2025/06/20
物理屋のための機械学習講義 第17回

実習用

機械学習の量子多体問題への応用

東北大学　金属材料研究所
野村 悠祐

研究室ウェブサイト https://www.nomura-lab.imr.tohoku.ac.jp

E-mail yusuke.nomura@tohoku.ac.jp

IMR 東北大学 金属材料研究所
Institute for Materials Research, Tohoku University

この資料の目標

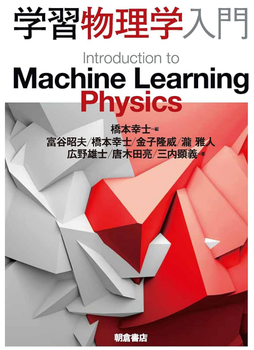
・人工ニューラルネットワークによる変分法の最低限の原理を理解する

・（少なくとも）最急降下法による最適化を、コードを一から組めるくらいの理解度にする

（数値手法を理解すること）＝（コードを実装できること）　☜　完全な私見です

https://github.com/yusukenomura/School_2024　一次元ハイゼンベルク模型（Fortran）

https://github.com/ryuikaneko/textbook_2024_machine_learning_physics　一次元横磁場イジング模型（Python）



人工ニューラルネットワークによる変分法

機械学習でプラクティカルに行われていること＝非線形関数（ここでは人工ニューラルネットワークを使う）の最適化

機械学習タスクで設定しなければいけないこと　（変分法の場合）

1. 人工ニューラルネットワークへのデータの入力方法を定義する

→　粒子の実空間配置

2. 人工ニューラルネットワークの出力を定義する

→　波動関数

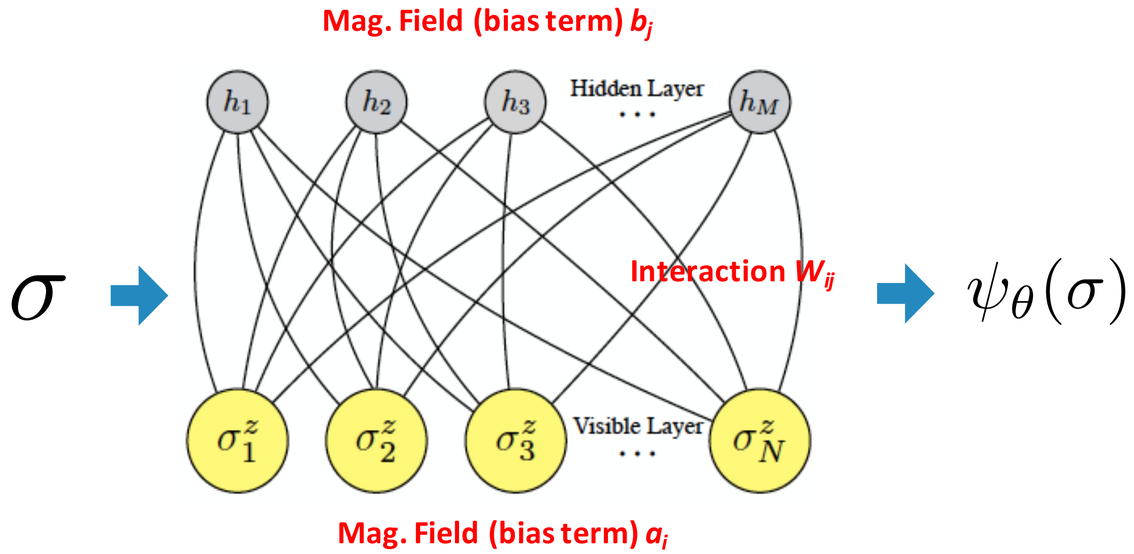
3. コスト関数を定義する

→　エネルギー期待値

4. パラメータの最適化方法を選択する

→　変分法ではnatural gradient法が使われることが多い（後述）

設定1, 2の例：制限ボルツマンマシン(RBM)による量子状態表現



Carleo and Troyer Science 355, 602 (2017)

