Streaming Power Iteration for Muon: 4. Principles

Su Jianlin

2026-04-13

After the three articles "Streaming Power Iteration for Muon: 1. First Glance", "Streaming Power Iteration for Muon: 2. Acceleration", and "Streaming Power Iteration for Muon: 3. Refinement", readers should already have a good grasp of the idea, implementation, and acceleration details of Streaming Power Iteration. Overall, this can be regarded as a highly competitive method for implementing Muon, and thanks to its direct approximation of SVD computation, it also offers better scalability.

Due to space limitations, the mathematical principles behind the relevant operations were described relatively briefly at that time. Therefore, in this article, we will supplement some mathematical derivations regarding power iteration and QR decomposition, aiming to establish a more complete theoretical picture. However, these derivations still emphasize intuition over rigor, primarily to help clarify thinking (including the author's), and we ask professional readers to be indulgent.

Coaxial Equivalence

Before beginning the derivation, we need to introduce the concept of "coaxial equivalence". For matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times m}$, if there exists a signature matrix $\boldsymbol{S}$ such that $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{S}$, then $\boldsymbol{A}$ and $\boldsymbol{B}$ are said to be "coaxially equivalent", and each is called a "coaxial matrix" of the other. Here, a "signature matrix" refers to a diagonal matrix whose diagonal entries are ± 1, i.e., $\text{diag}(\pm 1, \pm 1, \cdots, \pm 1)$.

It should be noted that the term "coaxial" is coined by the author to describe this equivalence relation, because from a coordinate system perspective, matrices $\boldsymbol{A}, \boldsymbol{B}$ satisfying this condition actually describe the same coordinate system, differing only in the positive direction choices of certain axes. Some literature calls them "sign equivalent", but the author finds "coaxial" more intuitive.

The reason for introducing the concept of coaxial equivalence is that many matrix decompositions are unique only up to coaxial equivalence (thus the author is puzzled why such a commonly used equivalence relation lacks a standard name). For example, in QR decomposition, suppose matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}(n \geq m)$ is full-rank, and $\boldsymbol{Q}_1\boldsymbol{R}_1$ and $\boldsymbol{Q}_2\boldsymbol{R}_2$ are two QR decompositions of $\boldsymbol{A}$; then $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ are coaxially equivalent, and so are $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$.

This uniqueness is not hard to understand, because in the Gram-Schmidt orthogonalization process, we can arbitrarily reverse the direction of the orthogonalized vectors without changing orthogonality. Standard textbooks often ensure uniqueness of QR decomposition by requiring the diagonal entries of $\boldsymbol{R}$ to be positive, which is another valid approach but can sometimes bring unnecessary complications. Additionally, the uniqueness of SVD also holds under coaxial equivalence.

Power Iteration

In "Streaming Power Iteration for Muon: 1. First Glance", we introduced power iteration by sequentially solving for eigenvectors, then directly switched to the parallel version, asserting that it has the same convergence behavior. Here, we start directly from the parallel version and prove its convergence.

Let matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}(n \geq m)$ be full-rank (rank m), and consider the power iteration

$$\boldsymbol{V}_t = \mathrm{QR}(\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{V}_{t-1}), \qquad \boldsymbol{V}_0 = \boldsymbol{I} \tag{1}$$

Let the QR decomposition of $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{V}_{t-1}$ be $\boldsymbol{Q}_t\boldsymbol{R}_t$, so that $\mathrm{QR}(\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{V}_{t-1}) = \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{V}_{t-1}\boldsymbol{R}_t^{-1}$. Iterating step by step gives

$$\boldsymbol{V}_t = (\boldsymbol{A}^\top \boldsymbol{A})^t \boldsymbol{R}_1^{-1}\boldsymbol{R}_2^{-1}\cdots\boldsymbol{R}_t^{-1} \tag{2}$$

