Lecture 4, 5, 6, 7: From Linear Perspectives and μP to the Muon Optimizer

Jincheng Ou

Fall 25

Contents

1 The Linear Perspective and Lazy Training

1.1 Taylor Expansion of Neural Networks

We begin by treating a neural network $f(x, \theta)$ not as a black box, but as a function of its parameters $\theta \in \mathbb{R}^m$ for a fixed input x. We can analyze the training dynamics by linearizing the network around the current parameters θ_0 using a first-order Taylor expansion:

$$f(x, \theta_0 + \Delta\theta) \approx f(x, \theta_0) + \langle \nabla_\theta f(x, \theta_0), \Delta\theta \rangle + O(||\Delta\theta||^2) \tag{1}$$

Here, $\nabla_\theta f(x, \theta)$ acts as the "feature vector" of the model.

Definition 1.1: Lazy Training Regime

The **Lazy Training** assumption posits that during training, the weights θ move very little from their initialization θ_0 (i.e., $||\Delta\theta||$ is small). Consequently, the model behaves essentially as a linear model over fixed random features defined by $\nabla_\theta f(x, \theta_0)$.

This creates a fundamental tension in optimization:

- **Desire for Speed:** We want large updates ($\Delta\theta$) to minimize loss quickly.
- **Validity of Approximation:** We need small updates to ensure the Taylor approximation holds (trust region).

2 The General Optimizer Recipe: Constrained Optimization

To resolve the tension above, we formalize optimization as minimizing the linearized loss subject to a geometry-aware constraint on the step size.

2.1 The Objective Function

We aim to find the update step $\Delta\theta$ that minimizes the change in loss $\mathcal{L}$:

$$\Delta\theta^* = \underset{\Delta\theta}{\operatorname{argmin}} \langle \nabla_\theta \mathcal{L}(\theta), \Delta\theta \rangle \quad \text{s.t.} \quad ||\Delta\theta||_P \leq \eta \tag{2}$$

where $|| \cdot ||_P$ is a chosen norm defining the geometry of the trust region, and η is the step size.

2.2 Derivation 1: The Infinity Norm (ℓ_∞) → SignSGD

Let the trust region be defined by the ℓ_∞ norm: $||\Delta\theta||_\infty \leq \eta$. This implies $\max_j |\Delta\theta_j| \leq \eta$. The problem decouples into coordinate-wise optimization:

$$\min_{\Delta\theta} \sum_j \nabla_\theta \mathcal{L}[j] \cdot \Delta\theta[j] \quad \text{s.t.} \quad -\eta \leq \Delta\theta[j] \leq \eta \quad \forall j \tag{3}$$

Derivation 2.1: Optimal Solution for ℓ_∞

Since the objective is linear, the minimum occurs at the boundaries.

- If $\nabla\mathcal{L}[j] > 0$, we need $\Delta\theta[j]$ to be negative to minimize the product. We choose the largest magnitude allowed: $-\eta$.
- If $\nabla\mathcal{L}[j] < 0$, we choose $+\eta$.

Thus:

$$\Delta\theta[j] = -\eta \cdot \text{sign}(\nabla\mathcal{L}[j]) \tag{4}$$

This recovers SignSGD, which is the geometric foundation of Adam. Adam essentially performs SignSGD with momentum and de-biasing.

2.3 Derivation 2: The Euclidean Norm (ℓ_2) $\to$ Standard GD

Let the constraint be $||\Delta\theta||_2 \leq \eta$. We minimize $\langle g, \Delta\theta \rangle$. By the Cauchy-Schwarz inequality:

$$|\langle g, \Delta\theta \rangle| \leq ||g||_2 ||\Delta\theta||_2$$

The inner product is minimal (most negative) when $\Delta\theta$ is exactly **anti-aligned** with the gradient g:

$$\Delta\theta = -\eta \frac{g}{||g||_2} \tag{5}$$

This corresponds to normalized Gradient Descent.

