Optimization in Machine Learning

Ishaya, Jeremiah Ayock

May 25, 2025

Agenda

Why Optimization Matters in ML

Optimization is the backbone of ML

- Training models = solving optimization problems (minimizing loss functions).
- Examples:
  - **Linear regression:** Least squares minimization.
  - **Neural networks:** Non-convex loss landscapes (SGD, Adam).
- **Beyond training:** Hyperparameter tuning, architecture search, RL.
- **Trade-offs:** Accuracy vs. computational cost, generalization vs. overfitting.

Mathematically:

$$\min_{\theta} \mathcal{L}(\theta; \mathcal{D}) \quad \text{where } \theta = \text{parameters}, \mathcal{D} = \text{data}$$

First-Order Methods

- Gradient Descent (GD)
  - **Update Rule:**
    $$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\theta_t)$$
  - **Intuition:** *"Walking downhill"* on the loss surface.
  - **Limitations:** Sensitive to learning rate (η), local minima, and saddle points.
  - **Types:**
    - Batch Gradient Descent
    - Stochastic Gradient Descent (SGD)
    - Mini-Batch Gradient Descent

- Stochastic Gradient Descent (SGD) and Variants
  - **Mini-batch SGD:** Trade-off between noise and computational efficiency.
  - **Momentum:** Accelerates convergence by smoothing updates.

$$v_{t+1} = \gamma v_t + \eta \nabla \mathcal{L}(\theta_t)$$

- Improved Gradient Methods
  - **Momentum:** Smoother convergence
  - **Nesterov Momentum:** Looks ahead before stepping
  - **AdaGrad:** Per-parameter learning rates
  - **RMSprop:** Handles non-stationary loss
  - **Adam:** Momentum + adaptive learning

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla \mathcal{L}(\theta_t)$$

Adam Optimizer

- **Adam: Combines momentum + adaptive learning rates.**
  - Updates rules for m_t(momentum) and v_t(squared gradients).
  - **Advantages:** Robusst to ill-conditioned landscapes
  - **Caveats:** May not generalize as well as SGD in some cases.Read Wilson et al., 2017

Newton's Method and Approximations

- **Newton's Method :** Uses Hessian (H) for curvature-aware updates.

$$\theta_{t+1} = \theta_t - H^{-1}\nabla\mathcal{L}(\theta_t)$$

  - **Pros:** Faster convergence (quadratic rate).
  - **Cons:** Hessian (H) is expensive ($O(d^3)$ for d parameters).
- **Quasi-Newton Methods (BFGS, L-BFGS):**
  - Approximate Hessian with gradient differences.

Method	Cost per Iteration	Convergence Rate
GD	$O(d)$	Linear
Newton	$O(d^3)$	Quadratic
BFGS	$O(d^2)$	Superlinear

  - Popular in traditional ML (e.g., logistic regression).

Advanced Topics

- **Conjugate Gradient:** An iterative method for large linear systems.
- **Natural Gradient:** Uses Fisher information matrix for probabilistic models.
- **Recent Trends in Optimisation:**
  - **Shampoo:** Preconditioned SGD for deep learning shampoo
  - **K-FAC:** Kronecker-factored approximate curvature for neural nets.

Hyperparameter Optimisation (HPO) Methods

- **Grid or Random Search:** Simple but inefficient.
- **Bayesian Optimization (BO):**
  - Models loss surface as a Gaussian process.
  - Balances exploration-exploitation.
- **Gradient-Based HPO:**
  - Differentiable hyperparameters (e.g., meta-learning).
- **Multi-Fidelity Methods:** Successive Halving, BOHB (Combines BO and Bandits).

Conclusion and Q&A

Summary

- Optimisation is central to ML (training, tuning, and beyond).
- Gradient-based methods dominate, but second-order methods offer efficiency.
- HPO is critical; Modern tools (BO, gradient-based) help.

Open Challenges:

- Non-convex optimisation guarantees.
- Scalable second-order methods for deep learning.
- AutoML and end-to-end optimization.

Q&A:

- **Which optimizer works best for transformers?**
- **When to prefer second-order methods over Adam?**

References

- Convex Optimization: Algorithms and Complexity by Sebastien Bubeck Sebastien Bubeck
- Boyd & Vandenberghe, *Convex Optimization*.
- Wilson et al., *The Marginal Value of Adaptive Gradient Methods in ML* (2017). https://arxiv.org/abs/1705.08292
- Shampoo: https://arxiv.org/abs/2002.09018