RBM (restricted Boltzmann machine) 波動関数

$$\psi_\theta(\sigma) = \sum_h \exp\left(\sum_i a_i \sigma_i^z + \sum_{i,j} W_{ij}\sigma_i^z h_j + \sum_j b_j h_j\right)$$

$$= e^{-E_{\mathrm{RBM}}(\sigma^z, h)} \quad \text{(Boltzmann weight)}$$

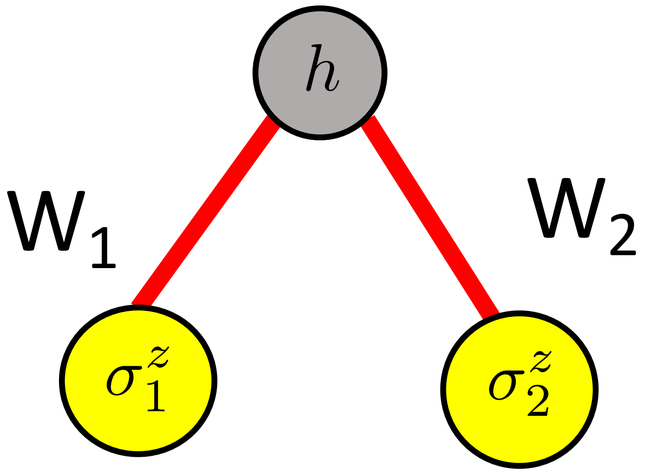
$\sigma = (\sigma_1^z, \sigma_2^z, \ldots, \sigma_N^z)$: スピン配置

$h_j = \pm 1$: 隠れスピン

- 入力＝スピン配置、出力＝波動関数
- 隠れスピンとの結合を通じて量子相関を表す
- 隠れスピンの数が無限大の極限で量子状態を任意の精度で表現可能（普遍近似性能）

→ α = M/N (hidden variable density)が精度の制御パラメータ

設定1, 2に関する具体的計算例：RBM波動関数（N=2, M=1の場合）



$$\theta = (W_1, W_2)$$

$$\begin{aligned}\psi_\theta(\sigma_1^z, \sigma_2^z) &= \sum_{h=\pm 1} \exp(W_1\sigma_1^z h + W_2\sigma_2^z h) \\ &= \exp(W_1\sigma_1^z + W_2\sigma_2^z) + \exp(-W_1\sigma_1^z - W_2\sigma_2^z) \\ &= 2\cosh(W_1\sigma_1^z + W_2\sigma_2^z)\end{aligned}$$

$$\psi_\theta(\uparrow,\uparrow) = \psi_\theta(\downarrow,\downarrow) = 2\cosh(W_1 + W_2)$$

$$\psi_\theta(\uparrow,\downarrow) = \psi_\theta(\downarrow,\uparrow) = 2\cosh(W_1 - W_2)$$

一般の場合

$$\psi_\theta(\sigma) = \exp\left(\sum_i a_i\sigma_i^z\right) \times \prod_j 2\cosh\left(b_j + \sum_i W_{ij}\sigma_i^z\right)$$

設定3, 4：コスト関数の設定と最適化

コスト関数＝エネルギー期待値

$$E_\theta = \frac{\langle \psi_\theta | \mathcal{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}$$

最適化に必要なもの＝勾配ベクトル（コスト関数の微分）

$$g_k^{(t)} = \left. \frac{\partial E_\theta}{\partial \theta_k} \right|_{\theta=\theta^{(t)}}$$

$\theta^{(t)}$: t番目のステップ時のパラメータセット

最急降下法 Steepest descent (SD) (or gradient descent)

$$\theta_k^{(t+1)} = \theta_k^{(t)} - \eta g_k^{(t)}$$

η : learning rate　☜ ハイパーパラメータ（人の手で決める）

人工ニューラルネットワークを用いた変分法の計算手順

1. **初期化**：人工ニューラルネットワークによって変分波動関数を定義し、パラメータを初期化する

※パラメータの初期化には小さな値の乱数を用いることが多い

2. **学習**：コスト関数（エネルギー）を最小化するようにパラメータを最適化する

3. **ポストプロセス**：得られた波動関数を用いて、物理量を計算する

※乱数を用いてパラメータを初期化した場合は、複数の初期パラメータを用いて最適化を行い
一番エネルギーの下がったものを採用する

実習の準備

1. MateriApps LIVE! を起動

2. MateriApps LIVE! 上の適当なディレクトリのもとで以下のコマンドを実行する

```
git clone https://github.com/yusukenomura/School_2024.git
```

```
user@malive:~$ git clone https://github.com/yusukenomura/School_2024.git
Cloning into 'School_2024'...
remote: Enumerating objects: 109, done.
remote: Counting objects: 100% (109/109), done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 109 (delta 34), reused 72 (delta 24), pack-reused 0
Receiving objects: 100% (109/109), 78.63 KiB | 267.00 KiB/s, done.
Resolving deltas: 100% (34/34), done.
```