Next, let the SVD of $\boldsymbol{A}$ be $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, so that $\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top$. Substituting into the above yields

$$\boldsymbol{V}_t = \boldsymbol{V}\boldsymbol{\Sigma}^{2t}\boldsymbol{V}^\top \boldsymbol{R}_1^{-1}\boldsymbol{R}_2^{-1}\cdots\boldsymbol{R}_t^{-1} = \boldsymbol{V}\underbrace{(\boldsymbol{\Sigma}^{2t}\boldsymbol{V}^\top\boldsymbol{\Sigma}^{-2t})}_{\boldsymbol{B}}\underbrace{(\boldsymbol{\Sigma}^{2t}\boldsymbol{R}_1^{-1}\boldsymbol{R}_2^{-1}\cdots\boldsymbol{R}_t^{-1})}_{\boldsymbol{C}} \tag{3}$$

The result is divided into three parts: $\boldsymbol{V}, \boldsymbol{B}, \boldsymbol{C}$. We aim to prove that $\boldsymbol{V}_t \to \boldsymbol{V}$ in the coaxial sense. It is known that both $\boldsymbol{V}_t$ and $\boldsymbol{V}$ are orthogonal matrices. By the closure property of upper triangular matrices, $\boldsymbol{C}$ is upper triangular. If we can show that $\boldsymbol{B}$ tends to an upper triangular matrix, then by the uniqueness of QR decomposition (up to coaxial equivalence), we obtain $\boldsymbol{V}_t \to \boldsymbol{V}$. For $\boldsymbol{B} = \boldsymbol{\Sigma}^{2t}\boldsymbol{V}^\top\boldsymbol{\Sigma}^{-2t}$, writing in components gives

$$B_{i,j} = V_{j,i}(\sigma_i/\sigma_j)^{2t} \tag{4}$$

Assuming $\sigma_1 > \sigma_2 > \cdots > \sigma_m$, then for $i > j$, $(\sigma_i/\sigma_j)^{2t} \to 0$, meaning the sub-diagonal part of $\boldsymbol{B}$ tends to zero; in other words, $\boldsymbol{B}$ tends to an upper triangular matrix, satisfying the condition. This also implies that the convergence rate of power iteration depends on the ratio of adjacent singular values: the larger σ_i/σ_{i+1}, the faster the convergence. The iteration fails if there are equal singular values, but in practice, we can assume that the probability of two singular values being exactly equal is zero, thus avoiding this problem.

Essential Considerations

Let us carefully examine the entire proof process and understand what it essentially achieves.

First, the part $\boldsymbol{C} = \boldsymbol{\Sigma}^{2t}\boldsymbol{R}_1^{-1}\boldsymbol{R}_2^{-1}\cdots\boldsymbol{R}_t^{-1}$ appears complex, but its only role is to be "an upper triangular matrix"that is, its precise form is irrelevant as long as it remains upper triangular. The core part driving the result is $\boldsymbol{B} = \boldsymbol{\Sigma}^{2t}\boldsymbol{V}^\top\boldsymbol{\Sigma}^{-2t}$, which tends to an upper triangular matrix, allowing the leading $\boldsymbol{V}$ to be extracted via QR decomposition.

The step $(\boldsymbol{A}^\top \boldsymbol{A})^t$ is exactly what achieves this effect! In theory, $\boldsymbol{V}_t = \mathrm{QR}((\boldsymbol{A}^\top \boldsymbol{A})^t)$, meaning the repeated QR operations beforehand are theoretically redundantthe orthogonalization could be done just once at the end. Of course, this equivalence is only of theoretical value; directly computing $(\boldsymbol{A}^\top \boldsymbol{A})^t$ would cause numerical overflow or underflow, making results unusable. Thus, performing QR periodicallyinstead of only once at the endessentially maintains numerical stability.

We can also understand the multiple QR operations from the perspective of "preconditioning". A preconditioner refers to pre-processing steps on inputs that theoretically do not change the output but are usually beneficial for numerical computation. For QR, right-multiplying any full-rank upper

triangular matrix $\boldsymbol{R}$ does not alter the result, i.e., $\text{QR}(\boldsymbol{A}) = \text{QR}(\boldsymbol{AR})$. Therefore, any such $\boldsymbol{R}$ can be called a preconditioner for QR.

