第1回量子ソフトウェア産学協働ゼミ

テンソルネットワークを使ったパラメータ圧縮

理学系研究科　量子ソフトウェア寄付講座　**大久保　毅**

コンテンツ

- テンソルネットワークとデータ圧縮
  - 行列の特異値分解とテンソルネットワーク
  - テンソルの行列積分解
  - 画像圧縮への適用例
- 時系列解析への応用に向けて
  - 演算子の行列積分解：行列積演算子
  - 効率的なテンソルネットワーク分解のポイント

テンソルネットワークとデータ圧縮

データ圧縮の例：行列の特異値分解

特異値分解

$$A : M \times N$$

$$A_{ij} \in \mathbb{C}$$

$$A = U \Sigma V^{\dagger}$$

$U : M \times M$ **ユニタリ行列**

$V : N \times N$ **ユニタリ行列**

$$\Sigma = \begin{pmatrix} \Sigma_{r \times r} & 0_{r \times (N-r)} \\ 0_{(M-r) \times r} & 0_{(M-r) \times N-r} \end{pmatrix}$$

$$\Sigma_{r \times r} = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix}$$

非負の実数を要素にもつ

対角行列

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$

特異値

SVD表現でのデータの量

$A : M \times N$

$$A = U\Sigma V^\dagger = U \begin{pmatrix} \Sigma_{r\times r} & 0_{r\times(N-r)} \\ 0_{(M-r)\times r} & 0_{(M-r)\times N-r} \end{pmatrix} V^\dagger$$

特異値がゼロの部分を消去 → $= \bar{U}\Sigma_{r\times r}\bar{V}^\dagger$

$$\bar{U} : M \times r, \bar{V}^\dagger : r \times N$$

行列Aのランクが M や Nよりも
ずっと小さい場合： $r \ll M, N$
A を表現するためのデータ量を減らせる
（この段階では何も近似は無い）

$$U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)$$
$$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

↓

$$\bar{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r)$$
$$\bar{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r)$$

SVDによる低ランク近似

小さい特異値を無視して近似行列を作る

$$A = \bar{U}\Sigma_{r\times r}\bar{V}^{\dagger} \quad \Rightarrow \quad \tilde{A} = \tilde{U}\Sigma_{k\times k}\tilde{V}^{\dagger} \qquad (k < r)$$

$$\Sigma_{r\times r} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$$
$$\bar{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r)$$
$$\bar{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r)$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

$$\mathrm{rank}(A) = r$$

$$\Sigma_{k\times k} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$$
$$\tilde{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k)$$
$$\tilde{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k)$$

