روش‌های یادگیری تانسوری

مهدی مولوی،

۲۳ بهمن ۱۳۹۶

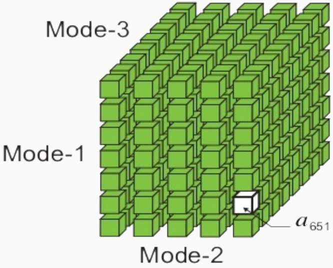
استاد راهنما: دکتر منصور رزقی

فهرست

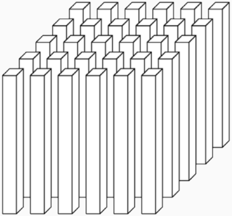
مقدمه

چند تعریف...

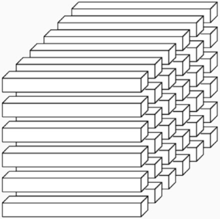
تانسور



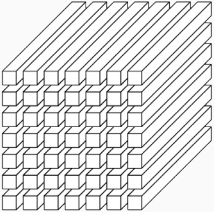
تانسور سه بعدی



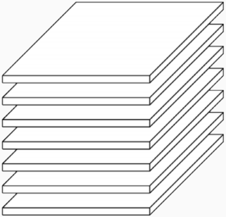
(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$



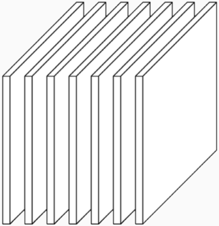
(b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$



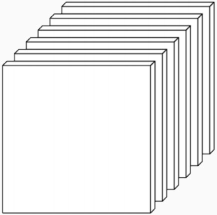
(c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$



(a) Horizontal slices: $\mathbf{X}_{i::}$



(b) Lateral slices: $\mathbf{X}_{:j:}$



(c) Frontal slices: $\mathbf{X}_{::k}$ (or $\mathbf{X}_k$)

ترانهاده ماتریس

ترانهاده تانسور ۳ـ بعدی $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ را با $\mathcal{A}^T$ نشان داده که یک تانسور با ابعاد $n_1 \times n_2 \times n_3$ است.

ترانهاده ماتریس

ترانهاده تانسور ۳ـ بعدی $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ را با $\mathcal{A}^T$ نشان داده که یک تانسور با ابعاد $n_1 \times n_2 \times n_3$ است.

تانسور همانی

یک تانسور سه بعدی صفر را که تنها در اسلایس اولی ماتریس همانی وجود دارد را تانسور همانی گوییم.

ترانهاده ماتریس

ترانهاده تانسور ۳ـ بعدی $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ را با $\mathcal{A}^T$ نشان داده که یک تانسور با ابعاد $n_1 \times n_2 \times n_3$ است.

تانسور همانی

یک تانسور سه بعدی صفر را که تنها در اسلایس اولی ماتریس همانی وجود دارد را تانسور همانی گوییم.

ضرب داخلی

ضرب داخلی دو تانسور هم‌اندازه $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ به‌صورت زیر تعریف می‌شود.

$$< \mathcal{X}, \mathcal{Y} >= \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} x_{i_1,i_2,\cdots,i_N} y_{i_1,i_2,\cdots,i_N}$$

انواع تجزیه

تجزیه CP

تجزیه CP یک تانسور، مجموع اجزای رتبه یک تانسور است. به عنوان مثال؛ تانسور سه بعدی $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ را می‌توانیم به صورت زیر تقریب بزنیم.

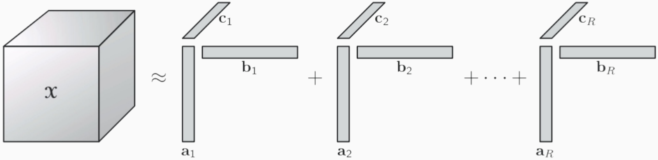
$$\mathcal{X} \approx \sum_{r=۱}^{R} a_r \circ b_r \circ c_r,$$

که در آن R یک عدد مثبت و $a_r \in \mathbb{R}^I$، $b_r \in \mathbb{R}^J$ و $c_r \in \mathbb{R}^K$ است.