We know that QR itself can be written as right-multiplication by an upper triangular matrix. Thus, QR is a preconditioner for itself. Hence, we can insert QR operations into $(\boldsymbol{A}^\top\boldsymbol{A})^t$, transforming it into

$$(\boldsymbol{A}^\top\boldsymbol{A})^t \qquad \to \qquad \boldsymbol{A}^\top\boldsymbol{A}\,\text{QR}(\boldsymbol{A}^\top\boldsymbol{A}\,\text{QR}(\boldsymbol{A}^\top\boldsymbol{A}\cdots\text{QR}(\boldsymbol{A}^\top\boldsymbol{A}))) \tag{5}$$

Performing QR at both ends does not change the final result. However, since each QR produces an orthogonal matrix, right-multiplying by an orthogonal matrix carries almost no risk of numerical explosion, making QR a good self-preconditioner.

Related Variants

From the preconditioning viewpoint, we can understand many variants of power iteration. For example, the ColNorm version used at the beginning of the first article "Streaming Power Iteration for Muon: 1. First Glance":

$$\boldsymbol{V}_t = \text{QR}(\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{V}_{t-1}) \qquad \to \qquad \boldsymbol{V}_t = \text{QR}(\boldsymbol{A}^\top\,\text{ColNorm}(\boldsymbol{A}\boldsymbol{V}_{t-1})) \tag{6}$$

The two $\boldsymbol{V}_t$ are equal because column normalization ColNorm of a matrix is equivalent to right-multiplication by a diagonal matrix:

$$\text{ColNorm}([\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_m]) = [\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_m]\begin{bmatrix}\|\boldsymbol{a}_1\|^{-1} & 0 & \cdots & 0 \\ 0 & \|\boldsymbol{a}_2\|^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \|\boldsymbol{a}_m\|^{-1}\end{bmatrix} \tag{7}$$

Diagonal matrices are a special case of upper triangular matrices, so ColNorm is also a preconditioner for QR and does not change the result. For the same reason, we can shift ColNorm one step outward, which also preserves the QR result:

$$\boldsymbol{V}_t = \text{QR}(\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{V}_{t-1}) \qquad \to \qquad \boldsymbol{V}_t = \text{QR}(\text{ColNorm}(\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{V}_{t-1})) \tag{8}$$

Additionally, the protagonists of "Streaming Power Iteration for Muon: 2. Acceleration" and "Streaming Power Iteration for Muon: 3. Refinement" are the double QR power iteration:

$$\boldsymbol{V}_t = \text{QR}(\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{V}_{t-1}) \qquad \to \qquad \boldsymbol{V}_t = \text{QR}(\boldsymbol{A}^\top\,\text{QR}(\boldsymbol{A}\boldsymbol{V}_{t-1})) \tag{9}$$

From the discussions in those sections, it is easy to show that adding an extra QR to $\boldsymbol{A}\boldsymbol{V}_{t-1}$ does not theoretically change the result for $\boldsymbol{V}_t$.

Finite Precision Errors

We have repeatedly used phrases like "in theory", because many results rely on infinite-precision, exact QR decompositions. In practice, QR is computed with finite precision, and for efficiency we often use the less accurate "Shifted Cholesky QR (SCQR)", so analyzing the accumulation of errors becomes necessary.

By now, readers should be familiar with SCQR from the previous articles. It decomposes the QR of matrix $\boldsymbol{A}$ into two steps: 1) perform Cholesky decomposition on $\boldsymbol{A}^\top\boldsymbol{A} + \lambda\boldsymbol{I}$ to obtain $\boldsymbol{R}$,

2) compute the orthogonal matrix $\boldsymbol{Q} = \boldsymbol{A}\boldsymbol{R}^{-1}$. When $\lambda = 0$, SCQR is equivalent to exact QR. However, $\lambda = 0$ almost always fails in practice due to ill-conditioning. Setting an appropriate $\lambda > 0$ introduces error.