大きい k 個の特異値だけを残す

（対応する特異ベクトルも）

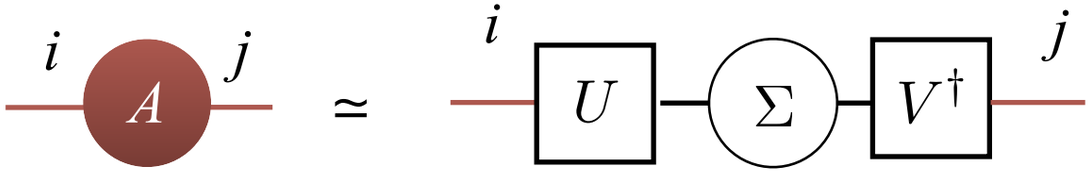
$$\mathrm{rank}(\tilde{A}) = k < r$$

特異値分解による低ランク近似は行列に対する最適な低ランク近似

Cf. Eckart - Young - Mirsky Theorem

テンソルの"低ランク"近似

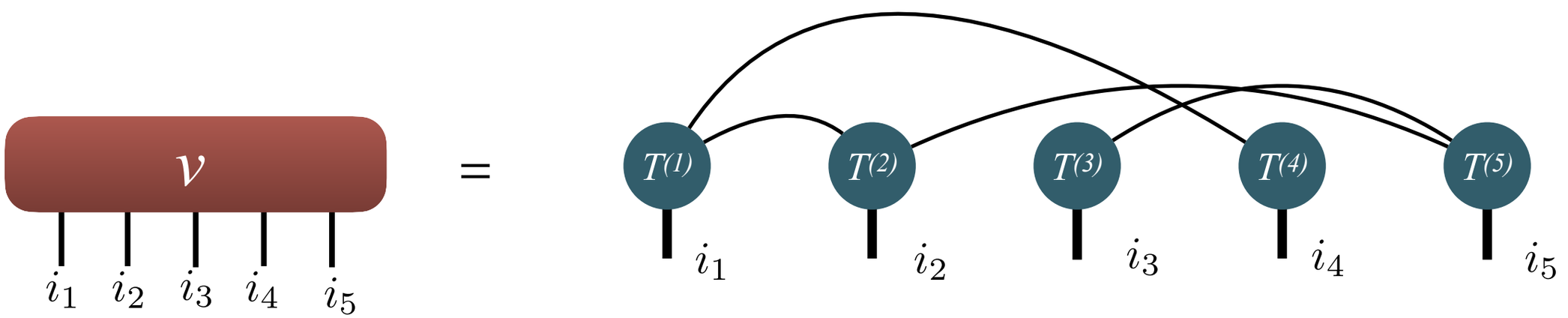
行列の特異値分解（低ランク近似）



行列を小さい行列の積に分解

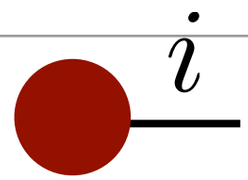
テンソルへの拡張？

テンソルネットワーク分解

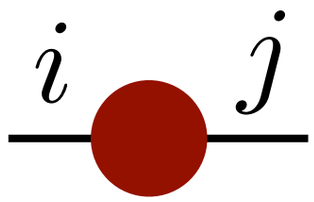


ダイアグラムを用いたテンソル表記

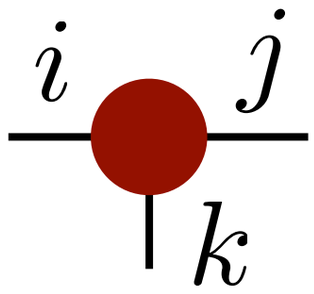
- ベクトル　　$\vec{v}$: v_i



- 行列　　M : $M_{i,j}$



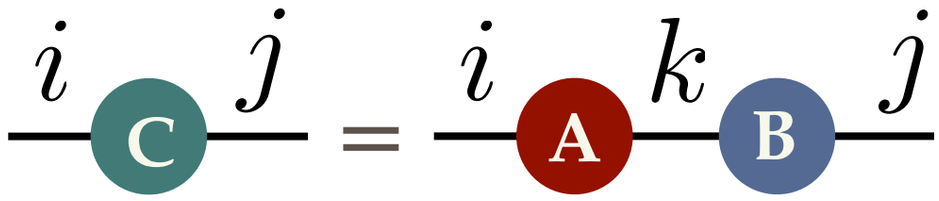
- テンソル　　T : $T_{i,j,k}$



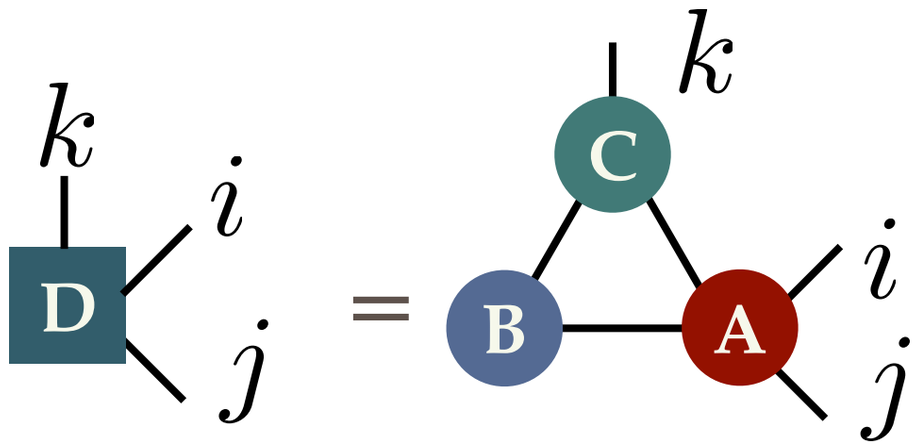
＊n階のテンソル＝n本の足

テンソルの積（縮約）の表現

$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$



$$D_{i,j,k} = \sum_{\alpha,\beta,\gamma} A_{i,j,\alpha,\beta} B_{\beta,\gamma} C_{\gamma,k,\alpha}$$



テンソルの行列積分解

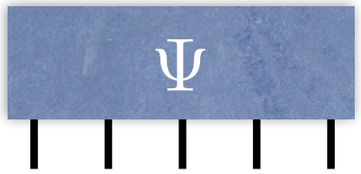
Good reviews: U. Schollwöck, Annals. of Physics **326**, 96 (2011).
R. Orús, Annals. of Physics **349**, 117 (2014).
J. C. Bridgeman and C. T. Chubb, J. Phys. A: Math. Theor. **50** 223001 (2017).

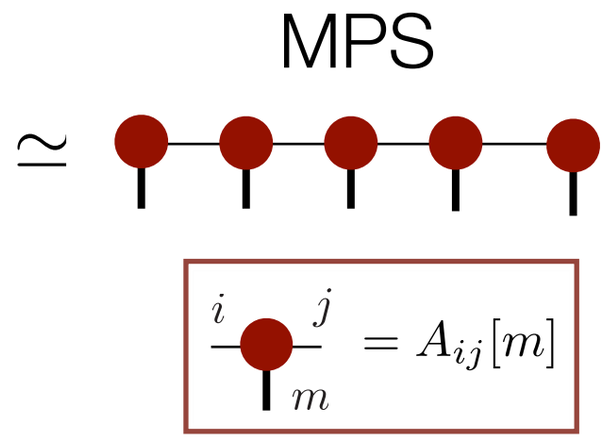
行列積状態：Matrix product state (MPS)

N本足のテンソルを"行列"の一次元的な繋がりで表現

$$\Psi_{i_1 i_2 \ldots i_N} \simeq A_1[i_1] A_2[i_2] \cdots A_N[i_N]$$

$A[i]$ ：index i に対応する行列





基本的なテンソルネットワーク分解の一つ

Note:

- MPSは量子多体状態の表現として物理学分野で発展してきた

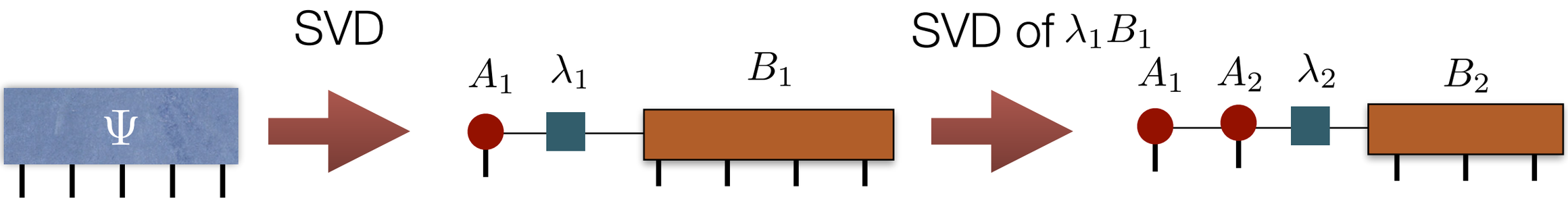
$$|\Psi\rangle = \sum_{\{i_1, i_2, \ldots i_N\}} \Psi_{i_1 i_2 \ldots i_N} |i_1 i_2 \ldots i_N\rangle$$

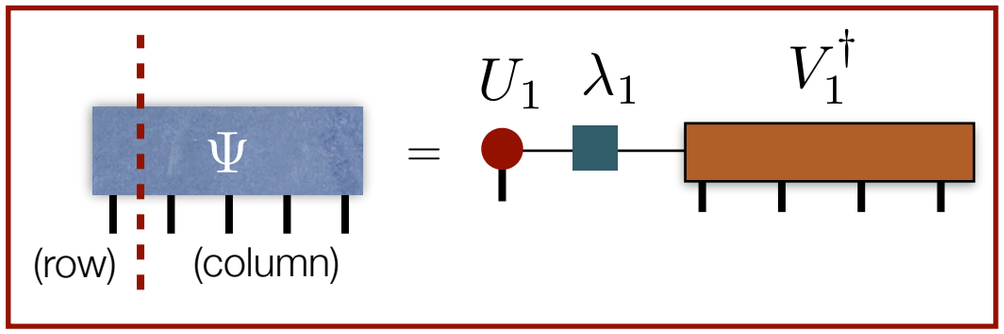
- MPS は "tensor train decomposition" とも呼ばれている

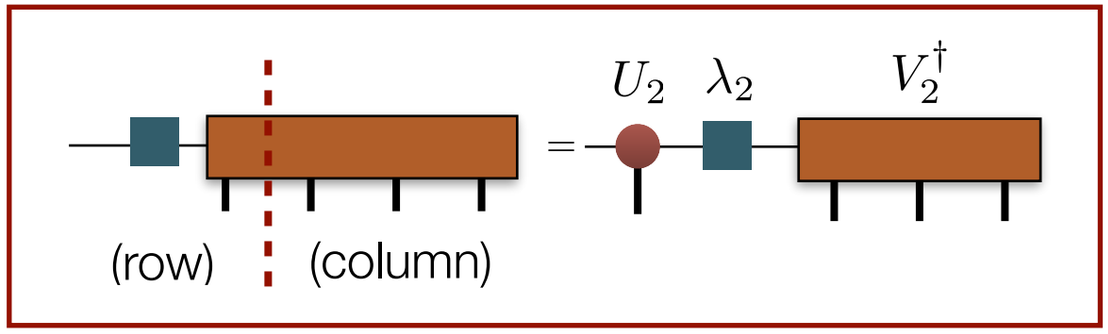
(I. V. Oseledets, SIAM J. Sci. Comput. **33**, 2295 (2011))