تجزیه CP یک تانسور، مجموع اجزای رتبه یک تانسور است. به عنوان مثال؛ تانسور سه بعدی $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ را می‌توانیم به صورت زیر تقریب بزنیم.

$$\mathcal{X} \approx \sum_{r=1}^{R} a_r \circ b_r \circ c_r,$$

که در آن R یک عدد مثبت و $a_r \in \mathbb{R}^I$، $b_r \in \mathbb{R}^J$ و $c_r \in \mathbb{R}^K$ است.



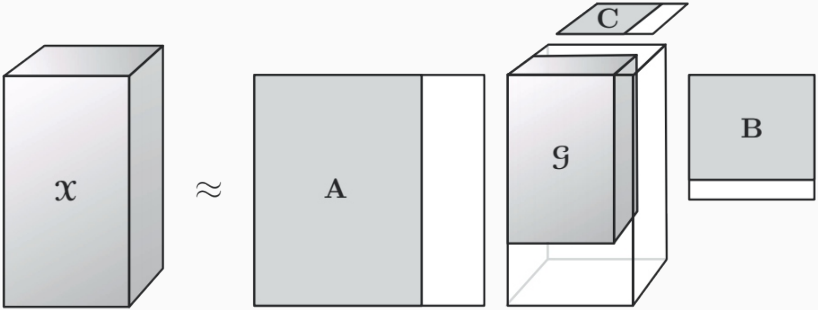
فرض کنید که هدف محاسبه بهترین تقریب برای تانسور سه بعدی $\mathcal{X}$ است.

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\| \quad \text{با} \quad \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r a_r \circ b_r \circ c_r = [\lambda; A, B, C].$$

که می‌توان با ثابت فرض کردن A و B، برای A حل کرد و همینطور به صورت تکراری برای ماتریس‌های دیگر.

HOSVD

HOSVD



تجزیه HOSVD

HOSVD

procedure HOSVD($\mathcal{X}, R_1, R_2, \ldots, R_N$)
  for $n = 1, \ldots, N$ **do**
    $\mathbf{A}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{X}_{(n)}$
  end for
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
  return $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

HOSVD

procedure HOSVD($\mathcal{X}, R_1, R_2, \ldots, R_N$)
    for $n = 1, \ldots, N$ **do**
        $\mathbf{A}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{X}_{(n)}$
    end for
    $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
    return $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

هدف محاسبه مسئله بهینه سازی زیر است:

$$\min_{\mathcal{G}, A^{(1)}, \cdots, A^{(N)}} \|\mathcal{X} - [\mathcal{G}; A^{(1)}, \cdots, A^{(N)}]\|$$

$$s.t \qquad \mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$$

که در آن $A^{(n)} \in \mathbb{R}^{I_n \times R_n}$ و متعامد است.

تعمیم یادگیری‌ها به ابعاد بالاتر

ماشین تانسور پشتیبان

تعمیم ماشین بردار پشتیبان

STM

ماشین بردار پشتیبان از بهینه‌سازی استفاده می‌کند و هدف آن پیدا کردن مرزهای جداسازی بهینه است. این مرزهای بهینه با کمینه کردن خطا میان تمام مرزهای ممکن تخمین زده شده توسط نمونه‌ها بدست می‌آید. تابع تصمیم ماشین بردار پشتیبان کلاسیک را می‌توان به صورت

$$y(\mathrm{a}) = sign(w^{\mathsf{T}} a + b)$$

نوشت. $\mathrm{a} \in \mathbb{R}^{L \times ۱}$ بردار بدون برچسب ورودی، $y \in \{-۱, +۱\}$ برچسب حدس زده شده و $w \in \mathbb{R}^{L \times ۱}$ ماتریس تصویر است.