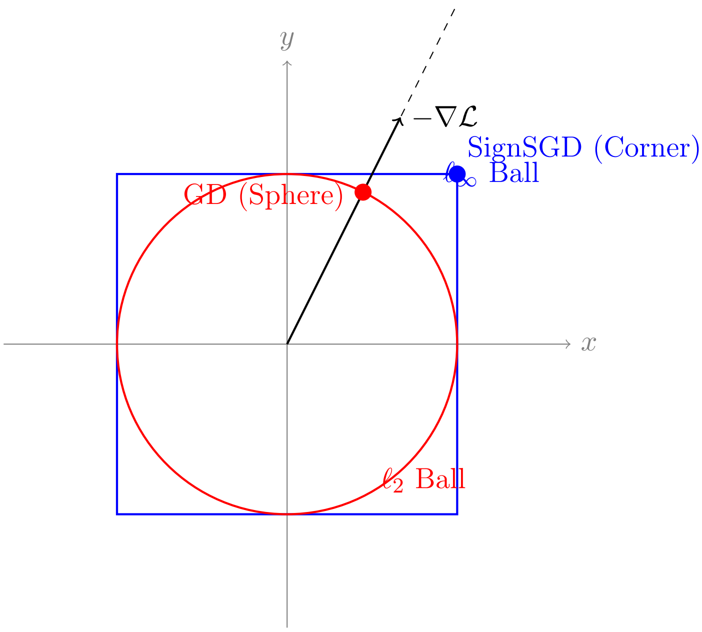

Figure 1: Geometric interpretation of Optimizers. SignSGD moves to the corners of the hypercube (using component-wise max steps), while GD moves to the boundary of the hypersphere.

3 Initialization and Data Standardization

Effective optimization requires that the inputs to non-linearities fall within their "active" regions.

3.1 The "ReLU Corner" Problem

Consider a ReLU unit $y = \max(0, wx + b)$. The "corner" or "elbow" occurs at $x = -b/w$.

- If input data is unstandardized (e.g., range $[100, 200]$) and initialization is standard normal $\mathcal{N}(0, 1)$, the probability of the "elbow" falling within the data range is nearly zero ($\approx 0.1\%$) .
- **Consequence:** The neuron behaves purely linearly or outputs zero for all data. Feature learning (which relies on non-linearity) fails.

This necessitates **data standardization** (mean 0, variance 1).

3.2 Xavier (Glorot) Initialization

We aim to preserve the variance of activations across layers to prevent vanishing/exploding gradients. Consider a linear layer $h_l = Wh_{l-1}$ where $W \in \mathbb{R}^{d_{out} \times d_{in}}$. Assuming inputs are independent with 0 mean:

$$\begin{aligned}
\text{Var}(h_l) &= \text{Var}\left(\sum_{j=1}^{d_{in}} W_{ij} h_{l-1,j}\right) && (6)\\
&= \sum_{j=1}^{d_{in}} \text{Var}(W_{ij} h_{l-1,j}) \quad \text{(Independence)} && (7)\\
&= \sum_{j=1}^{d_{in}} \mathbb{E}[W_{ij}^2]\mathbb{E}[h_{l-1,j}^2] \quad \text{(Zero Mean)} && (8)\\
&= d_{in} \cdot \text{Var}(W) \cdot \text{Var}(h_{l-1}) && (9)
\end{aligned}$$

To preserve variance ($\text{Var}(h_l) = \text{Var}(h_{l-1})$), we require:

$$d_{in} \cdot \text{Var}(W) = 1 \implies \text{Var}(W) = \frac{1}{d_{in}} \tag{10}$$

This derivation motivates the **RMS Norm** used in later optimizers.

4 Matrix Optimization: Shampoo and Semi-Orthogonality

Neural networks operate on matrices, not flat vectors. Optimizing in the matrix space yields better convergence properties.