行列積状態への（近似なしでの）変換

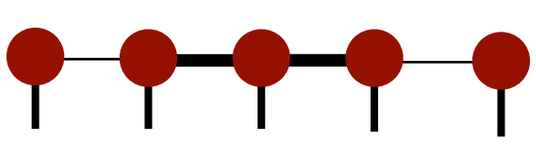
一般のテンソルは特異値分解を繰り返すことでMPSに厳密に変換可能







…



この構成では行列サイズは場所に依存

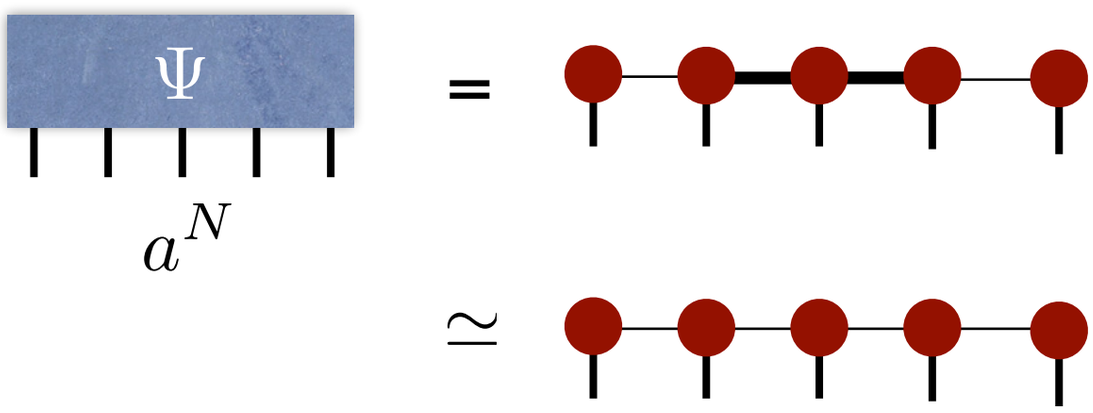
最大の**ボンド次元** = $a^{N/2}$

＊最終的に得られるMPSは、足をどう一次元的に並べるかに依存している

行列積分解による"低ランク近似"

水平方向の繋がりは特異値分解で生成された

→ 行列の場合と同様に「小さい特異値を無視する近似」が可能



a^N

最大のボンド次元= $a^{N/2}$

データ量～ a^N

最大のボンド次元 = χ

データ量 $\propto Na\chi^2$

元のテンソルがボンド次元 χのMPSで精度よく近似できる

→Nに関して指数関数的に大きなデータを線形にまで圧縮できる

＊この近似の精度も、足をどう一次元的に並べるかに依存する

画像圧縮への適用例

例：画像データの圧縮

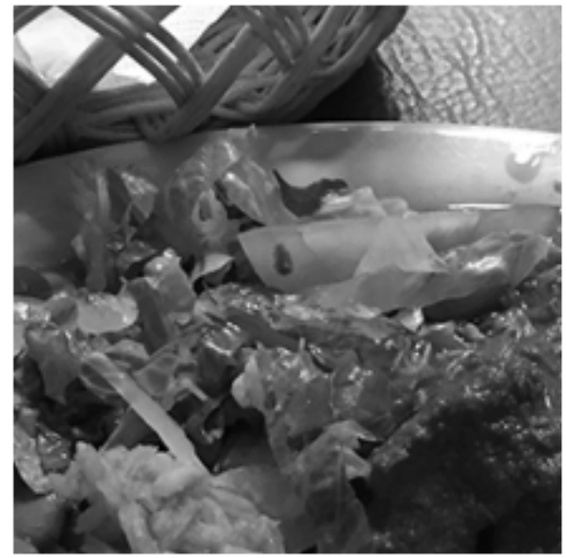
256 × 256 ピクセルの画像

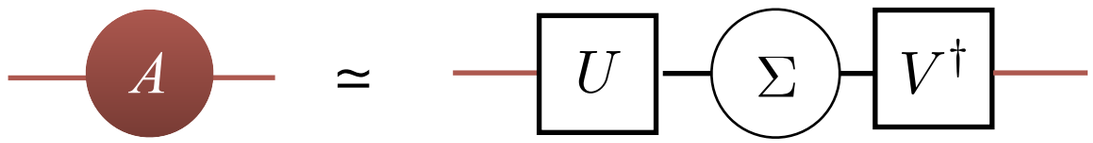
2種類のデータ構造

1. 256 × 256 の行列（画像見たまま）
   - 単純な特異値分解を適用可能

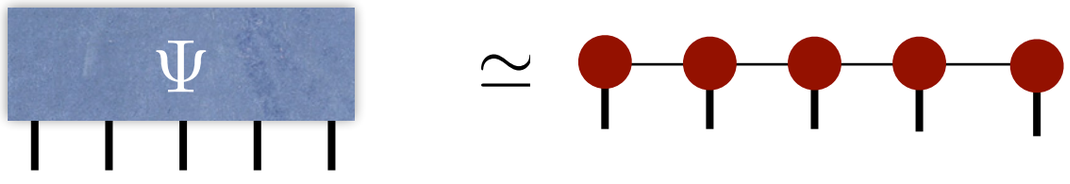
256=2^8

256=2^8

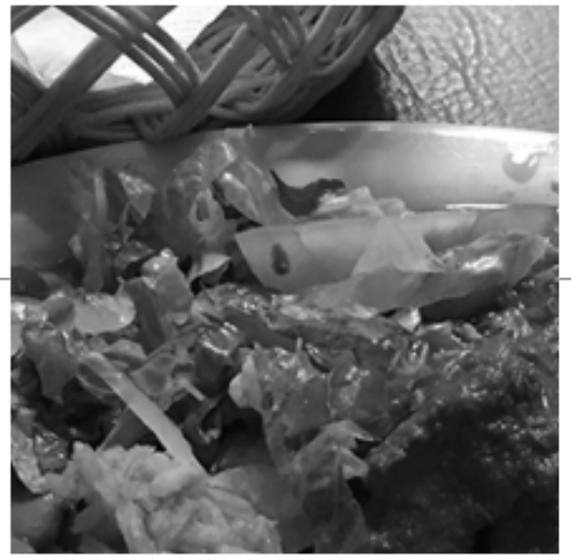




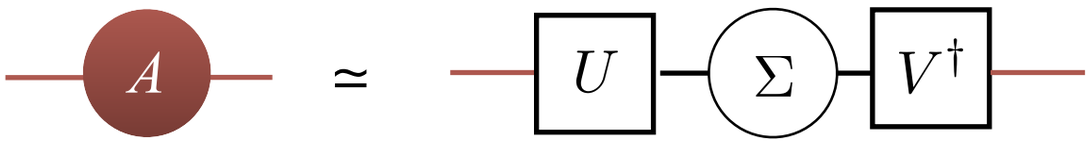
2. 2 × 2 × ...× 2の16本足のテンソル
   - MPSによる近似を適用できる形
   - どのように変形するか、足を並べるのかの自由度がある