STM

پارامترهای این مدل را می‌توان با حل کردن بهینه‌سازی زیر بدست آورد:

$$\min_{w,b,\xi} f(w,b,\xi) = \frac{۱}{۲}\|w\|^{۲} + c\sum_{i=۱}^{N}\xi_i$$

$$\text{s.t.} \quad y_i\left[w^{\top}\mathrm{a}_i + b\right] \geq ۱ - \xi_i, \quad \xi_i \geq ۰, \quad ۱ \leq i \leq N$$

هدف اصلی **ماشین تانسور پشتیبان** یافتن ابرصفحاتی است که داده‌های تانسوری را با توجّه به نمونه‌های آموزشی از هم جدا کند. تابع تصمیم‌گیری چندخطی برای ماشین تانسور پشتیبان، داده‌ی تانسوری بدون برچسب ورودی $\mathcal{A} \in \mathbb{R}^{I_۱ \times I_۲ \times \cdots I_N}$، به صورت

$$y(\mathcal{A}) = sgn\left(\mathcal{A}\prod_{k=۱}^{N} \times_k w_k + b\right)$$

تعریف می‌شود. که در آن $y \in \{+۱, -۱\}$، b است.

STM

وزن‌های w_k به صورت زیر محاسبه می‌شود:

$$\min_{w_k|_{k=1}^{N}} f\left(w_k|_{k=1}^{N}, b, \xi\right) = \frac{1}{2}\|w_1 \circ w_2 \circ \cdots \circ w_N\|^2 + c\sum_{i=1}^{N}\xi_i \qquad (1)$$

$$\text{s.t.} \quad y_i\left[\mathcal{A}_i \prod_{k=1}^{N} \times_k w_k + b\right] \geq 1 - \xi_i, 1 \leq i \leq M, \quad \xi_i \geq 0$$

STM

وزن‌های w_k به صورت زیر محاسبه می‌شود:

$$\min_{w_k|_{k=1}^{N}} f\left(w_k|_{k=1}^{N}, b, \xi\right) = \frac{1}{2}\|w_1 \circ w_2 \circ \cdots \circ w_N\|^2 + c\sum_{i=1}^{N}\xi_i \qquad (1)$$

$$\text{s.t.} \quad y_i\left[\mathcal{A}_i \prod_{k=1}^{N} \times_k w_k + b\right] \geq 1 - \xi_i, 1 \leq i \leq M, \quad \xi_i \geq 0$$

که در آن $w_k|_{k=1}^{N}$ بیانگر وزن در حالت kام است. در این معادله، ξ_i بردار متغیّر اسلک بوده و c ضریب منظّم‌ساز برای کنترل خطاهای طبقه‌بندی است. همچنین M تعداد کلّ نمونه‌ها و $\mathcal{A}_i$ نشان‌دهنده‌ی iامین تانسور داده‌های آموزشی است.

STM

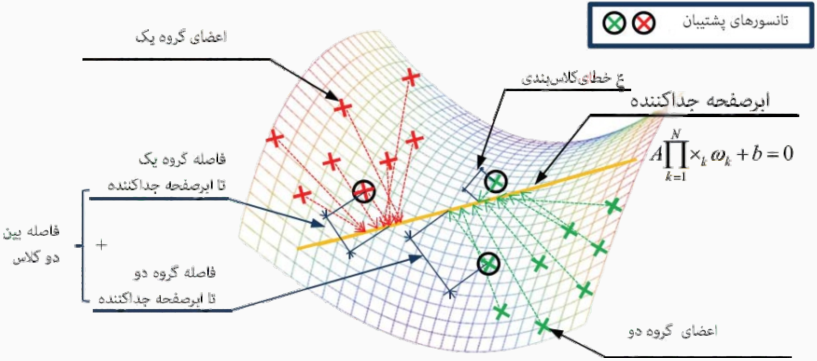
می‌توان از روش‌های *متناوب* برای پیدا کردن جوابی مناسب استفاده کرد. گرچه این روش‌ها تضمینی برای رسیدم به نقطه‌ی کمینه به ما نمی‌دهند اما ثابت شده که همواره به یک نقطه‌ی کمینه‌ی محلی خواهند رسید. پس از محاسبه‌ی w_j و b می‌توان ابرصفحه‌ی جداساز را با استفاده از رابطه‌ی زیر محاسبه کرد:

STM

می‌توان از روش‌های *متناوب* برای پیدا کردن جوابی مناسب استفاده کرد. گرچه این روش‌ها تضمینی برای رسیدم به نقطه‌ی کمینه به ما نمی‌دهند اما ثابت شده که همواره به یک نقطه‌ی کمینه‌ی محلی خواهند رسید. پس از محاسبه‌ی w_j و b می‌توان ابرصفحه‌ی جداساز را با استفاده از رابطه‌ی زیر محاسبه کرد:

$$\mathcal{A}\prod_{k=۱}^{N} \times_k w_k + b = ۰.$$

نمایی از جداسازی دودویی به وسیله‌ی ماشین تانسور پشتیبان در فضای با ابعاد بالا



Clustering Spectral Tensor

Regression Tensor Logistic

سرعت بخشی

HOSVD for Dimensionality Reduction

مقدمه

فرض کنید $M_1 \in \mathbb{R}^{m_1 \times n}$ و $M_2 \in \mathbb{R}^{m_2 \times n}$ که $m_1 < m_2$ است و $M_2 = [M_1, \ \ 0]$.

هرگاه تجزیه SVD آن‌ها به صورت زیر باشد:

$$
\begin{aligned}
M_1 &= U_1 \Sigma_1 V_1^T \\
M2 &= U_1 \Sigma_1 V_1^T
\end{aligned}
$$

در این صورت ماتریس متعامد U_1 معادل با U_2 است.

مقدمه

فرض کنید $M_۱ \in \mathbb{R}^{m_۱ \times n}$ و $M_۲ \in \mathbb{R}^{m_۲ \times n}$ که $m_۱ < m_۲$ است و $M_۲ = [M_۱, \ \ ۰]$.

هرگاه تجزیه SVD آن‌ها به صورت زیر باشد:

$$M_۱ = U_۱ \Sigma_۱ V_۱^T$$
$$M۲ = U_۱ \Sigma_۱ V_۱^T$$

در این صورت ماتریس متعامد $U_۱$ معادل با $U_۲$ است.

نتیجه: فرض کنید $M_۱ = [v_۱, \cdots, v_n]$ و $M_۲ = [v_۱, v_۲, \cdots, ۰, \cdots, ۰, v_n]$، در این صورت پایه‌هی معادل دارن.

معادل بودن تانسوری هسته در $HOSVD$

فرض کنید $T \in \mathbb{R}^{I_1 \times \cdots I_N}$ و $G \in \mathbb{R}^{I_1 \times \cdots \times (kI_n) \times \cdots I_N}$ باشد. با در نظر گرفتن ماتریس

$$M = \begin{pmatrix} I \\ 0 \end{pmatrix}, \qquad M \in \mathbb{R}^{I_n \times (kI_n)}$$

تانسور T و G در رابطه زیر صدق می‌کند:

$$T = G \times_n M = G \times_n \begin{pmatrix} I_n \\ 0_{kn} \end{pmatrix},$$

IHOSVD

این روش شامل سه الگوریتم است که در اولین الگوریتم، یک الگوریتم بازگشتی می‌باشد که تابع زیر را بدست می‌آورد:

$$f(M_n, C_n) = \begin{cases} svd(M_۱), & n = ۱ \\ mix(f(M_{n-۱}, C_{n-۱}), C_n), & n > ۱ \end{cases}$$

IHOSVD

Algorithm 1 Recursive matrix singular value decomposition, $(U, \Sigma, V) = R - MSvd(M_n, C_n)$.

Input:

Initial matrix M_n.

Incremental matrix C_n.

Output:

Decomposition results U, S, V of matrix $[M_n \ C_n]$.

if $(n == 1)$ **then**

  $[U, \Sigma, V] = svd(M_1)$.

else

  $[U_m, \Sigma_m, V_m] = R - MSvd(M_{n-1}, C_{n-1})$.