3. School_2024/RBM_solverの下に移動する

```
cd School_2024/RBM_solver/
```

```
user@malive:~$ cd School_2024/RBM_solver/
user@malive:~/School_2024/RBM_solver$ ls
README.txt  examples  src
```

実習１：１次元ハイゼンベルク模型（8サイト, 16サイト）、SD法による最適化

ソースコード：　`./src/exact_sampling/SD`

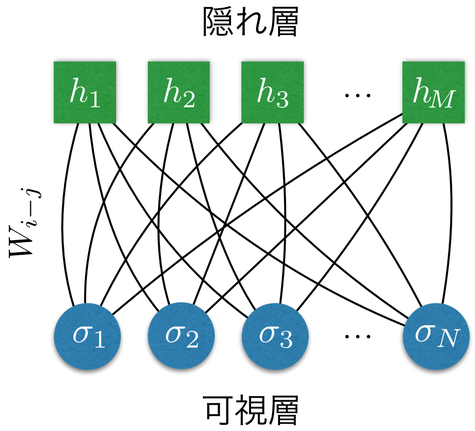
実習場所：　`./examples/1D_AF_chain/exact_sampling/08site`
`./examples/1D_AF_chain/exact_sampling/16site`

実行コマンド：　`./RBM_solver_SD.x > log`

必要なファイル：　`RBM.input`（インプット）, `spin_configurations.txt`（全スピン配置パターンのリスト）

ハミルトニアン：　$\mathcal{H} = \sum_i (-\sigma_i^x \sigma_{i+1}^x - \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z)$

変分波動関数：



$$\alpha(\text{hidden variable density}) = M/N = 1$$

$$\psi_\theta(\sigma) = \prod_j 2\cosh\left(\sum_i W_{i-j}\sigma_i^z\right)$$

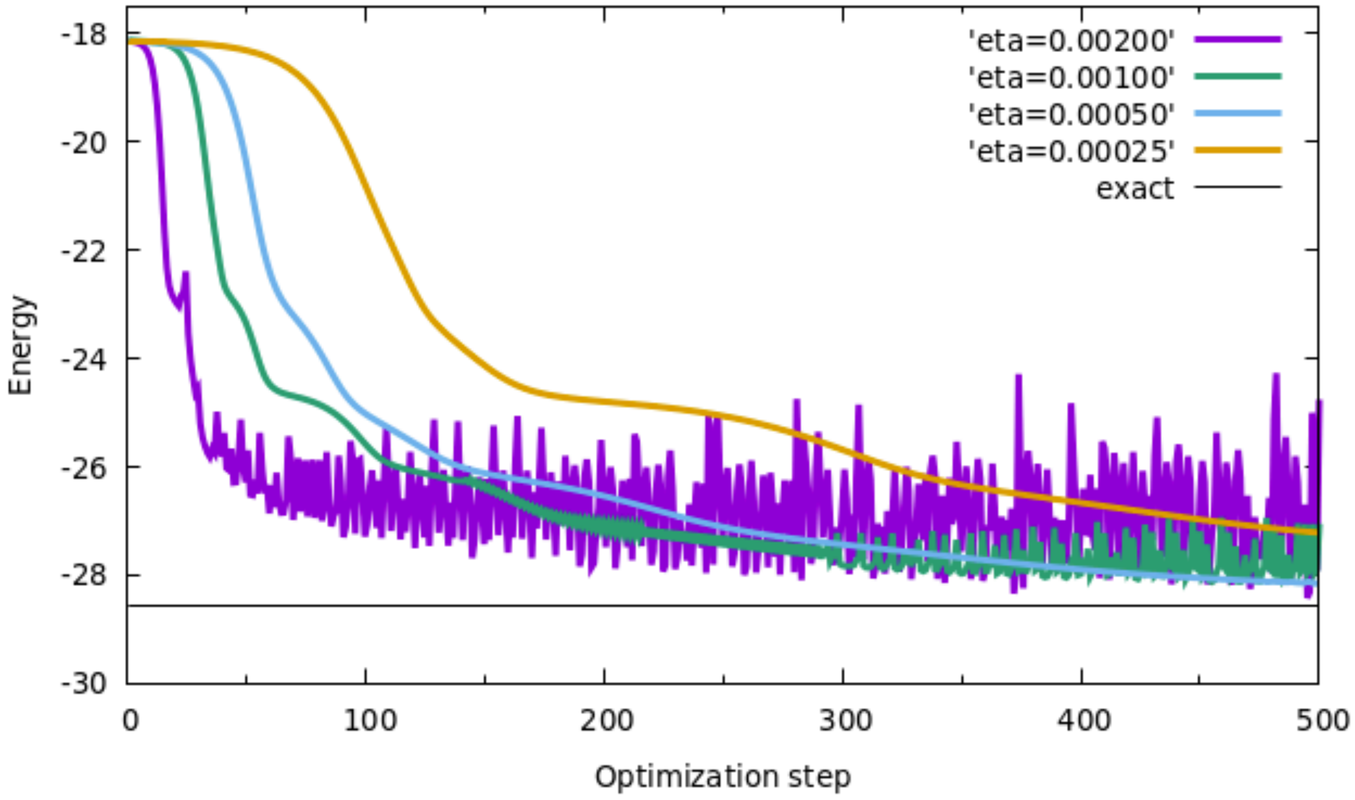
Carleo and Troyer Science **355**, 602 (2017)とほぼ同じ設定