特異値分解による画像の圧縮

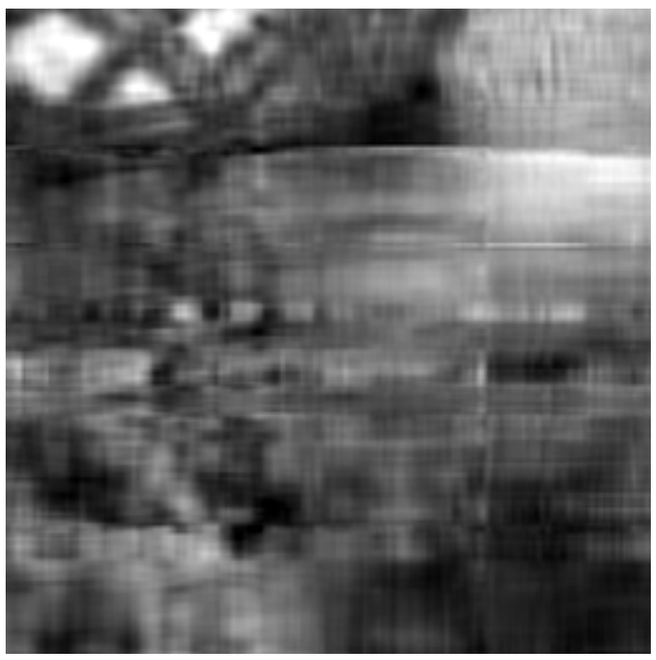


1. 256 × 256 の行列（画像見たまま）



近似精度：$\Delta \equiv \|A - \tilde{A}\| / \|A\|$

データ量：$D \equiv 256 \times \chi + \chi \times 256 + \chi = 513\chi$



$\chi = 10$

$\Delta = 0.2025$

$D = 5130$

$\chi = 20$

$\Delta = 0.1529$

$D = 10260$

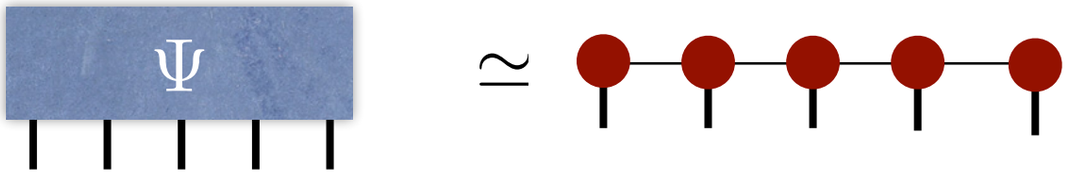
$\chi = 50$

$\Delta = 0.0885$

$D = 25650$

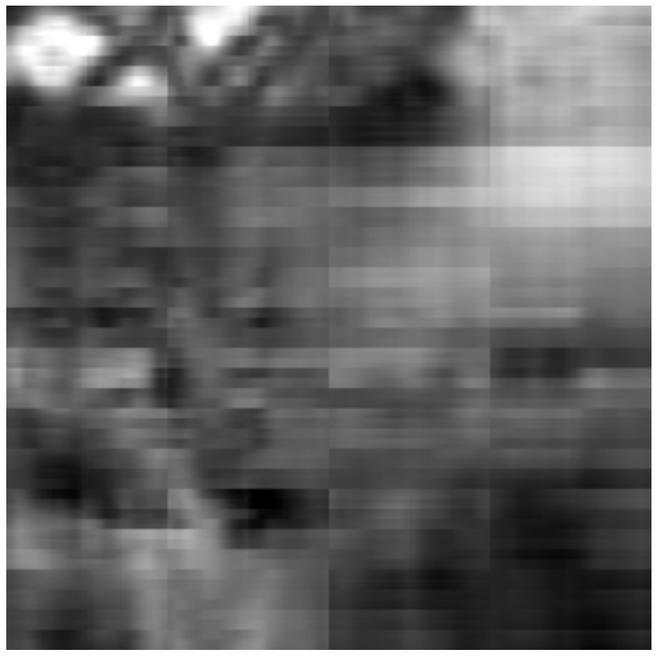
MPSによる画像の圧縮：case 1

2. 2 × 2 × ...× 2の16本足のテンソル



近似精度：$\Delta \equiv \|A - \tilde{A}\|/\|A\|$

データ量：$D \simeq N\chi_{max}^2$



$\chi_{max} = 10$

$\Delta = 0.2405$

$D = 2088$

$\chi_{max} = 20$

$\Delta = 0.1873$

$D = 6760$

$256=2^8$

$256=2^8$

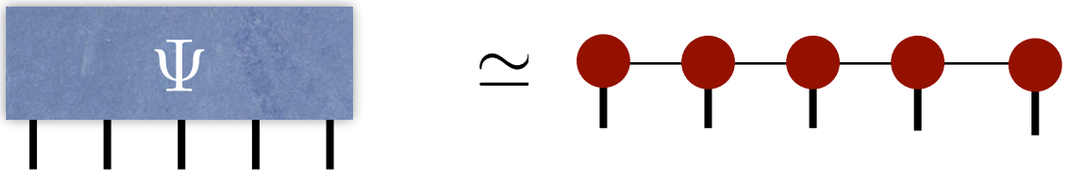
1	2	3	…	255	256
257	258	250	…		
				…	2^{16}