Fortunately, this error does not accumulate! Both $\boldsymbol{A}^\top\boldsymbol{A}$ and $\boldsymbol{A}^\top\boldsymbol{A} + \lambda\boldsymbol{I}$ yield upper triangular matrices upon Cholesky decomposition. Then $\boldsymbol{A}$ is right-multiplied by the inverse of such a triangular matrix. As we know, right-multiplication by an upper triangular matrix does not alter the QR result. Thus, ignoring errors from matrix multiplication, we can consider SCQR to be "lossless" with respect to QR.

In other words, as long as an exact QR is used in the final step, no matter how many inaccurate SCQR steps are performed earlier, the final result remains exact. Conversely, the total error is equivalent to the error of a single SCQR step (the last one), and does not accumulate. This principle underlies also the "SCQR2" acceleration technique mentioned in "Streaming Power Iteration for Muon: 2. Acceleration".

Conversely, if a transformation cannot be written as "right-multiplication by an upper triangular matrix", it will lossily destroy information of the original matrix. Over long iterations, such errors accumulate until the process fails completely. The simplification scheme proposed by @Ji_Ha_Kim in the previous article "Streaming Power Iteration for Muon: 3. Refinement" is a classic counterexample.

Cholesky Decomposition

Finally, let us briefly review "Cholesky decomposition". Its goal is to factor a given symmetric positive definite matrix $\boldsymbol{B}$ into $\boldsymbol{L}\boldsymbol{L}^\top$, where $\boldsymbol{L}$ is a lower triangular matrix. Denoting $\boldsymbol{R} = \boldsymbol{L}^\top$, we obtain the upper triangular form $\boldsymbol{R}^\top\boldsymbol{R}$.

Cholesky decomposition is fast because it admits a direct recursive analytic solution. Specifically, equating the entries from

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,m} \end{bmatrix} = \begin{bmatrix} l_{1,1} & & & \\ l_{2,1} & l_{2,2} & & \\ \vdots & \vdots & \ddots & \\ l_{m,1} & l_{m,2} & \cdots & l_{m,m} \end{bmatrix} \begin{bmatrix} l_{1,1} & l_{2,1} & \cdots & l_{m,1} \\ & l_{2,2} & \cdots & l_{m,2} \\ & & \ddots & \vdots \\ & & & l_{m,m} \end{bmatrix} \tag{10}$$

and solving entry by entry gives the recursive formulas

$$l_{j,j} = \pm\sqrt{a_{j,j} - \sum_{k=1}^{j-1} l_{j,k}^2}, \qquad l_{i,j} = \frac{1}{l_{j,j}}\left(a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}l_{j,k}\right) \quad \text{for } i > j \tag{11}$$

Cholesky QR is a classic application of Cholesky decomposition. Like Gram-Schmidt orthogonalization, it allows us to skip the orthogonalization procedure and directly obtain the triangular matrix $\boldsymbol{R}$. However, both face the same numerical difficulties: Gram-Schmidt proceeds vector by vector, and fails if linearly dependent or zero vectors are encountered, which manifests in singular values as low effective rank or large condition number; Cholesky decomposition similarly tends to fail under such conditions.

Another classic application of Cholesky decomposition is inversion of symmetric positive definite matrices. For example, solving $\boldsymbol{B}\boldsymbol{X} = \boldsymbol{C}$, where $\boldsymbol{B}$ is symmetric positive definite, reduces to solving two triangular systems once we decompose $\boldsymbol{B}$ as $\boldsymbol{L}\boldsymbol{L}^\top$. This process is efficient at each step. In particular, the iterative method proposed by @Ji_Ha_Kim for computing msign is based on inverting matrices via this approach.

Summary

This article mainly supplements some mathematical derivations regarding streaming power iteration, including convergence of power iteration, QR preconditioning, and a brief introduction to Cholesky decomposition, aiming to help readers better understand this method from a theoretical standpoint.