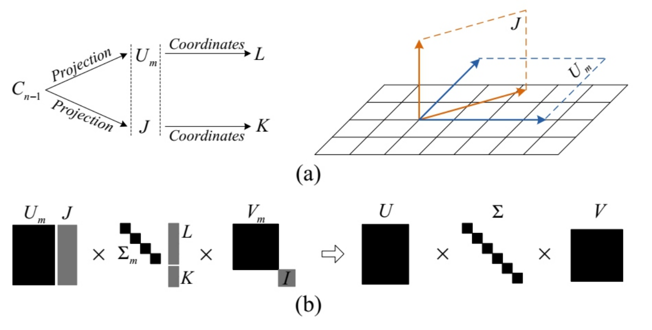
  $[U, \Sigma, V] = mix(M_{n-1}, C_{n-1}, U_m, \Sigma_m, V_m)$.

end if

return U, S, V.

IHOSVD

دومین الگوریتم، SVD را بروزرسانی می‌کند. از ورودی ماتریس $C_{n-۱}$ را خوانده و به فضای متعامد تولید شده از U_m تصویر می‌کند.



Algorithm 2 Merge incremental matrix with decomposition results, $(U, \Sigma, V) = mix(M_{n-1}, C_{n-1}, U_m, \Sigma_m, V_m)$.

Input:

Initial matrix M_{n-1} and incremental matrix C_{n-1}.

Decomposition results U_m, Σ_m, V_m of matrix M.

Output:

New decomposition results U, Σ, V.

1: Project C_{n-1} on the orthogonal space spanned by U_m, $L = U_m^{\mathrm{T}} \times C_{n-1}$.

2: Compute H which is orthogonal to U_m, $H = C_{n-1} - U_m \times L$.

3: Obtain the unitary orthogonal basis J from matrix H.

4: Compute the coordinates of matrix H, $K = J^{\mathrm{T}} \times H$.

5: Execute SVD on the new matrix $[U \ J]$, $[U', \Sigma', V'] = svd([U \ J])$.

6: Obtain new decomposition results, $([U \ \ J], \ \ U') \rightarrow U$, $\Sigma' \rightarrow \Sigma$, $V' \rightarrow V$.

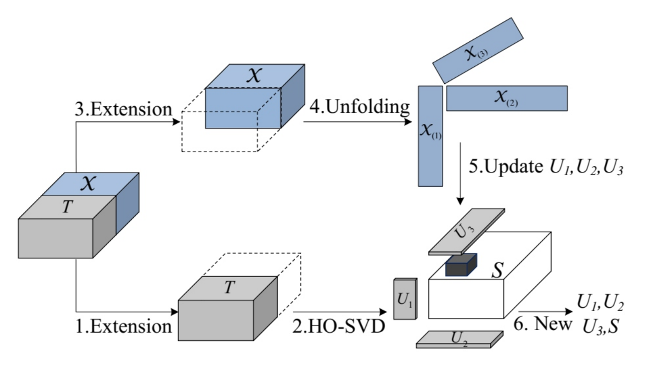
7: **return** U, S, V.

IHOSVD

با استفاده از سومین الگوریتم ما هسته تانسور را حساب خواهیم کرد.

IHOSVD

با استفاده از سومین الگوریتم ما هسته تانسور را حساب خواهیم کرد.



Algorithm 3 Incremental tensor singular value decomposition, $(S,\ [U,\ \Sigma,\ V]_{new}) = I - TSvd(\chi,\ T,\ [U,\ \Sigma,\ V]_{initial})$.

Input:

New tensor $\chi \in R^{I_1 \times I_2 \times \ldots \times I_N}$.

Previous tensor $T \in R^{I_1 \times I_2 \times \ldots \times I_N}$.

Previous unfolded matrices SVD results $[U,\ \Sigma,\ V]_{initial}$.

Output:

New truncated SVD results $[U,\ \Sigma,\ V]_{new}$.

New core tensor S.

1: Extend tensor χ and tensor T to identical dimensionality.
2: Unfold new tensor χ to matrices $\chi_{(1)}, \ldots, \chi_{(N)}$.
3: Call algorithm $R - MSvd$ to update above unfolded matrices.
4: Truncate the new orthogonal bases.
5: Combine new tensor χ with initial tensor T.
6: Obtain new core tensor S with n-mode product.
7: **return** S, and $[U,\ \Sigma,\ V]_{new}$.