数字を二進数表記して

16本足のテンソル化

画像の圧縮：テンソル化依存性

2. 2 × 2 × ...× 2の16本足のテンソル



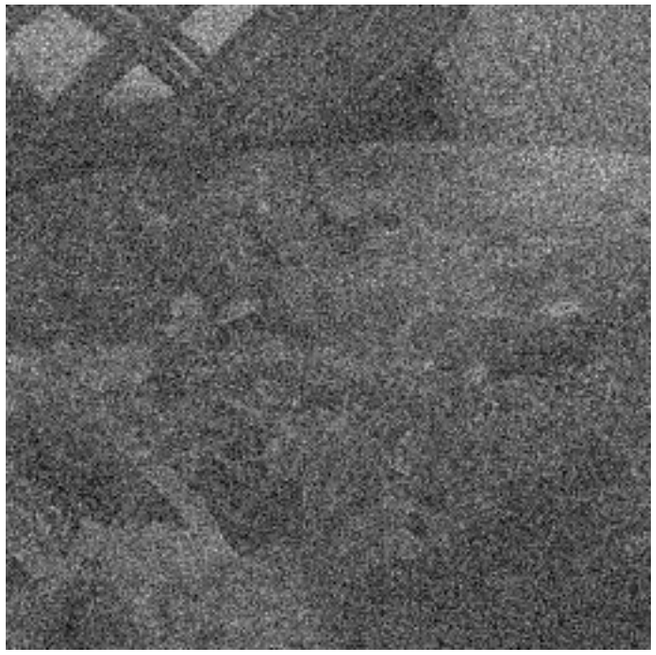
$\simeq$

$$\chi_{max} = 50$$

$$D = 29128$$

case 1

$\Delta = 0.1122$

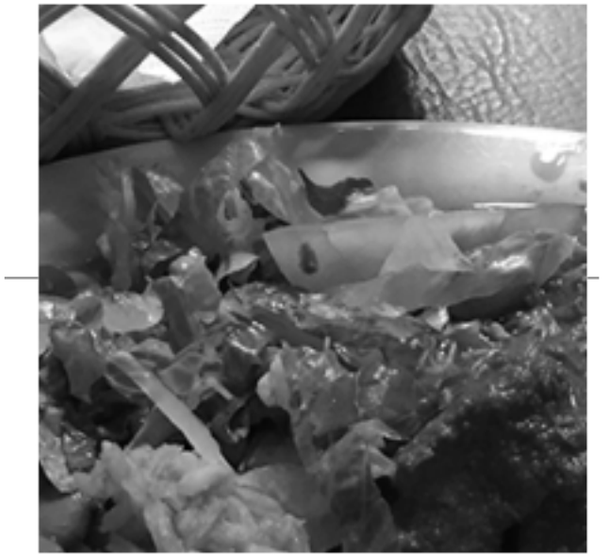


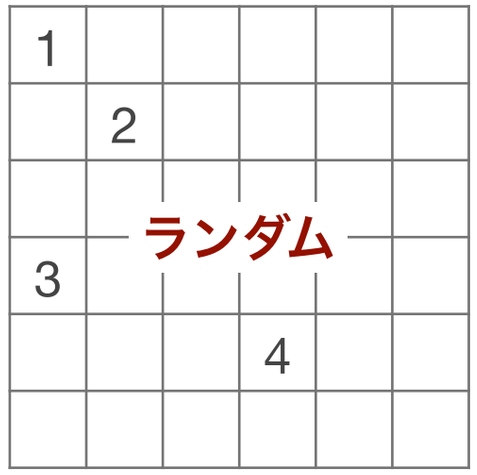
ランダム

$\Delta = 0.4227$

256=2^8

256=2^8



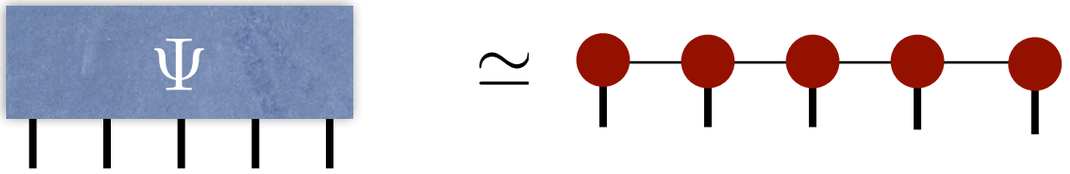


数字を二進数表記して

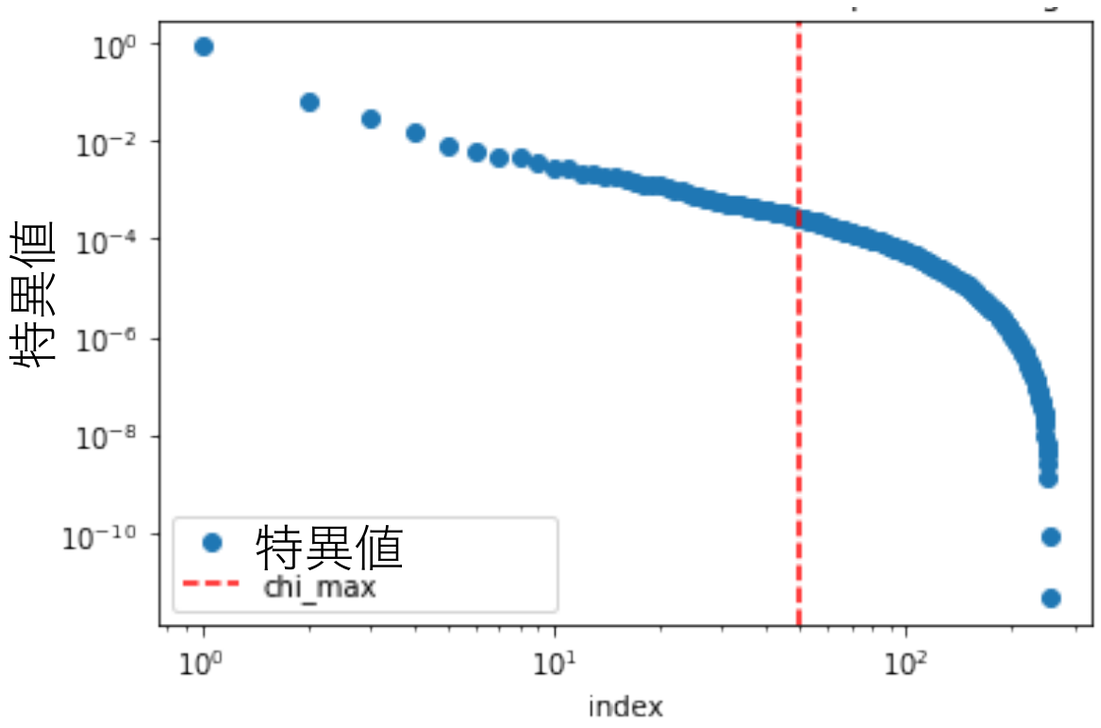
16本足のテンソル化

低ランク近似と特異値スペクトル

2. 2 × 2 × ...× 2の16本足のテンソル

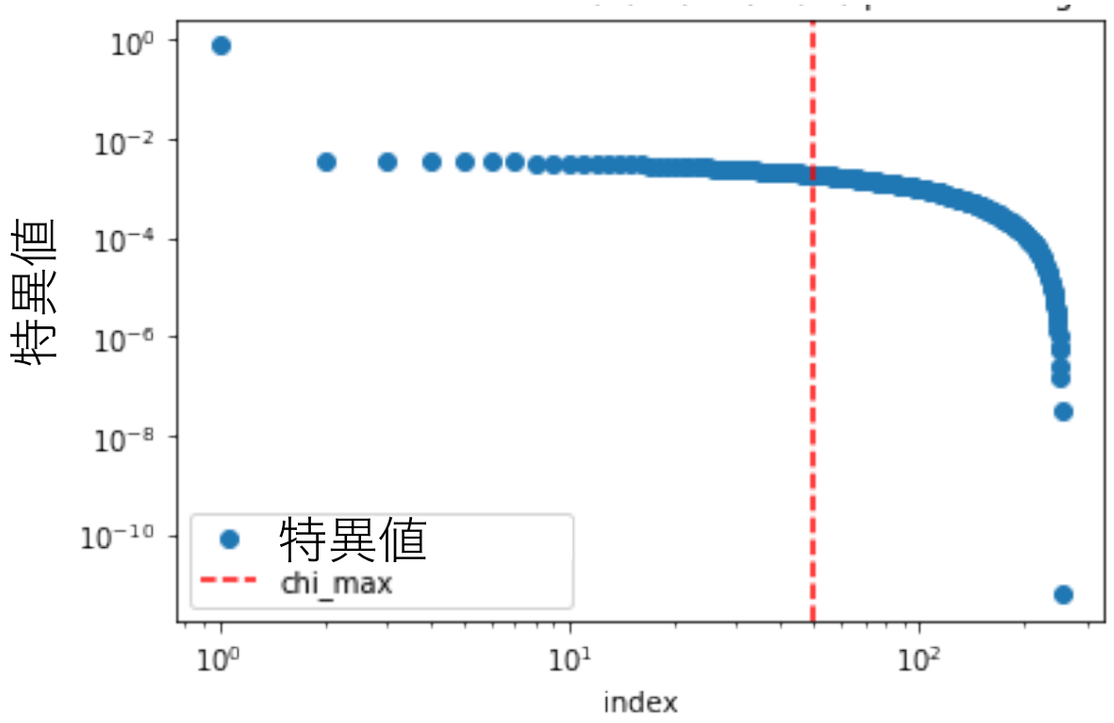


"中央"で分割した時の特異値



case 1

$\Delta = 0.1122$



ランダム

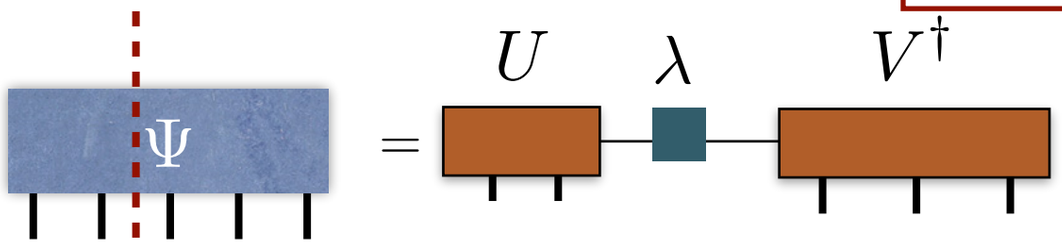
$\Delta = 0.4227$

テンソル化の方法で"エンタングルメント"が違う

エンタングルメントエントロピー

テンソルを"左右"に分解した際の
特異値λの分布

→ 左右の情報の相関（エンタングルメント）を特徴付ける



(row) ⋮ (column)

エンタングルメントエントロピー＝規格化された特異値のエントロピー

$$\tilde{\lambda}_i \equiv \lambda_i / \sqrt{\sum_i \lambda_i^2} \quad \rightarrow \quad S = -\sum_i \tilde{\lambda}_i^2 \log \tilde{\lambda}_i^2$$

$$0 \leq S \leq \log \chi$$ χ：特異値の総数

