoretical Guarantees:** With convexity, you have mathematical guarantees that the solution you find is the absolute best (globally optimal) solution. This is crucial for ensuring the best possible performance of your model.

Non-Convexity and Its Challenges:

- **Multiple Minima/Maxima:** Non-convex functions can have multiple local minima (or maxima). This means the optimization algorithm might get stuck in a suboptimal solution that isn't the absolute best. It's like finding a valley in the landscape but not necessarily the lowest one.
- **Finding the Global Optimum:** For non-convex functions, there's no guarantee that the optimization algorithm will find the global minimum/maximum. The outcome can depend on the initialization of the algorithm and the specific problem structure.

Real-World Examples:

- **Linear Regression:** This is a classic example of a convex optimization problem. The mean squared error loss function used in linear regression is convex, ensuring a single minimum that represents the best fit for the data.
- **Logistic Regression:** The binary cross-entropy loss function used in logistic regression is also convex, leading to a well-defined optimal solution for classifying data points.
- **Deep Neural Networks (DNNs):** The loss functions used in training DNNs can be non-convex, especially with complex architectures and large datasets. This makes it more

challenging to guarantee that the training process finds the absolute best solution for the model.

Note: *A saddle point is a point in the parameter space where the loss function has a minimum value in one direction and a maximum value in another direction. At a saddle point, the gradient of the loss function is zero, which means that the optimization algorithm may get stuck and not be able to converge to the global minimum.*

Lecture 5: Convex Optimization

What is Convex Optimization in Machine Learning?

The ideal model parameters that minimize the loss function are found using convex optimization, a mathematical optimization technique. A model that can generalize to new data is what machine learning seeks to learn from data. By minimizing a loss function, which gauges the discrepancy between expected and actual output, the model's parameters are discovered. Typically, the optimization problem is represented as a convex optimization problem with linear constraints and a convex objective function.

Convex optimization is well suited for machine learning because it has several advantages, such as convergence guarantees, efficient techniques, and robustness. Gradient descent, a well-liked optimization method in machine learning, is built on convex optimization. Gradient descent is used to update the parameters in the direction of the negative gradient of the objective function. The learning rate determines the size of each iteration's step. Gradient descent will consistently find the optimal solution if the learning rate is sufficiently low, and the objective function is convex.

Newton's method, interior point methods, and stochastic gradient descent are some more convex optimization-based optimization techniques. The trade-offs between convergence speed and computing complexity in these algorithms differ.

Theoretical Guarantees: Convex optimization provides a strong theoretical foundation for understanding the optimization process and the properties of the solutions obtained.

Examples of Convex Optimization in Machine Learning:

- **Linear Regression**: Finding the optimal weights for a linear regression model involves minimizing the mean squared error, which is a convex function.
- **Logistic Regression**: The binary cross-entropy loss function used in logistic regression is also convex, leading to a well-defined optimal solution for classifying data points.
- **Support Vector Machines (SVMs) with linear kernels**: The optimization problem in linear SVMs involves maximizing the margin, which can be formulated as a convex optimization problem.

Challenges and Considerations:

- **Non-convex Problems:** Many real-world machine learning problems involve non-convex objective functions. This can occur due to factors like complex model architectures or certain loss functions. In these cases, optimization algorithms may get stuck in local minima, and finding the global optimum becomes more challenging.
- **Choosing the Right Algorithm:** Depending on the specific problem and the properties of the objective function and constraints, different optimization algorithms might be more suitable. Some common algorithms include gradient descent, interior-point methods, and stochastic gradient descent variants.

7. Keywords

- ❖ optimization problem
- ❖ objective function
- ❖ constraints
- ❖ convexity
- ❖ convex optimization
- ❖ machine learning
- ❖ loss function
- ❖ gradient descent
- ❖ algorithms
- ❖ convergence

8. Sample Questions

Remember

1. Define "optimization" in the context of mathematics and machine learning.
2. What is the difference between an objective function and a constraint in an optimization problem?
3. Explain what a convex function is. Provide an example.
4. Why is convexity important in optimization?
5. Name at least two common optimization algorithms used in machine learning.
6. What is the purpose of a loss function in machine learning?
7. Briefly describe how gradient descent works.
8. What does the term "convergence" mean in the context of optimization?
9. State whether the following is True or False: All machine learning problems are convex optimization problems.
10. List two application areas of convex optimization.

Understanding

1. Explain how a machine learning model's performance might change depending on whether the optimization problem is convex or non-convex.
2. Describe a scenario where a machine learning problem might have constraints. Provide a specific example.
3. In simple terms, explain the goal of gradient descent and how it relates to minimizing a loss function.
4. How would you determine if a particular function is convex or not?
5. Consider a linear regression problem. Sketch a possible loss function and explain how gradient descent would find the optimal solution.
6. Why might different optimization algorithms be better suited to different machine learning problems?
7. What might happen if the optimization process in a machine learning model does not converge?
8. How can you tell if a machine learning model might be overfitting the data? How does this relate to optimization?
9. Explain the concept of regularization in the context of optimization.
10. In your own words, summarize the connection between optimization and machine learning.

Apply

1. Given a simple dataset, formulate a linear regression problem as a convex optimization problem. Identify the objective function, decision variables, and any potential constraints.
2. You are training a machine learning classifier and notice the loss function is not decreasing with successive iterations. Suggest potential reasons why this might happen and propose steps to troubleshoot.
3. Select a real-world dataset (e.g., housing prices, image classification). Describe how you would structure a machine learning task on this data as an optimization problem.
4. You are given two optimization algorithms: gradient descent and Newton's method. Discuss the potential advantages and drawbacks of each in the context of training a neural network.