زمان پیچیدگی

زمان محاسبه برای روش ارائه شده به صورت مجموع سه الگوریتم است که به صورت زیر محاسبه می‌شود:

$$Time = Time_{unf} + Time_{isvd} + Time_{prod}.$$

که عمله تبدیل unfolding یک عمل از مرتبه $\mathcal{O}(۱)$ است. زمان $Time_{isvd}$ مجموع زمان‌های مصرفی برای unfold کردن ماتریس $T_{(i)}$ است.

$$Time_N = \sum_{i=۱}^{N} Time_i$$

$$Time(n) = \begin{cases} C_۱, & n = ۱ \\ Time(n-۱) + C_۲, & n > ۱ \end{cases}$$

پس زمان کل برابر $\mathcal{O}(k^۲ n)$ است.

رویکرد دیگر برای تجزیه SVD برای ابعاد بالاتر

T_SVD

تعاریف مقدماتی

ماتریس پیچشی

تانسور سه بعدی $\mathcal{A} \in \mathbb{R}^{l\times m\times n}$ با $l \times m$ اسلایس پیشی مفروض است.

در این صورت

$$circ(\mathcal{A}) = \begin{pmatrix} A^{(1)} & A^{(n)} & A^{(n-1)} & \dots & A^{(2)} \\ A^{(2)} & A^{(1)} & A^{(n)} & A^{(n-1)} & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{(n)} & A^{(n-1)} & \dots & A^{(2)} & A^{(1)} \end{pmatrix}$$

که ماتریس پیچشی بلوکی از سایز $ln \times mn$ است.

T_Prudact

عملگرهای $fold$ و $MatVec$

عملگر $MatVec$ تانسور سه بعدی به ابعاد $l \times m \times n$ را به ماتریس بلوکی با ابعاد $ln \times m$ برمی‌گرداند و عملگر $fold$ معکوس این عملگر را انجام می‌دهد.

$$fold(MatVec(\mathcal{A})) = \mathcal{A}, \quad MatVec(\mathcal{A}) = \begin{pmatrix} A^{(۱)} \\ A^{(۲)} \\ \vdots \\ A^{(n)} \end{pmatrix}$$

tـضرب

فرض کنید $\mathcal{A} \in \mathbb{R}^{l\times p\times n}$ و $\mathcal{B} \in \mathbb{R}^{p\times m\times n}$. در این صورت tـضرب با ابعاد $l \times m \times n$ را به صورت $\mathcal{A} \star \mathcal{B}$ نشان داده و به صورت زیر تعریف می‌شود.

$$\mathcal{A} \star \mathcal{B} = fold\left(circ(\mathcal{A}) MatVec(\mathcal{B})\right).$$

tـضرب

فرض کنید $\mathcal{A} \in \mathbb{R}^{l \times p \times n}$ و $\mathcal{B} \in \mathbb{R}^{p \times m \times n}$. در این صورت tـضرب با ابعاد $l \times m \times n$ را به صورت $\mathcal{A} \star \mathcal{B}$ نشان داده و به صورت زیر تعریف می‌شود.

$$\mathcal{A} \star \mathcal{B} = fold\left(circ(\mathcal{A}) MatVec(\mathcal{B})\right).$$

این ضرب در حالت کلی خاصیت جابجایی ندارد.

tـضرب

فرض کنید $\mathcal{A} \in \mathbb{R}^{l\times p\times n}$ و $\mathcal{B} \in \mathbb{R}^{p\times m\times n}$. در این صورت tـضرب با ابعاد $l \times m \times n$ را به صورت $\mathcal{A} \star \mathcal{B}$ نشان داده و به صورت زیر تعریف می‌شود.

$$\mathcal{A} \star \mathcal{B} = fold\left(circ(\mathcal{A})MatVec(\mathcal{B})\right).$$

این ضرب در حالت کلی خاصیت جابجایی ندارد.