例：画像の特異値スペクトル

	case 1	random
$\chi_{max} = 50$ での誤差	$\Delta = 0.1122$	$\Delta = 0.4231$
エントロピー	$S = 0.9494$	$S = 1.801$

$\log 256 \simeq 5.545$

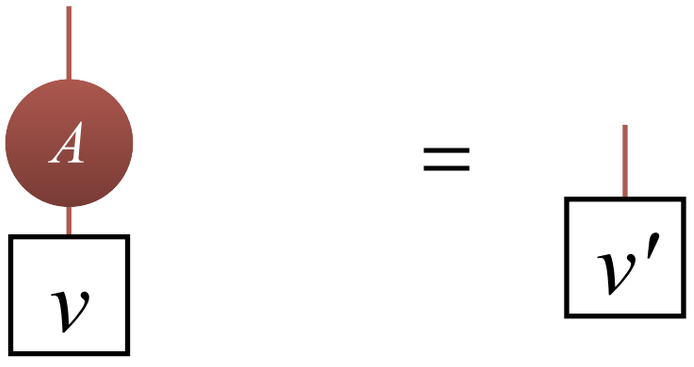
時系列解析への適用に向けて

演算子の行列積分解：行列積演算子

行列のテンソル化と近似

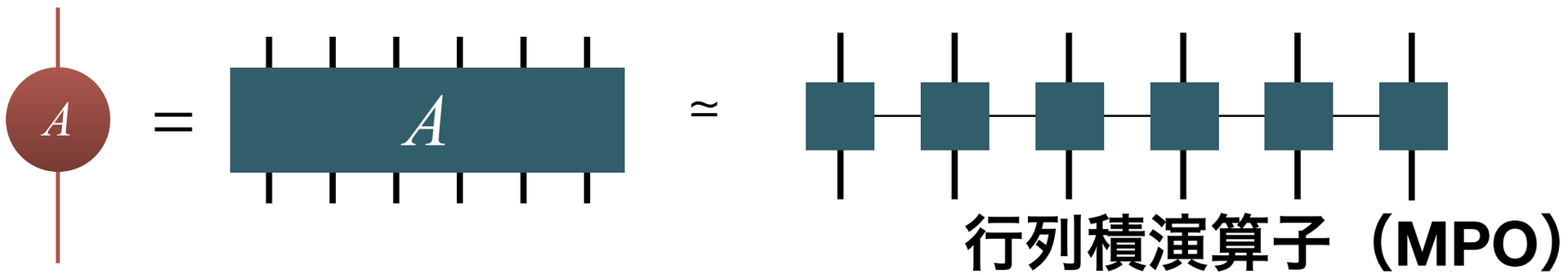
行列＝線形演算子

input と outputがある



この構造を積極的に使ったテンソルネットワーク分解？

input と outputの空間をそれぞれテンソル化した後に、
"組"でテンソル分解



行列積演算子（MPO）

＊MPSとほぼ同じ手続きで近似できる

- input と outputの空間の関係が明確な時に有利
  - 関係が非自明だと、足の組み合わせに任意性→近似精度に影響
- 特異値分解と異なり、行列としてのランクは（必ずしも）低下しない

MPOによるDeep neural networkの圧縮例

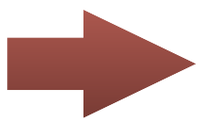
Z.-F. Gao et al, Phys. Rev. Research **2**, 023300 (2020).

ニューラルネットワークの一部

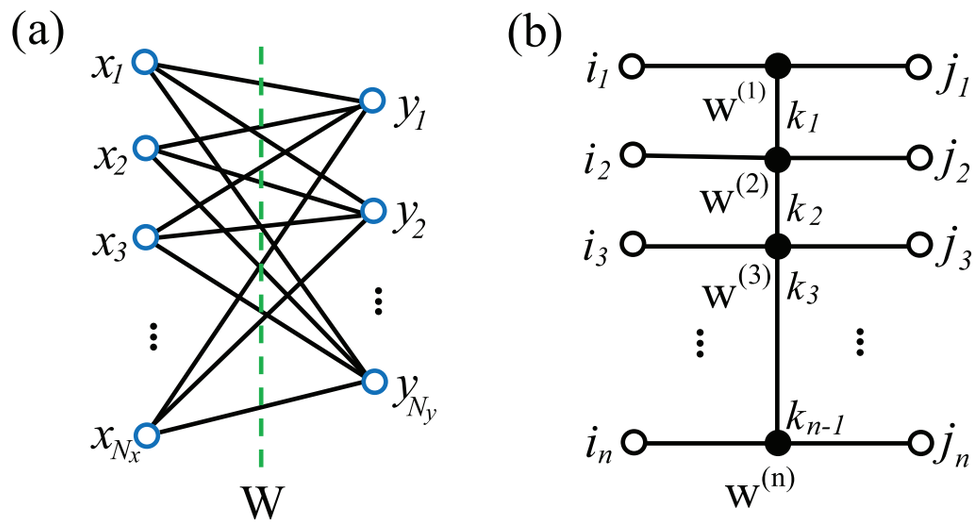
x_i: input neuron

y_i: output neuron

W_{ij}: x と yを結ぶ重み行列



Wを行列積演算子で近似



例：画像識別問題への適用

TABLE I. Test accuracy a and compression ratios ρ obtained in the original and MPO representations of LeNet-5 on MNIST and VGG on CIFAR-10.