4.1 The Matrix Trace Optimization

We reformulate the optimizer recipe for a weight matrix W. Objective: Maximize the correlation with the gradient $G = \nabla_W \mathcal{L}$ subject to a Spectral Norm constraint.

$$\max_{\Delta W} \text{Trace}(G^\top(-\Delta W)) \quad \text{s.t.} \quad ||\Delta W||_{\text{spectral}} \leq \eta \tag{11}$$

Derivation 4.1: Deriving the Shampoo Update

Let the Singular Value Decomposition (SVD) of the gradient be $G = U\Sigma V^\top$. Substitute into the trace:

$$\begin{aligned}
\text{Trace}(G^\top(-\Delta W)) &= \text{Trace}((U\Sigma V^\top)^\top(-\Delta W))\\
&= \text{Trace}(V\Sigma U^\top(-\Delta W))\\
&= \text{Trace}(\Sigma U^\top(-\Delta W)V) \quad \text{(Cyclic property)}
\end{aligned}$$

Let $Z = U^\top(-\Delta W)V$. The problem becomes maximizing $\text{Trace}(\Sigma Z)$ subject to $||Z||_2 \leq \eta$ (since unitary matrices U, V don't change spectral norm).

$$\text{Trace}(\Sigma Z) = \sum_i \sigma_i Z_{ii}$$

Since $\sigma_i \geq 0$, we maximize this by making Z diagonal with entries equal to the max allowed value η.

$$Z = \eta I \implies U^\top(-\Delta W)V = \eta I \implies \Delta W = -\eta U V^\top$$

Shampoo Implementation: Computing SVD is expensive (Shampoo approximates this using 4th roots of

$$W_{t+1} = W_t - \eta L_t^{-1/4} G_t R_t^{-1/4}$$

where $L_t \approx GG^\top$ and $R_t \approx G^\top G$.

5 Maximal Update Parameterization (μP)

Standard optimizers (like Adam) often require re-tuning the Learning Rate (LR) when the model width changes. μP provides a scaling law to transfer hyperparameters from small to large models.

5.1 The RMS-RMS Norm

Motivated by Xavier initialization, we define a matrix norm that is invariant to matrix .

Definition 5.1: Induced RMS-RMS Norm

$$||A||_{RMS \to RMS} = \max_{||x||_{RMS}=1} ||Ax||_{RMS}$$

where $||x||_{RMS} = \frac{1}{\sqrt{d_{in}}}||x||_2$.

Theorem 5.1: Relation to Spectral Norm

The RMS-RMS norm is a scaled version of the spectral norm:

$$||A||_{RMS \to RMS} = \sqrt{\frac{d_{in}}{d_{out}}}||A||_2 \tag{12}$$

5.2 Scaling Laws for Feature Learning

For a network to learn features effectively as width increases, two conditions must hold:

1. **Activations are O(1):** $||h_l||_{RMS} = \Theta(1)$.
2. **Updates are O(1):** $||\Delta h_l||_{RMS} = \Theta(1)$.

Using $\Delta h = \Delta W h$:

$$||\Delta h||_{RMS} \leq ||\Delta W||_{RMS \to RMS}||h||_{RMS}$$

Since $||h|| \approx 1$, we need $||\Delta W||_{RMS \to RMS} \approx 1$. Converting this back to the standard spectral norm updates used by optimizers:

$$||\Delta W||_2 \leq \eta \sqrt{\frac{d_{out}}{d_{in}}} \tag{13}$$

This implies the optimal learning rate should scale with $\sqrt{\frac{d_{out}}{d_{in}}}$ or inversely with fan-in d_{in} depending on the specific parameterization.

6 Muon: Momentum Orthogonalized by Newton-Schulz

Muon is an optimizer designed to combine the spectral efficiency of Shampoo with the low computational cost required for large-scale training (like LLMs).