تبدیل فوریه گسسته (DFT)

ماتریس مربعی F با ابعاد N در N را به صورت $F = \left(\frac{w^{jk}}{\sqrt{N}}\right)_{j,k=0,\cdots,N-1}$ که در آن $\omega = e^{-2\pi i/N}$ است. به صورت ماتریس داریم:

$$\mathrm{F} = \frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix},$$

T_SVD

هرگاه F ماتریس نرمال شده تبدیل فوریه باشد، برای $\mathcal{A} \in \mathbb{R}^{l \times m \times n}$، n ماتریس با درآیه‌های مختلط $\hat{A}^{(i)}$ با ابعاد $l \times m$ وجود دارد به طوریکه:

$$(F \otimes I)circ(\mathcal{A})(F^* \otimes I) = \begin{pmatrix} \hat{A}^{(1)} & 0 & \cdots & 0 \\ 0 & \hat{A}^{(2)} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{A}^{(n)} \end{pmatrix} = \hat{\mathcal{A}},$$

اما نیاز نیست که به صورت صریح $\hat{\mathcal{A}}$ از $circ(\mathcal{A})$ تولید شود. با تبدیل فوریه از مد سوم تانسور $\mathcal{A}$ تانسور $\hat{\mathcal{A}}$، تولید می‌شود.

T_SVD

هرگاه F ماتریس نرمال شده تبدیل فوریه باشد، برای $\mathcal{A} \in \mathbb{R}^{l\times m\times n}$، n ماتریس با درآیه‌های مختلط $\hat{A}^{(i)}$ با ابعاد $l \times m$ وجود دارد به طوریکه:

$$(F \otimes I)circ(\mathcal{A})(F^* \otimes I) = \begin{pmatrix} \hat{A}^{(۱)} & ۰ & \cdots & ۰ \\ ۰ & \hat{A}^{(۲)} & \cdots & ۰ \\ \vdots & \ddots & \ddots & \vdots \\ ۰ & ۰ & \cdots & \hat{A}^{(n)} \end{pmatrix} = \hat{\mathcal{A}},$$

اما نیاز نیست که به صورت صریح $\hat{\mathcal{A}}$ از $circ(\mathcal{A})$ تولید شود. با تبدیل فوریه از مد سوم تانسور $\mathcal{A}$ تانسور $\hat{\mathcal{A}}$، تولید می‌شود.

توجه: برای تانسور $\mathcal{A}$ و $\mathcal{B}$ داریم:

$$\mathcal{A} \star \mathcal{B} = \mathcal{C} \Longleftrightarrow \hat{\mathcal{A}}\hat{\mathcal{B}} = \hat{\mathcal{C}}$$

T_SVD and T_QR

تجزیه $T-QR$

با استفاده از تعریف فوق می‌توان تانسور $\mathcal{A}$ را بصورت $\mathcal{A} = \mathcal{Q} \star \mathcal{R}$ نوشت که در آن $\mathcal{Q}$ معکوس تبدیل فوریه $\hat{\mathcal{Q}}$ مد ۳ است.

T_SVD and T_QR

تجزیه $T-QR$

با استفاده از تعریف فوق می‌توان تانسور $\mathcal{A}$ را بصورت $\mathcal{A} = \mathcal{Q} \star \mathcal{R}$ نوشت که در آن $\mathcal{Q}$ معکوس تبدیل فوریه $\hat{\mathcal{Q}}$ مد ۳ است.

تجزیه $T-SVD$

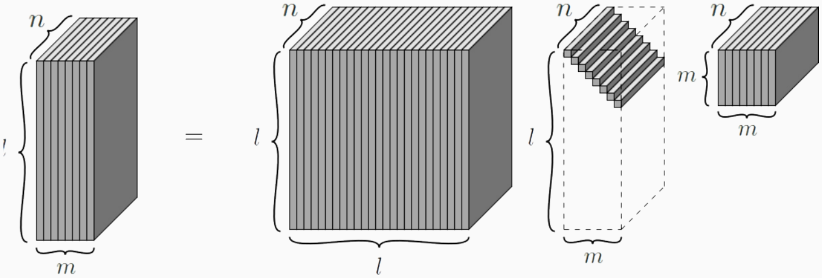
با استفاده از تعریف می‌توان تانسور $\mathcal{A}$ را بصورت $\mathcal{A} = \mathcal{U} \star \mathcal{S} \star \mathcal{V}^{\top}$ نوشت.

