Introduction to Optimization in Machine Learning

Course:
INFO-6154 Data Science and Machine Learning

Developed by:
Mohammad Noorchenarboo

May 3, 2025

Current Section

1. Introduction to Optimization in Machine Learning
   - What Is Optimization in ML?
   - Key Components of Optimization
   - Types of Problems Optimization Solves in ML
   - Practical Failures of Optimization
   - Real-World Problem Framing and Optimization
   - Examples and Mini Case Studies
   - Takeaways: Setting the Right Optimization Mindset

What Is Optimization in ML?

Optimization is the mathematical process of adjusting model parameters to minimize a specific objective function, known as the **loss function**. In machine learning, this is how models *learn from data*.

Gradient Descent Flow (from scratch)

```
# Loss function: f(x) = (x - 3)^2
x = 0                          # Initial guess
lr = 0.1                       # Learning rate
for i in range(10):
    grad = 2 * (x - 3)         # Derivative of f(x)
    x -= lr * grad             # Update step
```

What Is Optimization in ML?

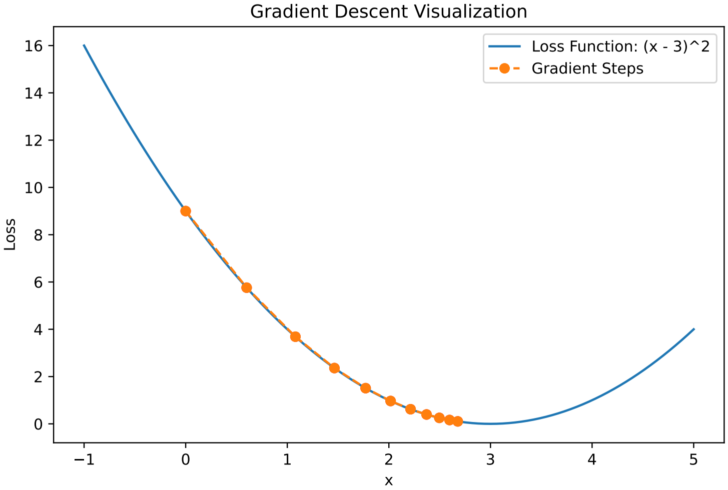
Definition

In the context of machine learning, **optimization** refers to the process of finding the best model parameters that minimize (or maximize) a predefined loss function on the training data.

Why is it essential for model training?

- Without optimization, models cannot adjust to learn from data.
- All training procedures – from linear regression to deep neural networks – rely on optimization algorithms to adjust weights.
- The quality of the optimization directly affects model performance.

What Is Optimization in ML?



What Is Optimization in ML?

Insight

Optimization is not a post-processing step – it *is* the core of the learning process itself.

Optimization vs. Inference

- **Optimization** happens during training: finding the best parameters.
- **Inference** happens during testing/deployment: using the learned parameters to make predictions.
- The optimization phase enables inference to work properly.

Note

Do not confuse model *training* (which involves optimization) with *using* the model (inference).

Key Components of Optimization

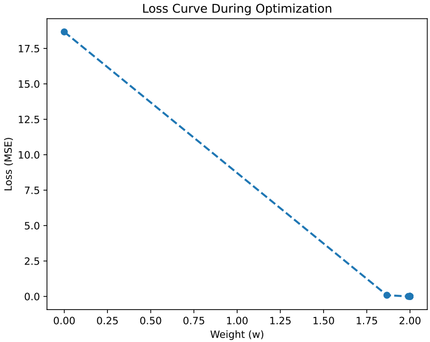
1. **Loss Function**
   - Measures how far predictions are from the true values.
   - Defines the objective to minimize.
   - Common examples: Mean Squared Error (MSE), Cross-Entropy Loss.
2. **Model Parameters**
   - The internal values (like weights) that the model adjusts during training.
   - These are what optimization seeks to improve.
3. **Optimizer**
   - The algorithm that updates model parameters based on the loss.
   - Examples: Gradient Descent, Adam, RMSProp.
4. **Training Data**
   - The source of feedback for optimization.
   - Includes features X and true labels Y.

