

# Introduction to Optimization in Machine Learning

Course:  
INFO-6154 Data Science and Machine Learning



Developed by:  
Mohammad Noorchenarboo

May 12, 2025

# Current Section

- 1 What Is Optimization in ML?
- 2 Key Components of Optimization
- 3 Practical Failures of Optimization
- 4 Real-World Problem Framing and Optimization
- 5 Examples and Mini Case Studies
- 6 Takeaways: Setting the Right Optimization Mindset
- 7 Hands-On Optimization Challenge

# 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
```

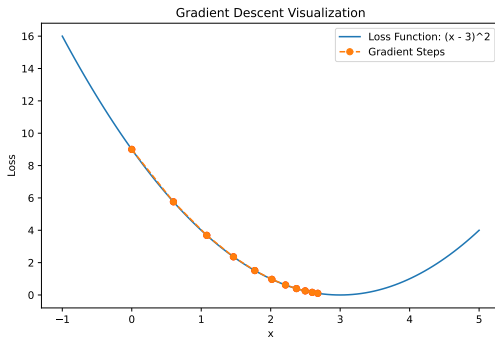
## Definition

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.

# What Is Optimization in ML?

## 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?

## 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).

# Summary: What Is Optimization in ML?

Concept	Explanation
Optimization	Process of minimizing a loss function by adjusting model parameters
Loss Function	Quantifies the model's prediction error during training
Gradient Descent	Common algorithm to iteratively minimize the loss
Inference vs. Optimization	Optimization happens during training; inference occurs at deployment

# Current Section

- 1 What Is Optimization in ML?
- 2 Key Components of Optimization**
- 3 Practical Failures of Optimization
- 4 Real-World Problem Framing and Optimization
- 5 Examples and Mini Case Studies
- 6 Takeaways: Setting the Right Optimization Mindset
- 7 Hands-On Optimization Challenge

# 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

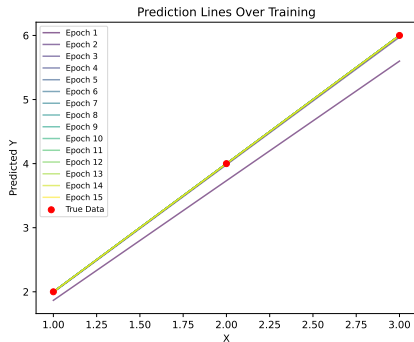
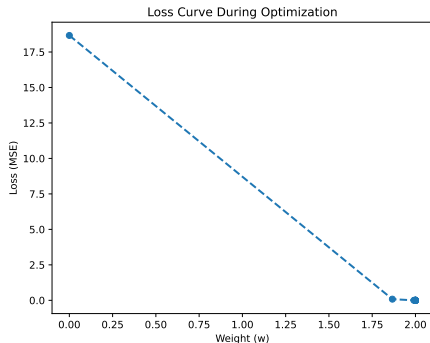
```
# 1. Training Data
X = [1, 2, 3]
Y = [2, 4, 6] # True values

# 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.

# Summary: Key Components of Optimization

Component	Role in Optimization
Loss Function	Measures prediction error; goal is to minimize it
Model Parameters	Values (like weights) updated during optimization
Optimizer	Algorithm that adjusts parameters to reduce loss
Training Data	Provides input-output pairs for learning the function mapping

# Current Section

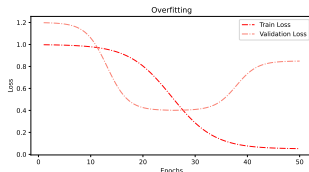
- 1 What Is Optimization in ML?
- 2 Key Components of Optimization
- 3 Practical Failures of Optimization**
- 4 Real-World Problem Framing and Optimization
- 5 Examples and Mini Case Studies
- 6 Takeaways: Setting the Right Optimization Mindset
- 7 Hands-On Optimization Challenge

# 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

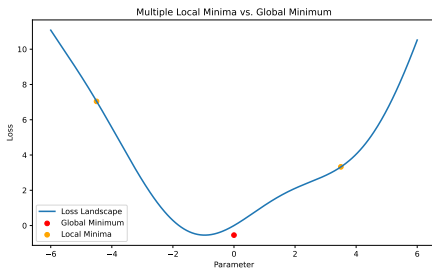
- The optimizer fails to reduce the loss meaningfully.
- Can be caused by a poor learning rate, bad initialization, or vanishing/exploding gradients.
- **Example:** Deep CNN on X-ray images not improving due to poor learning rate scheduling.