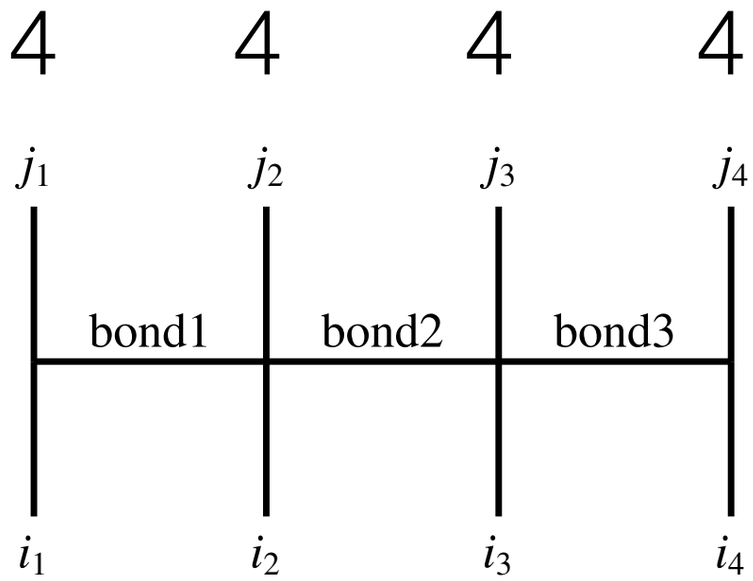
Data set	Network	Original Rep a (%)	MPO-Net a (%)	MPO-Net ρ
MNIST	LeNet-5	99.17 ± 0.04	99.17 ± 0.08	0.05
CIFAR-10	VGG-16	93.13 ± 0.39	93.76 ± 0.16	$\sim$0.0005
	VGG-19	93.36 ± 0.26	93.80 ± 0.09	$\sim$0.0005

a:**精度 (%)**

ρ:**データ圧縮率**

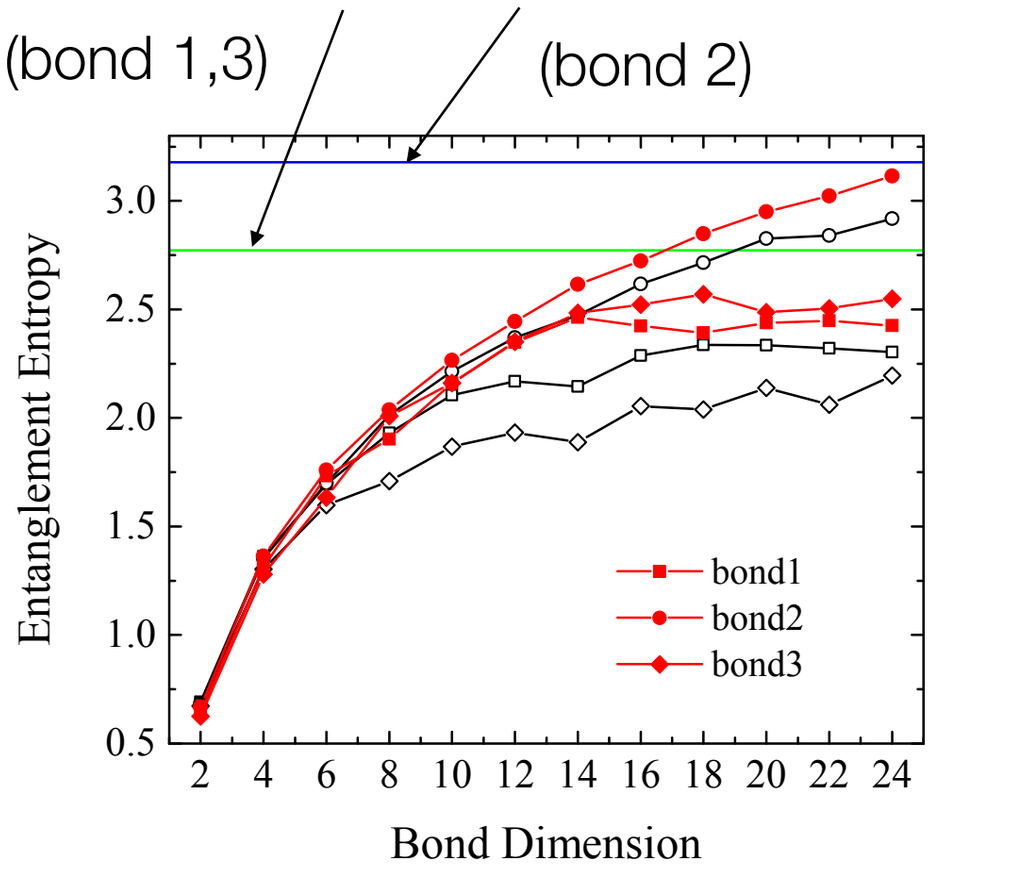
訓練されたMPOのエンタングルメント

Z.-F. Gao et al, Phys. Rev. Research **2**, 023300 (2020).



input: 28 x 28 pixel image

エンタングルメントエントロピーの最大値

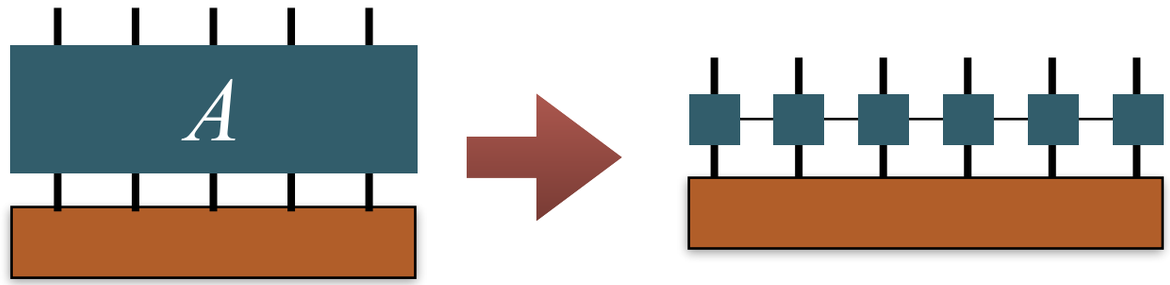


- Fashion-MNIST はMNISTよりも大きなエンタングルメントエントロピー
  - データセットの複雑さに関係？
- Bond 1と 3ではエンタングルメントエントロピーが理論的な最大値よりも小さい値で飽和
  - MPOでの近似が上手くいくことの一つの証拠