Analyze

1. Analyze the structure of a support vector machine (SVM) optimization problem. How does the concept of a margin relate to the objective function and constraints?
2. Compare and contrast gradient descent, stochastic gradient descent, and Adam optimizers. Discuss their strengths, weaknesses, and suitability for different types of problems.
3. Examine a graph of a loss function over iterations during a machine learning training process. Identify patterns or anomalies that might suggest issues with convergence, overfitting, or underfitting.

9. Stimulating Question (s)

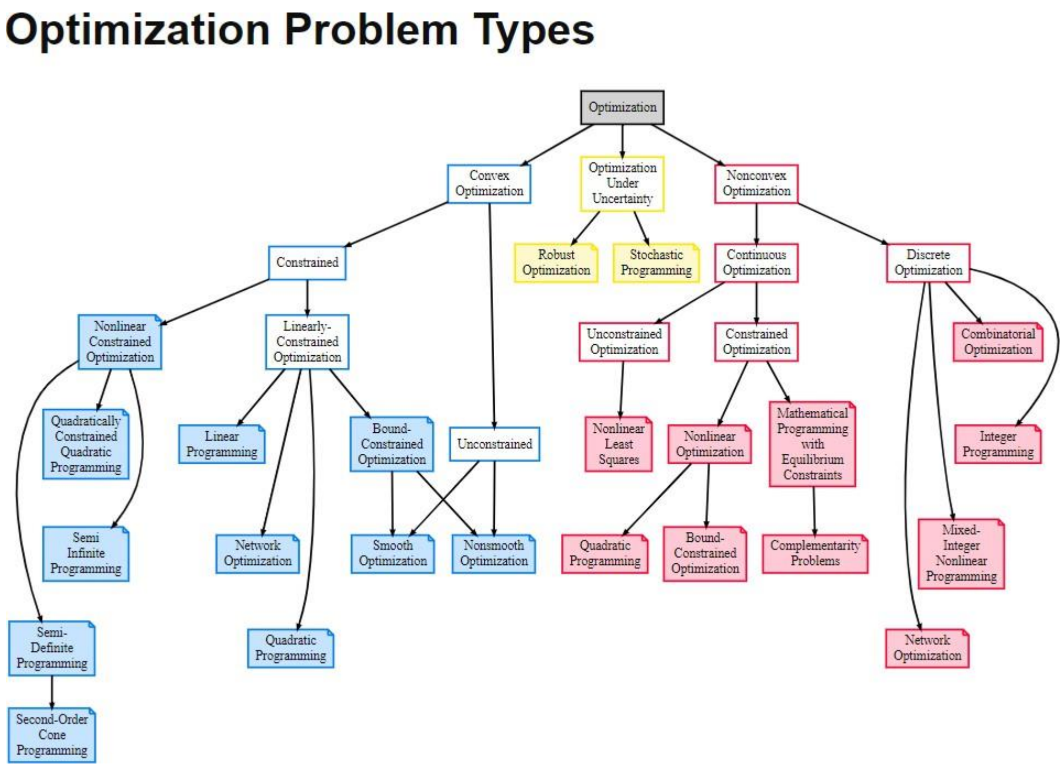
1. The trade-off between optimality and practicality:

- Many real-world optimization problems are incredibly complex or computationally expensive to solve perfectly. How do machine learning practitioners balance the desire for the best possible model with the need for solutions that can be obtained within reasonable time and resource limitations? Can you provide real-world examples to illustrate this trade-off?

2. The limitations of optimization in the face of uncertainty:

- Optimization often assumes perfect knowledge of the problem and data. In the real world, data is noisy and systems can change over time. How can machine learning models be made more robust to uncertainty, and how do the limitations of optimization influence model design and deployment?

10. Mind Map



11. Student Summary

At the end of this session, the facilitator (Teacher) shall randomly pic-up few students to summarize the deliverables

12. Reading Materials

“1. "Convex Optimization" by Stephen Boyd and Lieven Vandenberghe.
2. "Optimization for Machine Learning" by Suvrit Sra, Sebastian Nowozin, and Stephen Wright.
3. "Numerical Optimization" by Jorge Nocedal and Stephen J. Wright.

13. Scope for Mini Project

Following are some of the Mini project ideas.

Project 1: Optimization Algorithm Comparison

- **Goal:** Investigate the performance of different optimization algorithms on various machine learning problems.
  Steps:
1. Select a simple dataset (e.g., Iris dataset for classification or a dataset suitable for linear regression).
2. Choose several optimization algorithms (e.g., gradient descent, stochastic gradient descent, Adam).
3. Implement these algorithms or use libraries like SciPy (for Python).
4. Train models using each algorithm, tracking metrics like accuracy, loss, and training time.
5. Analyze and compare the results, discussing which algorithms perform best under different conditions.

- **Skills:** Algorithm implementation, machine learning basics, data analysis, visualization.

Project 2: Visualizing Convex Optimization

- **Goal:** Create interactive visualizations to illustrate the concepts of convexity and gradient descent.
- **Steps:**
1. Choose a programming language suitable for visualization (e.g., Python with Matplotlib or JavaScript with D3.js).
2. Design visualizations demonstrating:
   - The difference between convex and non-convex functions.
   - How gradient descent iteratively finds the minimum of a convex function.
3. Allow users to potentially adjust parameters (learning rate, starting point) and see the effect on the optimization process.

- **Skills:** Programming, visualization, understanding of convex optimization concepts.