課題1-1：最適化（学習）の安定性を調べる

RBM.input内のlearning rate（η=delta_tau）を変えながら、(コスト関数) vs (最適化のIteration数) をプロットする

関連するファイル：`Energy_vs_Iteration.txt`

16サイトに対する結果



※ 乱数シードの値をシステムクロックで設定しているので、完全に同一の結果が得られるわけではない。

コスト関数（=エネルギー）の具体的な計算方法＋コードの解説

$$\langle \mathcal{H} \rangle = \frac{\langle \psi_\theta | \mathcal{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} = \frac{\sum_{\sigma\sigma'} \psi_\theta^*(\sigma) \mathcal{H}_{\sigma\sigma'} \psi_\theta(\sigma')}{\sum_\sigma |\psi_\theta(\sigma)|^2} = \frac{\sum_\sigma |\psi_\theta(\sigma)|^2 E_{\mathrm{loc}}(\sigma)}{\sum_\sigma |\psi_\theta(\sigma)|^2}$$

$$E_{\mathrm{loc}}(\sigma) = \sum_{\sigma'} \mathcal{H}_{\sigma\sigma'} \frac{\psi_\theta(\sigma')}{\psi_\theta(\sigma)}$$

```
175       Eloc_x = 0d0
176       do i1 = 1, N
177         i2 = i1 + 1
178         if( i1 == N ) i2 = 1
179         !
180         ! SzSz contribution
181         !
182         Eloc_x = Eloc_x + x(i1)*x(i2)
183         !
184         ! ( SxSx + SySy ) = 2 * ( S+S- + S-S+ ) contribution
185         !
186         if( x(i1) /= x(i2) ) then
187           xp(:) = x(:)
188           xp(i1) = -xp(i1)
189           xp(i2) = -xp(i2)
190           call calc_theta(xp,thetap)
191           call calc_amplitude_RBM(M,thetap,psi_xp)
192           Eloc_x = Eloc_x - 2d0*psi_xp/psi_x
193         end if
194
195       end do ! i1
196       E = E + p_x * Eloc_x
```