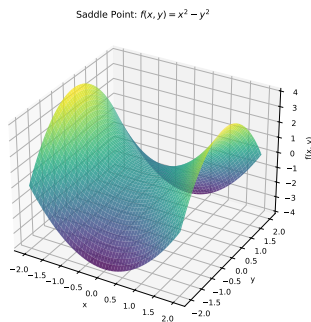
# Practical Failures of Optimization

## 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

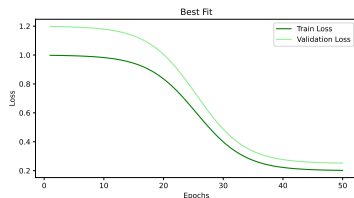
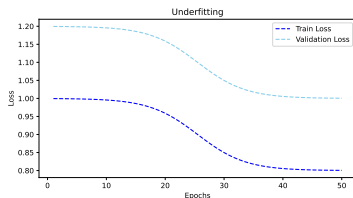


Saddle Point Landscape

# 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.

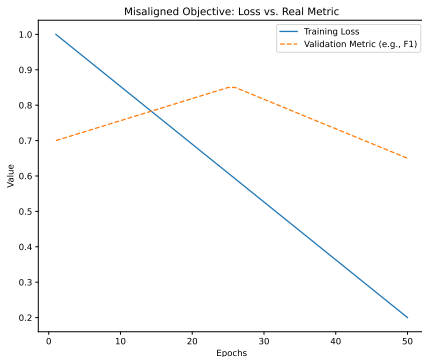
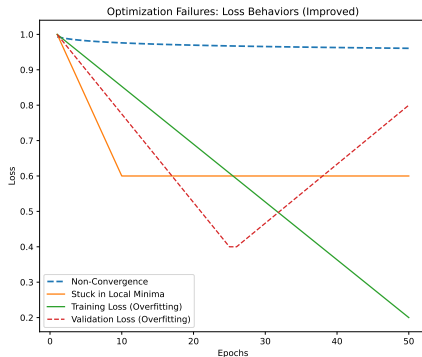




# Practical Failures of Optimization

## 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.

# Summary: Practical Failures of Optimization

Failure Type	Description or Cause
Non-Convergence	Model fails to learn due to learning rate, initialization, or gradient issues
Local Minima or Saddle Points	Gets stuck in suboptimal solutions with flat gradients
Overfitting	Model over-learns training data, leading to poor generalization
Misaligned Objectives	Optimization minimizes the wrong metric for the real-world task

# Current Section

- 1 What Is Optimization in ML?
- 2 Key Components of Optimization
- 3 Practical Failures of Optimization
- 4 Real-World Problem Framing and Optimization**
- 5 Examples and Mini Case Studies
- 6 Takeaways: Setting the Right Optimization Mindset
- 7 Hands-On Optimization Challenge

# 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.

# Summary: Real-World Problem Framing and Optimization

Framing Issue	Impact on Optimization
Poor Target Definition	Leads to meaningless or unmeasurable predictions
Proxy Variables	Misaligns the optimization objective with business goals
Noisy Labels	Reduces optimizer's ability to learn meaningful patterns
Lack of Domain Knowledge	Results in suboptimal feature selection and error tolerance

# Current Section

- 1 What Is Optimization in ML?
- 2 Key Components of Optimization
- 3 Practical Failures of Optimization
- 4 Real-World Problem Framing and Optimization
- 5 Examples and Mini Case Studies**
- 6 Takeaways: Setting the Right Optimization Mindset
- 7 Hands-On Optimization Challenge



# 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.

# Summary: Examples and Mini Case Studies

Case	Optimization Insight
Customer Churn	Loss should reflect business value (e.g., customer lifetime value weighting)
Fraud Detection	Requires cost-sensitive optimization and prioritizing recall or F1 score
General Tip	Custom loss functions often outperform generic ones in real-world settings

# Current Section

- 1 What Is Optimization in ML?
- 2 Key Components of Optimization
- 3 Practical Failures of Optimization
- 4 Real-World Problem Framing and Optimization
- 5 Examples and Mini Case Studies
- 6 Takeaways: Setting the Right Optimization Mindset**
- 7 Hands-On Optimization Challenge