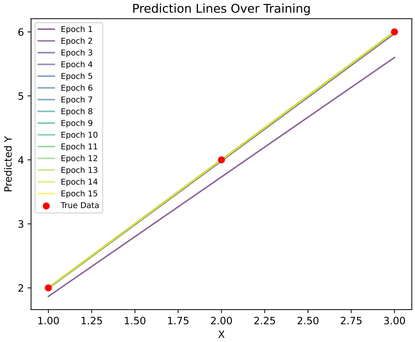
Key Components of Optimization

All Key Components in One View

```python
# 1. Training Data
X = [1, 2, 3]
Y = [2, 4, 6]  # True values

# 2. Parameters (initialize weight)
w = 0.0

# 3. Loss Function: MSE
def loss(w):
    return sum((w*x - y)**2 for x, y in zip(X, Y)) / len(X)

# 4. Optimizer: Gradient Descent
lr = 0.1
for _ in range(10):
    grad = sum(2 * x * (w*x - y) for x, y in zip(X, Y)) /
        len(X)
    w = w - lr * grad
```

Key Components of Optimization





Key Components of Optimization

Real-World Example: AI for Medical Diagnosis

- **Loss Function** – Penalizes incorrect diagnosis (e.g., missing a cancer case).
- **Parameters** – Weights in the neural network that analyze CT scans.
- **Optimizer** – Algorithm adjusting weights to reduce false negatives.
- **Training Data** – Thousands of labeled scans (e.g., benign vs. malignant).

Insight

Well-optimized AI models in healthcare reduce diagnostic errors and improve early detection rates.

Types of Problems Optimization Solves in ML

Optimization in machine learning is not limited to one kind of task – it underpins a wide variety of problems by defining and minimizing suitable loss functions.

1. Classification

- Predicting discrete categories (e.g., spam vs. not spam, positive vs. negative).
- **Common Loss**: Cross-Entropy Loss.
- **Example**: Classifying skin lesions as benign or malignant from dermoscopic images.

2. Regression

- Predicting continuous values.
- **Common Loss**: Mean Squared Error (MSE).
- **Example**: Predicting house prices or ICU patient length-of-stay.

Types of Problems Optimization Solves in ML

Comparing Loss Types in Classification

```
# Predicted probabilities from sigmoid
preds = 1 / (1 + np.exp(-w * X))

# Mean Squared Error
mse = np.mean((preds - labels)**2)

# Cross-Entropy Loss
ce = -np.mean(labels * np.log(preds) +
              (1 - labels) * np.log(1 - preds))
```

Warning

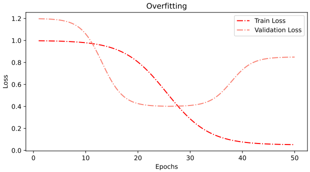
Applying the wrong loss type (e.g., using regression loss for a classification task) can cause the optimizer to converge on meaningless or harmful solutions.

Practical Failures of Optimization

Even with well-designed models and datasets, optimization can fail in subtle or catastrophic ways. Understanding these pitfalls is crucial for diagnosing poor model performance.

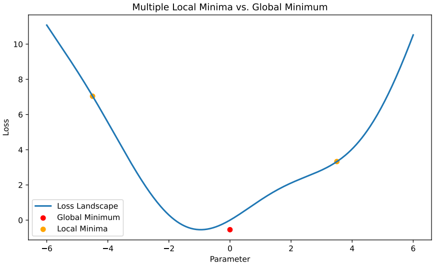
1. Non-Convergence

- The optimizer fails to reduce the loss meaningfully.
- Can be caused by a poor learning rate, bad initialization, or vanishing/exploding gradients.
- **Example**: Deep CNN on X-ray images not improving due to poor learning rate scheduling.

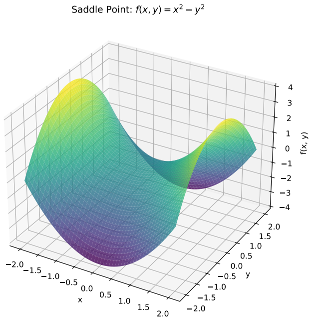


2. Stuck in Local Minima or Saddle Points

- The model reaches a point where gradients are small but performance is poor.
- Especially problematic in non-convex functions like those in neural networks.



Local Minima vs Global Minimum

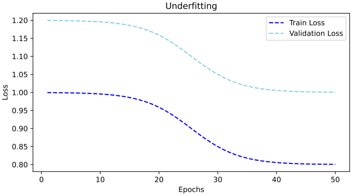


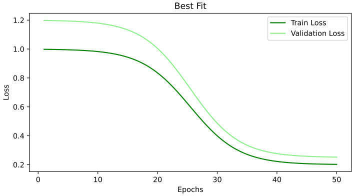
Saddle Point Landscape

Practical Failures of Optimization

3. Overfitting

- The optimizer fits the training data too well, harming generalization.
- Caused by overtraining or lack of regularization.
- **Example**: A fraud detection model that performs perfectly on historical data but fails in deployment.

