Optimization Theory and Methods

Compiled Notes

November 29, 2025

Contents

Chapter 1

Introduction to Optimization

1.1 Course Context and Motivation

Optimization is fundamental across various fields, including Machine Learning (ML). This course emphasizes **algorithmic aspects** and **implementation** in dynamic contexts, especially regarding Large Language Models (LLMs).

1.1.1 Two Core Perspectives of Optimization

Optimization in ML can be viewed through two complementary lenses:

(1) **Mathematical Modeling:** Formulating problems (e.g., fitting data) using objectives and constraints.

(2) **Computational Optimization:** Focusing on the efficiency and properties of the algorithms used to solve the problems (e.g., convergence speed).

1.1.2 Importance and Necessity

The study of optimization remains critical due to:

- **Computational Efficiency:** Making large models (like LLMs) more efficient, reducing GPU consumption, and innovating parameter learning.
- **Operational Deployment:** Enabling models to operate in **frugal environments** (e.g., Edge Computing) with limited computational resources.
- **Informed Decision-Making:** Providing the knowledge base to make informed choices about objectives and algorithms, pushing the boundaries of research.

1.2 Optimization in Supervised Learning: Regression

The most famous example is **regression**, which involves learning weights ($\mathbf{w}$) to approximate an output $\mathbf{y}$ based on input features $\phi(\mathbf{x})$.

1.2.1 The General Model

The output y is approximated as a linear combination of features $\phi(\mathbf{x})$ and associated weights $\mathbf{w}$:

$$y \approx \phi(\mathbf{x})^T \mathbf{w}$$

Where $\phi(\mathbf{x})$ (the feature map) and $\mathbf{w}$ (the weights) are column vectors. In this context, $\phi(\mathbf{x})$ is treated as a known set of functions.

1.2.2 Regularized Least Squares Problem

The goal is to minimize the error over a set of observations $(\mathbf{x}_i, y_i)$, often incorporating a regularizer:

Problem 1.1 (Regularized Least Squares)**.** The optimization problem is defined as learning the weights $\mathbf{w}$ by minimizing the sum of squared errors plus an L2 regularizer:

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \left(y_i - \phi(\mathbf{x}_i)^T \mathbf{w}\right)^2 + \lambda ||\mathbf{w}||_2^2$$

Where:

- The first term is the **Loss Function** (Sum of Squares of Errors).
- The second term, $\lambda ||\mathbf{w}||_2^2$, is the **Regularizer**.

1.2.3 Rationale for Regularization

The regularizer term $\lambda ||\mathbf{w}||_2^2$ serves multiple roles:

(i) **Prevent Overfitting (Generalization):** It prevents the model from just memorizing the training data, ensuring it can generalize to new inputs $\mathbf{x}_{\text{new}}$.

(ii) **Introducing Prior/Bias:** It is equivalent to imposing a constraint or a **prior** on the weights.

(iii) **Bayesian Equivalence:** L2 regularization ($||\mathbf{w}||_2^2$) is equivalent to placing a **Gaussian prior** on the parameters $\mathbf{w}$. L1 regularization ($||\mathbf{w}||_1$) is equivalent to a **Laplace/Lidstone prior**.

(iv) **Computational Improvement:** From an algorithmic perspective, it promotes desirable properties like **strong convexity**, leading to better convergence (e.g., faster convergence for Gradient Descent).

1.3 Optimization Formulation for Classification

In supervised learning, the model is generally $F_\theta(\mathbf{x})$ (where θ are the parameters/weights). The general objective is to minimize the loss plus a regularizer:

$$\min_{\theta} \mathcal{L}(\theta) + \lambda \Omega(\theta)$$

1.3.1 Classification Losses

Classification losses generally attempt to approximate the ideal, non-differentiable **Step Loss** (0 loss for correct class, 1 for wrong class) with a continuous function.

Loss Function	Characteristics	Common Use
Step Loss	Ideal, Non-differentiable	Theoretical Benchmark
Logistic Loss	Continuous, used to maximize likelihood	Classification (Binary/Multiclass)
SoftMax Loss	Generalization of Logistic Loss	Multiclass Classification
Hinge Loss	Focuses on margin; continuous but non-differentiable at $yF(\mathbf{x}) = 1$	Support Vector Machines (SVMs)

Table 1.1: Common Loss Functions for Classification

1.3.2 Regularizers and Sparsity

The regularizer can also be viewed as a constraint on the magnitude of weights, $\Omega(\theta) \leq \gamma$.

(i) **L2 Regularization ($||\mathbf{w}||_2^2$):** $\Omega(\mathbf{w}) = \sum w_i^2$. The constraint set ($\Omega(\mathbf{w}) \leq \gamma$) is a **sphere**.

(ii) **L1 Regularization (Lasso):** $\Omega(\mathbf{w}) = \sum |w_i|$. The constraint set is a **diamond/square** (in 2D), forcing optimal solutions to occur at the corners. This leads to **sparse solutions**, crucial for **quantization** and compact models (e.g., for Edge Computing).

Example 1.1 (Closed-Form Solution: Ridge Regression)**.** The L2 Regularized Least Squares problem (Ridge Regression) possesses a **closed-form solution**.

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

The regularization term $\lambda\mathbf{I}$ increases the eigenvalues of the matrix $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$, bringing **stability** to the solution. Most other problems require **iterative algorithms**.

1.4 Flavors of Optimization

Optimization problems in ML fall into three categories:

(1) **Continuous Optimization**: Variables are continuous (e.g., weights $\mathbf{w}$). **Examples:** Least Squares, Neural Networks, LLMs, Convex Optimization. Often viewed as **relaxations** of hard discrete problems.

(2) **Discrete Optimization**: Variables are discrete (e.g., selections). **Examples:** Hyperparameter Optimization, Network Architecture Search, Identifying data subsets. Often used as a **subroutine** to make continuous optimization more effective.

(3) **Mixed Continuous and Discrete Optimization**: Involves both continuous (e.g., location/intensity) and discrete variables (e.g., selection/membership). **Example: K-Means Clustering** (locating centroid is continuous; assigning membership is discrete).