O(N)でEloc(σ)が計算できる！ ☜相互作用が局所的なため

※(コード内のx) = σ

$= H_{\sigma\sigma'}$

→ 分子の計算（分母は後で割る）

$= |\psi_\theta(\sigma)|^2$

コスト関数の微分の具体的な計算方法＋コードの解説

$$g_k = \left(\frac{\langle \psi_\theta | \mathcal{H} | \partial_k \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} + \text{c.c.} \right) - \left(\frac{\langle \psi_\theta | \mathcal{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \frac{\langle \psi_\theta | \partial_k \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} + \text{c.c.} \right)$$
$$= 2\mathrm{Re}\langle \mathcal{H} O_k \rangle - 2\langle \mathcal{H} \rangle \mathrm{Re}\langle O_k \rangle$$

$$O_k = \sum_\sigma |\sigma\rangle O_k^{\mathrm{loc}}(\sigma) \langle \sigma|$$
$$O_k^{\mathrm{loc}}(\sigma) = \frac{\partial_k \psi_\theta(\sigma)}{\psi_\theta(\sigma)}$$

```
200       k = 0
201       do f  = 1, alpha
202       do iw = 0, N-1
203
204         k = k + 1
205         Ovec_loc_x(k) = 0d0
206
207         do jj = 1, N
208           j = (f-1)*N+jj
209           i = jj + iw
210           if( i > N ) i = i - N
211           Ovec_loc_x(k) = Ovec_loc_x(k) + tanh(theta(j))*dble(x(i))
212         end do ! jj
213
214       end do ! iw
215       end do ! f
216       if( k /= Nv ) stop 'k /= Nv'
217       !
218       ! summation over x for gvec and Ovec
219       !
220       do k = 1, Nv
221         gvec(k) = gvec(k) + 2d0 * p_x * Eloc_x * Ovec_loc_x(k)
222         Ovec(k) = Ovec(k) + p_x * Ovec_loc_x(k)
223       end do ! k
```

coshの微分はsinhであることに注意
（coshで割るとtanhがでてくる）

< H O_k >の分子の計算

< O_k >の分子の計算

$$\langle O_k \rangle = \frac{\langle \psi_\theta | \partial_k \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} = \frac{\sum_\sigma |\psi_\theta(\sigma)|^2 O_k^{\mathrm{loc}}(\sigma)}{\sum_\sigma |\psi_\theta(\sigma)|^2}$$

ここまで：SD法を用いて変分原理に従った計算を行った。

次の課題：安定した最適化を行うには？　→ Stochastic reconfiguration (SR)法

最適化手法の改善：Stochastic reconfiguration法

S. Sorella, Phys. Rev. B **64**, 024512 (2001)

虚時間発展を変分波動関数の表現能力の範囲内でできるだけ正確に再現する

$$\delta\theta = \arg\min_{\delta\theta} \mathcal{F}\left(e^{-2\delta_\tau \mathcal{H}}|\psi_\theta\rangle, |\psi_{\theta+\delta\theta}\rangle\right)$$

$\mathcal{F}$ ：フビニ・スタディ計量

$\delta\theta$：パラメータ変化　$e^{-2\delta_\tau \mathcal{H}}|\psi_\theta\rangle$：厳密な発展　$|\psi_{\theta+\delta\theta}\rangle$：変分状態による近似

$$= -\delta_\tau S^{-1} \partial_\theta \langle \mathcal{H} \rangle$$

S^{-1}：計量テンソル　$\partial_\theta\langle\mathcal{H}\rangle$：勾配

虚時間発展の性質：

$$e^{-\tau\mathcal{H}}|\psi^{(0)}\rangle = \sum_i e^{-\tau E_i} c_i^{(0)} |\phi_i\rangle$$ 固有状態

$$= e^{-\tau E_0} \sum_i e^{-\tau(E_i - E_0)} c_i^{(0)} |\phi_i\rangle$$

$$= e^{-\tau E_0} c_0^{(0)} |\phi_0\rangle \quad (\tau \to \infty)$$

もう少し詳しく書くと

$$\delta\theta_k^{(t)} = -\delta_\tau \sum_l \left(S^{(t)}\right)^{-1}_{kl} g_l^{(t)}$$

cf. Steepest descent (SD) 法

$$\delta\theta_k^{(t)} = -\delta_\tau g_k^{(t)}$$

※ learning rateをδ_τとおいた

S : フビニ・スタディ計量テンソル（量子幾何テンソルの実部）

$$\mathcal{F}^2[|\psi_{\theta+\delta\theta}\rangle, |\psi_\theta\rangle] = \sum_{kl} S_{kl}\delta\theta_k\delta\theta_l$$

$$S_{kl} = \mathrm{Re}\left(\frac{\langle\partial_k\psi_\theta|\partial_l\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} - \frac{\langle\partial_k\psi_\theta|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}\frac{\langle\psi_\theta|\partial_l\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle}\right)$$

Quantum geometric tensor (量子幾何テンソル)

$$\mathcal{F}[|\psi\rangle, |\phi\rangle] := \arccos\sqrt{\frac{\langle\psi|\phi\rangle\langle\phi|\psi\rangle}{\langle\psi|\psi\rangle\langle\phi|\phi\rangle}}$$

※ 機械学習分野の自然勾配法 (natural gradient)と本質的に等価　S.-I. Amari, Neural Comput. **10**, 251 (1998).