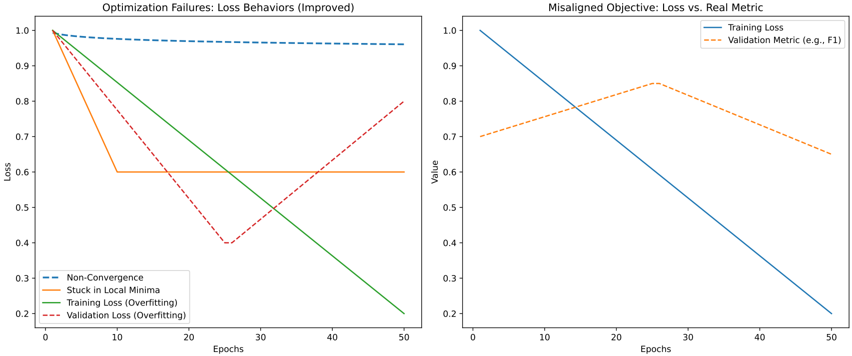




Practical Failures of Optimization

4. Misaligned Objectives

- The loss being minimized doesn't match the real-world goal.
- **Example**: Optimizing for accuracy in a highly imbalanced cancer screening dataset.
- Real goal may be sensitivity or F1 score, not raw accuracy.



Practical Failures of Optimization

Key Pitfalls to Watch For

- Loss decreases, but validation performance is poor.
- Model performs well on paper but fails in production.
- Training behaves erratically with large gradients or exploding loss.

Tip

Always monitor both training and validation metrics. Use visualizations like loss curves to catch optimization issues early.

Real-World Problem Framing and Optimization

Many machine learning projects fail not because of poor models – but because the problem was *framed incorrectly* from the start.

Why framing matters:

- The way we define the prediction target, features, and loss function shapes the entire optimization process.
- A misaligned problem frame leads to optimizing the wrong thing.

Common Mistakes in Problem Framing

- Predicting a variable that cannot be reliably measured.
- Using poor proxies (e.g., clicks instead of satisfaction).
- Treating subjective or noisy data as ground truth.
- Ignoring domain-specific constraints (e.g., false positives in healthcare).

Real-World Problem Framing and Optimization

Takeaway

Bad framing leads to bad labels, bad loss functions, and ultimately – bad models.

Role of Domain Expertise

- Helps define meaningful prediction targets.
- Guides which features to include or exclude.
- Informs what types of errors are tolerable or critical.

Best Practice

Start every ML project with a clearly defined real-world objective and discuss how it will be reflected in your loss function and data structure.

Examples and Mini Case Studies

Let's explore how optimization decisions play out in real-world projects – especially when the loss function does not align well with actual goals.

Case Study 1: Customer Churn Prediction

- **Business Goal**: Retain high-value customers to reduce revenue loss.
- **Naive ML Goal**: Predict if a customer will churn (binary classification).
- **Common Pitfall**: Optimizing for log-loss or AUC, regardless of customer value.
- **Better Framing**: Weight the loss function by customer lifetime value (CLV).

Optimization Adjustment

Instead of treating all churns equally, the loss could penalize errors on high-value customers more heavily.

Examples and Mini Case Studies

Case Study 2: Fraud Detection

- **Business Goal**: Catch fraudulent transactions while minimizing friction for real users.
- **Naive ML Goal**: Minimize classification error.
- **Challenge**: Fraud cases are rare and false positives harm user experience.
- **Better Framing**: Optimize for precision-recall balance or F1 score. Use cost-sensitive loss if available.

Examples and Mini Case Studies

Impact

Aligning the optimization objective with business priorities significantly improves model utility and adoption.

Tip

Don't rely solely on off-the-shelf loss functions. Customize your optimization objective to match the problem's real-world context.

Takeaways: Setting the Right Optimization Mindset

Understanding optimization is more than just knowing algorithms – it's about having the right mindset when designing and training models.

Key Conceptual Shifts

- **Loss** is what the model optimizes – not necessarily what we care about.
- **Metric** is what we report – it may not guide training unless we make it part of the loss.
- **Optimization** is not interpretation – the most accurate model may be the least explainable.

Warning

A model learns what you tell it to optimize – not what you wish it would learn.

Takeaways: Setting the Right Optimization Mindset

Good Optimization Practice Starts With:

- Framing the problem in real-world terms.
- Choosing a loss function that reflects your goal.
- Anticipating practical failures during training.

Closing Thought

Before you optimize a model, make sure you’re solving the right problem – with the right signal.