Black: MNIST
Red: Fashion-MNIST (MNISTより複雑)

機械学習での行列圧縮のメリット

- 行列の情報を蓄えるためのデータ量を大幅に圧縮できる
  - テンソルの行列積分解と同様
- 行列の演算コストの低下
  - 大きな行列を直接かける代わりに、小さいテンソルを順番にかけることでコストが低下
  - 訓練・予測時間の低減



- 無駄な自由度の削減による最適化（訓練）の効率アップ
  - テンソルネットワーク近似で性能が変わらないことは、元の表現には「無駄」が多いことを示唆
  - 過学習、汎化性能への影響も期待？

効率的なテンソルネットワーク分解のポイント

テンソルネットワーク分解の使い所

テンソルネットワーク分解は、（足の数に対して）指数関数的に大きな自由度を大幅に（足の数に比例する程度に）圧縮できる

→ 対象とするテンソル・行列のサイズが大きいほど、データ圧縮の効果が大きい

使い所の例：

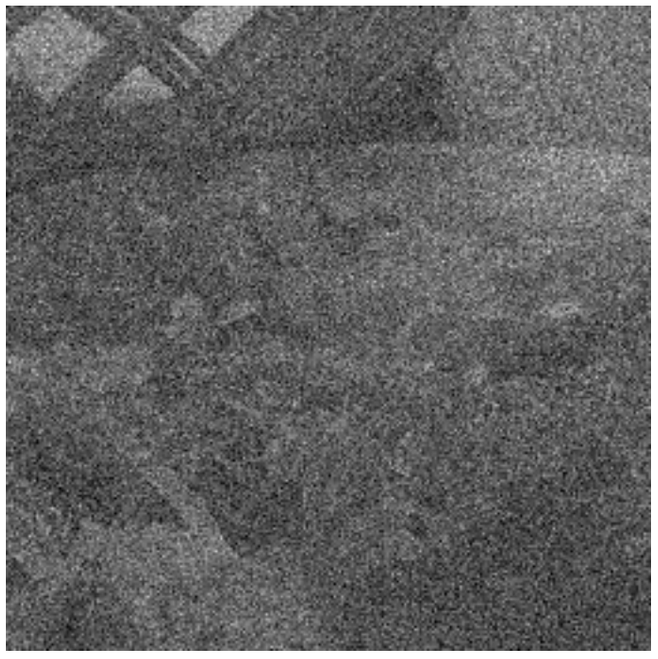
- （機械学習）ネットワーク中で密に結合している部分
- 変数が多すぎて計算（訓練・予測）時間がかかりすぎる場合
- （中間）変数を大幅に増やせば、予測性の向上が期待できる場合

indexの順番とエンタングルメント

「適切なデータの並び」になっていないと、
テンソルネットワーク分解の効率が悪い

元のデータ並びでの近似

ランダムなデータ並びでの近似

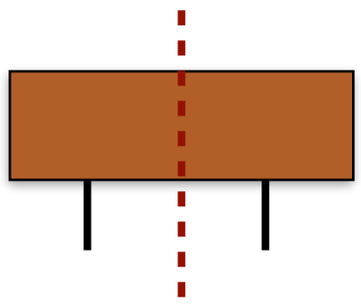


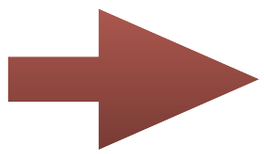
良い分解が可能かどうかは、エンタングルメントで
（ある程度）定量化できる

→ エンタングルメントが小さいところで分解すると良い

indexの順番とdisentangling

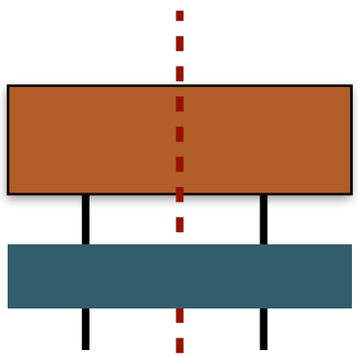
Indexの並び替えは行列で表現可能

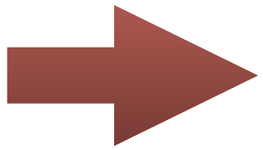




エンタングルメント：大

Indexの並び替え

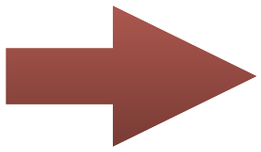




エンタングルメント：小

このようなエンタングルメントを小さくする

（ユニタリ）演算子をDisentangler と呼ぶ

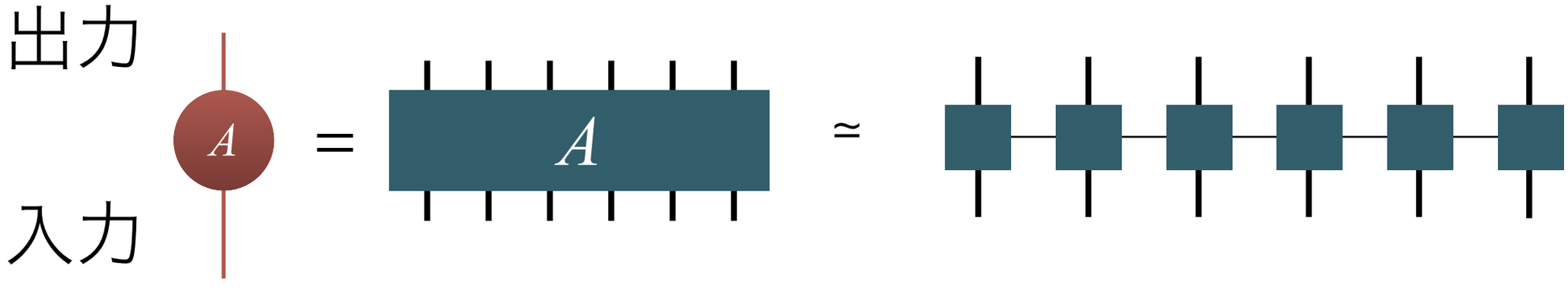


「事前知識（cf. 画像）」や「最適化」などで、

適切なDisentanglerを見つけられれば分解効率up

演算子の入力・出力空間

演算子のテンソルネットワーク分解では、入力・出力の対応も重要



演算子がニューラルネットワークの一部の場合には、一般に、

入力側と出力側の空間の関係が非自明

→ 入出力を整合させてテンソル化するのは非常に難しい

これを避けるには...

- 入力と出力の空間が同じ場合を考える
  - 今回の演習で扱うRNNは、おそらく、そうなっている
- 学習の段階からテンソルネットワーク構造を仮定する
  - 学習により自動的に最適なテンソル化が選ばれる

まとめ

- テンソルネットワーク分解を用いることで、データ量を大幅に圧縮できる場合がある
- この分解を機械学習などに用いると、学習・予測の計算量を削減できる可能性がある
- 効率的にテンソルネットワーク分解できるかどうかは、問題依存
  - ターゲットの行列・テンソルは十分に大きいか？
  - エンタングルメントが小さい（ところがある）か？
  - エンタングルメントが小さくなるようにテンソル化できるか？