6.1 Key Idea 1: RMS-RMS Optimization Geometry

Muon modifies the optimizer recipe to use the RMS-RMS norm constraint instead of the Spectral norm.

$$\text{argmin}\langle G, \Delta W\rangle \quad \text{s.t.} \quad ||\Delta W||_{RMS\to RMS} \le \eta$$

Using the scaling relation derived in Theorem 9.1, the update becomes:

$$\Delta W^* = -\eta\sqrt{\frac{d_{out}}{d_{in}}}UV^\top \tag{14}$$

This automatically injects the architecture-dependent scaling factors (μP) into the update.

6.2 Key Idea 2: Newton-Schulz Iteration

Instead of computing SVD ($UV^\top$) which is slow, Muon approximates the semi-orthogonal matrix $UV^\top$ using an iterative polynomial method.

Theorem 6.1: Newton-Schulz Convergence

The iteration $X_{k+1} = \frac{3}{2}X_k - \frac{1}{2}X_kX_k^\top X_k$ drives singular values towards 1.

Derivation 6.1: Polynomial SVD Interaction

Since odd polynomials commute with SVD, applying polynomial $P(X)$ to matrix $X = U\Sigma V^\top$ is equivalent to applying $P(\sigma)$ to the singular values [cite: 76-78]. Consider $P(x) = 1.5x - 0.5x^3$.

- If $\sigma = 1$: $P(1) = 1.5(1) - 0.5(1) = 1$ (Stable fixed point).
- If σ is small: $P(\sigma) \approx 1.5\sigma$ (Amplification).

Iterating this pushes singular values $\sigma \in (0, \sqrt{3})$ towards 1.

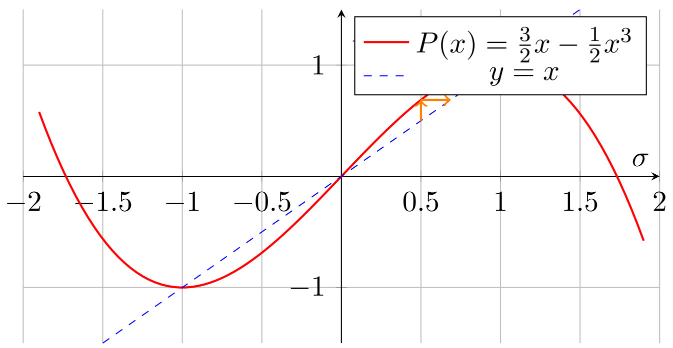

Figure 2: Newton-Schulz Polynomial. Values in $(0, \sqrt{3})$ converge to 1.

6.3 The Muon Algorithm

Muon replaces the adaptive arithmetic of Adam with Newton-Schulz orthogonalization:

Muon Update Step

1. **Momentum:** $B_t = \mu B_{t-1} + \nabla \mathcal{L}_t$
2. **Normalization:** To ensure convergence of Newton-Schulz (singular values $< \sqrt{3}$), normalize by Frobenius norm:
$$\hat{B}_t = \frac{B_t}{\max(1, ||B_t||_F / \sqrt{d_{in} d_{out}})}$$
3. **Newton-Schulz (approx 5 iterations):**
$$X_0 = \hat{B}_t, \quad X_{k+1} = \frac{3}{2} X_k - \frac{1}{2} X_k X_k^\top X_k$$
4. **Parameter Update:**
$$W_{t+1} = W_t - \eta \cdot \sqrt{\frac{\max(1, d_{out})}{\max(1, d_{in})}} \cdot X_{\text{final}}$$

7 Summary: The Evolution of Optimizers

1. **SGD:** Linear ℓ_2 constraint $\to$ Scale by gradient magnitude.
2. **Adam/SignSGD:** Linear ℓ_∞ constraint $\to$ Scale by coordinate-wise magnitude (sign).
3. **Shampoo:** Matrix Spectral constraint $\to$ Scale by singular values (Preconditioners).
4. **Muon:** Matrix RMS-RMS constraint + Newton-Schulz $\to$ Architecture-aware scaling + Efficient semi-orthogonalization.