# 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.

# Summary: Setting the Right Optimization Mindset

Concept Shift	Why It Matters
Loss vs. Metric	The loss guides learning; the metric is what we care about
Optimization $\neq$ Interpretation	Highly optimized models may be less explainable
Framing First	Success depends on matching the problem to the optimization signal



# Current Section

- 1 What Is Optimization in ML?
- 2 Key Components of Optimization
- 3 Practical Failures of Optimization
- 4 Real-World Problem Framing and Optimization
- 5 Examples and Mini Case Studies
- 6 Takeaways: Setting the Right Optimization Mindset
- 7 Hands-On Optimization Challenge**

# Guess the Best Line (No Math Yet)

Imagine you are given the following 3 data points:

$$(1,2), \quad (2,3), \quad (3,5)$$

**Your task:** Find the best-fitting line of the form:

$$\hat{y} = w \cdot x + b$$

You do **not** need any formula right now. Just:

- Guess a slope  $w$  and intercept  $b$
- Predict the values  $\hat{y}$
- Compare them to actual values  $y$
- Try to make your guesses better

## Think Like a Machine

Try to make small changes and see if the prediction gets better – you're doing optimization without knowing it.

# Step 1: Try Your First Guess

Choose any values of  $w$  and  $b$ . For example:

$$w = 1, \quad b = 0$$

Then compute predictions:

$$\hat{y}_1 = 1 \cdot 1 + 0 = 1$$

$$\hat{y}_2 = 1 \cdot 2 + 0 = 2$$

$$\hat{y}_3 = 1 \cdot 3 + 0 = 3$$

**True  $y$  values:** 2, 3, 5

**Compare:**

- $\hat{y}_1 = 1$  vs  $y_1 = 2$
- $\hat{y}_2 = 2$  vs  $y_2 = 3$
- $\hat{y}_3 = 3$  vs  $y_3 = 5$

# Step 1: Try Your First Guess

## Clearly Not a Good Fit

Can you find better  $w$  and  $b$ ?

## Step 2: Calculate Errors

Using your predictions:

$$\hat{y} = w \cdot x + b$$

Compute the error:

$$\text{Error}_i = \hat{y}_i - y_i$$

With  $w = 1, b = 0$ , the errors are:

$$\text{Error}_1 = 1 - 2 = -1$$

$$\text{Error}_2 = 2 - 3 = -1$$

$$\text{Error}_3 = 3 - 5 = -2$$

### What Could You Try Next?

Should you increase the slope? Raise the intercept? Try changing both.

## Step 3: Try Again and Track Improvements

Let's try a better guess:

$$w = 1.5, \quad b = 0.2$$

**Predictions:**

$$\hat{y}_1 = 1.5 \cdot 1 + 0.2 = 1.7$$

$$\hat{y}_2 = 1.5 \cdot 2 + 0.2 = 3.2$$

$$\hat{y}_3 = 1.5 \cdot 3 + 0.2 = 4.7$$

**True  $y$ :** 2, 3, 5

**Errors:**

$$-0.3, \quad +0.2, \quad -0.3$$

### Getting Closer

You improved the predictions by adjusting  $w$  and  $b$  – this is optimization!

## Step 4: What Should We Minimize?

### How can we tell which line is better?

Idea: Combine the errors into a single number – called a **loss**.

Try different ways:

- Total error:  $\sum(\hat{y}_i - y_i)$
- Total absolute error:  $\sum|\hat{y}_i - y_i|$
- Total squared error:  $\sum(\hat{y}_i - y_i)^2$

### Emerging Concept

The best-fitting line is the one that **minimizes total error** – depending on how we define it.

# Step 5: You Just Performed Optimization

What you just did:

- Defined a structure (a line:  $w x + b$ )
- Made predictions using parameters
- Computed how wrong the predictions were
- Adjusted the parameters to make the model better

## Conclusion

This process is called **optimization**. You used trial and error to minimize a loss – just like machine learning models do.

## Next Steps

In the next session, we'll introduce a formal algorithm: **Gradient Descent** – which automates what you just did.



# Summary: Hands-On Optimization Challenge

Step	Learning Outcome
Guess Parameters	Explore the effect of $w$ and $b$ on predictions
Compute Errors	Quantify prediction quality using error metrics
Improve by Adjustment	Learn through trial/error to minimize a loss
Define a Loss Function	Understand how total error gives a single measure to minimize