（おまけ）練習問題

ブロッホ球上の量子状態（以下）に対して、量子幾何テンソルとフビニ・スタディ計量テンソルの値を求めよ。

パラメータはθ、φの二つなので、それぞれ2x2の行列となる。

$$|\psi(\theta,\phi)\rangle = \cos\frac{\theta}{2}|\uparrow\rangle + e^{i\phi}\sin\frac{\theta}{2}|\downarrow\rangle$$

$$Q = \frac{1}{4}\begin{pmatrix} 1 & i\sin\theta \\ -i\sin\theta & \sin^2\theta \end{pmatrix} \qquad S = \frac{1}{4}\begin{pmatrix} 1 & 0 \\ 0 & \sin^2\theta \end{pmatrix}$$

$$\mathcal{F}^2\Big[\,|\psi(\theta+\delta\theta,\phi+\delta\phi)\rangle, |\psi(\theta,\phi)\rangle\,\Big] = \frac{1}{4}(\delta\theta)^2 + \frac{1}{4}(\sin\theta\delta\phi)^2$$

実習２：１次元ハイゼンベルク模型（8サイト, 16サイト）、SR法による最適化

ソースコード：　`./src/exact_sampling/SR`

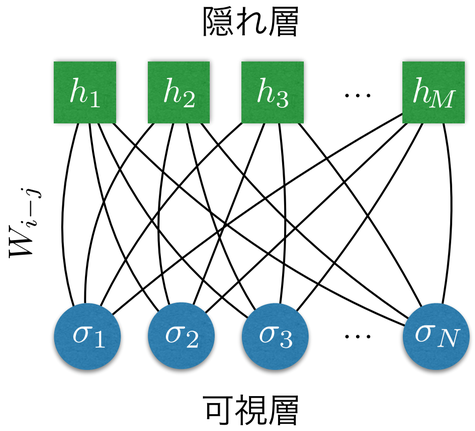
実習場所：　`./examples/1D_AF_chain/exact_sampling/08site`
`./examples/1D_AF_chain/exact_sampling/16site`

実行コマンド：　`./RBM_solver_SR.x > log`

必要なファイル：　`RBM.input`（インプット）, `spin_configurations.txt`（全スピン配置パターンのリスト）

ハミルトニアン：　$\mathcal{H} = \sum_i (-\sigma_i^x \sigma_{i+1}^x - \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z)$