T-SVD

تجزیه $t-SVD$ که در آن S تانسور قطری است به فرم زیر نشان داده می‌شود.

T_SVD

تجزیه $t - SVD$ که در آن S تانسور قطری است به فرم زیر نشان داده می‌شود.



الگوریتم $t-SVD$ برای تانسور سه بعدی به صورت زیر است:

Algorithm 1 t-SVD for third order tensors

Input: $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.
Output: $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$.

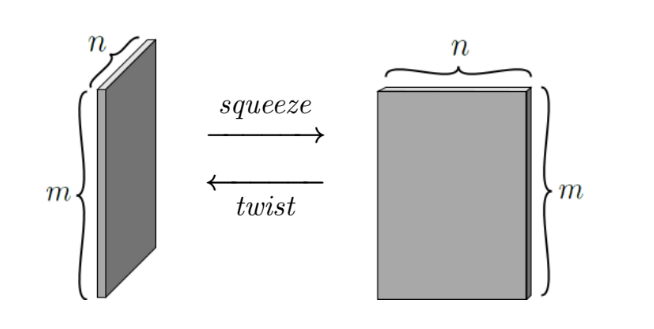
$\mathcal{D} \leftarrow \mathrm{fft}(\mathcal{M}, [\,], 3)$
for $i = 1$ to n_3 **do**
  $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathtt{svd}(\mathcal{D}^{(i)})$
  $\widehat{\mathcal{U}}^{(i)} = \mathbf{U}$; $\widehat{\mathcal{S}}^{(i)} = \mathbf{S}$; $\widehat{\mathcal{V}}^{(i)} = \mathbf{V}$;
end for
$\mathcal{U} \leftarrow \mathrm{ifft}(\widehat{\mathcal{U}}, [\,], 3)$; $\mathcal{S} \leftarrow \mathrm{ifft}(\widehat{\mathcal{S}}, [\,], 3)$; $\mathcal{V} \leftarrow \mathrm{ifft}(\widehat{\mathcal{V}}, [\,], 3)$

۱ تعریف معین مثبت.

۱ تعریف معین مثبت.
۲ تانسور متعامد

۱ تعریف معین مثبت.
۲ تانسور متعامد
۳ $squeenz$

۱ تعریف معین مثبت.
۲ تانسور متعامد
۳ $squeenz$



ضرب داخلی

۴ ضرب داخلی برای $y \star x^T =: < y, x >$

ضرب داخلی

۴ ضرب داخلی برای $< x, y >:= x^T \star y$

$$\begin{aligned}
&1.\ \langle \vec{\mathcal{X}}, \vec{\mathcal{Y}} + \vec{\mathcal{Z}} \rangle = \langle \vec{\mathcal{X}}, \vec{\mathcal{Y}} \rangle + \langle \vec{\mathcal{X}}, \vec{\mathcal{Z}} \rangle \\
&2.\ \langle \vec{\mathcal{X}}, \vec{\mathcal{Y}} * \mathbf{a} \rangle = (\vec{\mathcal{X}}^T * \vec{\mathcal{Y}}) * \mathbf{a} = \mathbf{a} * (\vec{\mathcal{X}}^T * \vec{\mathcal{Y}}) = \mathbf{a} * \langle \vec{\mathcal{X}}, \vec{\mathcal{Y}} \rangle \\
&3.\ \langle \vec{\mathcal{X}}, \vec{\mathcal{Y}} \rangle = \langle \vec{\mathcal{Y}}, \vec{\mathcal{X}} \rangle^T
\end{aligned}$$

کاربرد $TSVD$

$$\min_{\mathcal{X}} \ \|\mathcal{X}\|_{TNN}$$
$$\text{subject to } \mathcal{X}_{ijk} = \mathcal{M}_{ijk}, (i,j,k) \in \Omega$$

نتیجه گیری

...

سوال؟