変分波動関数：



$$\alpha(\text{hidden variable density}) = M/N = 1$$

$$\psi_\theta(\sigma) = \prod_j 2\cosh\left(\sum_i W_{i-j}\sigma_i^z\right)$$

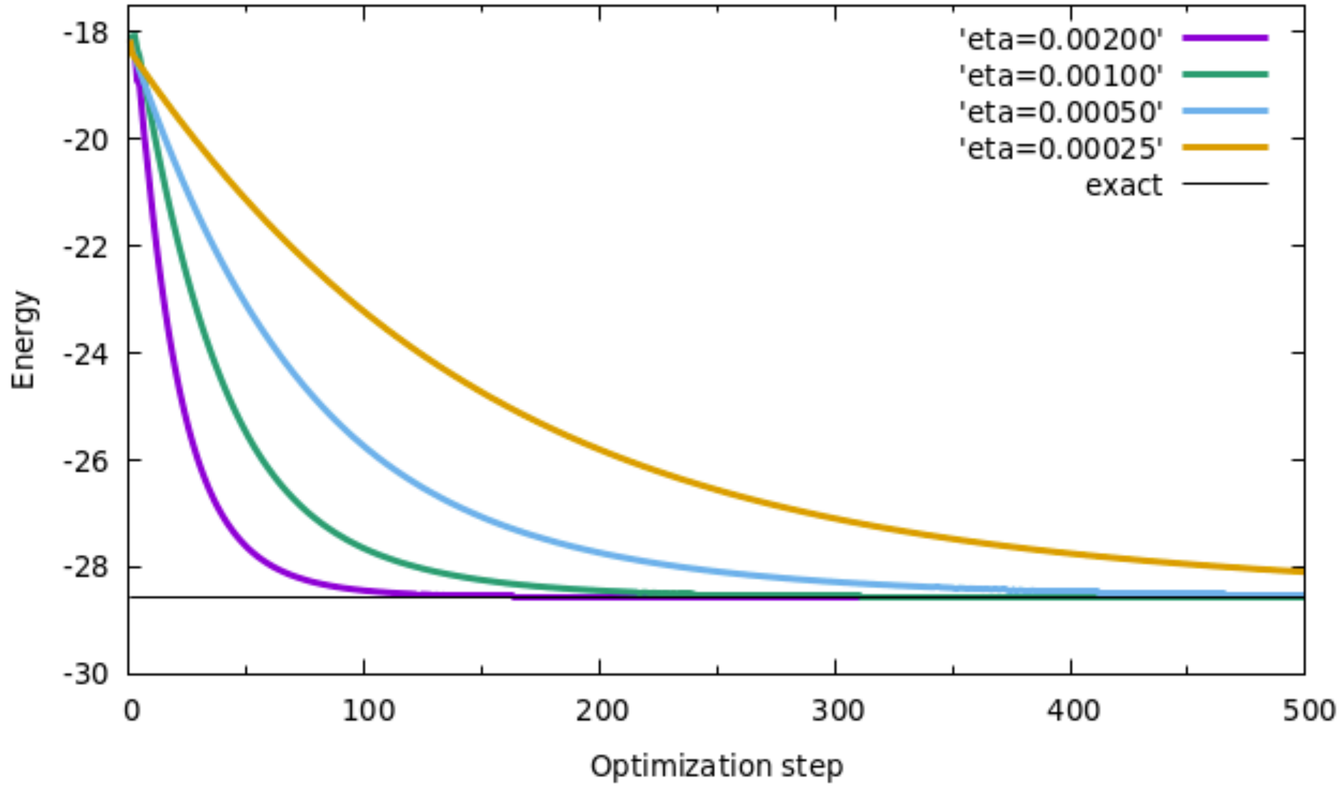
Carleo and Troyer Science **355**, 602 (2017)とほぼ同じ設定

課題2-1：最適化（学習）の安定性を調べる

RBM.input内のlearning rate（η=delta_tau）を変えながら、(コスト関数) vs (最適化のIteration数) をプロットする

関連するファイル：`Energy_vs_Iteration.txt`

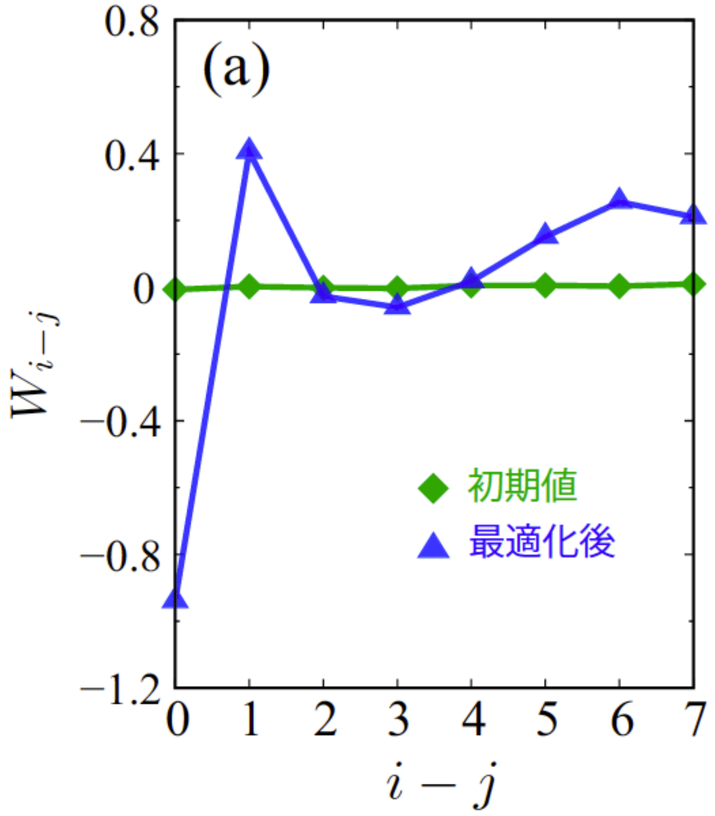
16サイトに対する結果



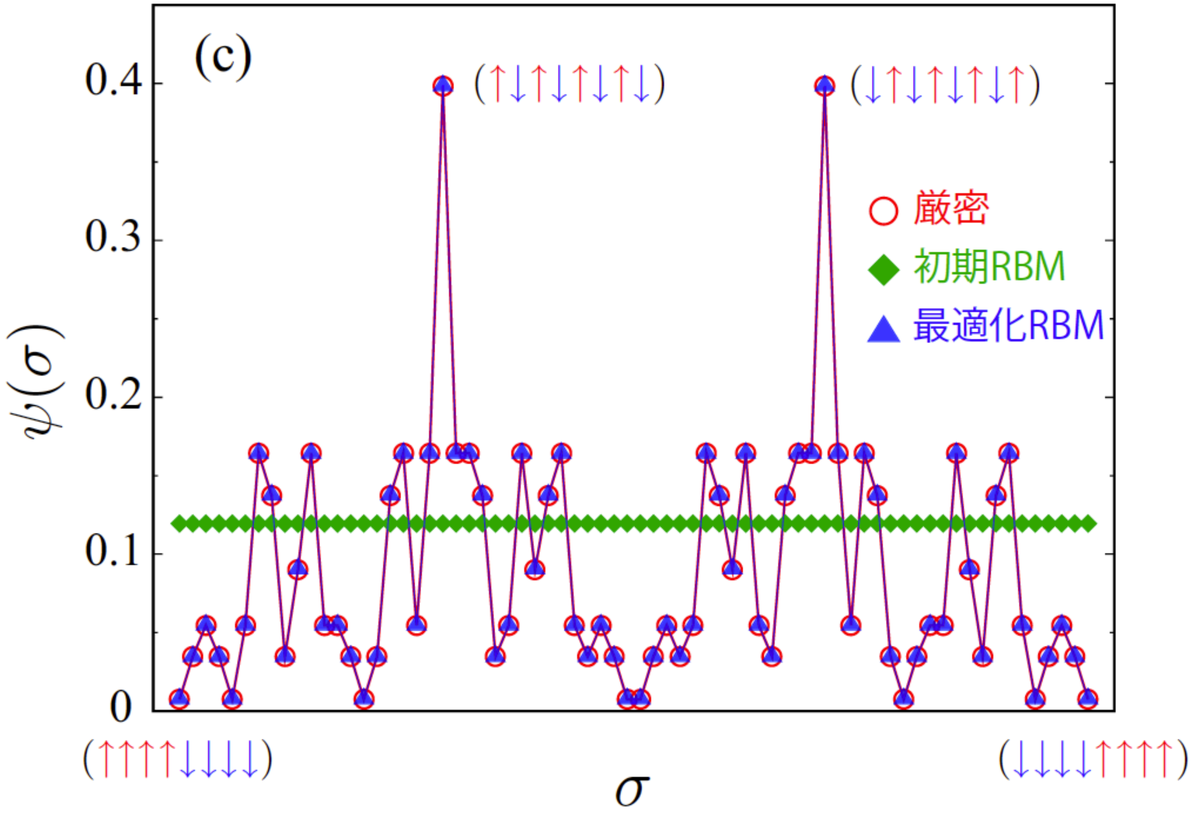
※乱数シードをシステムクロックで設定しているので、同一の結果が得られるわけではない。SD法よりラン毎のばらつきは少ないはず。

課題2-2：最適化前と後のパラメータと波動関数（出力）を調べる

RBMのパラメータ



RBMの入力σと出力$\psi(\sigma)$の関係



関連するファイル：

```
initial_W.txt
optimized_W.txt
```

```
initial_wf.txt
optimized_wf.txt
reference/exact_ground_state.txt
```

ここまで：SR法の導入によって、最適化が安定化する様子をみた。

次の課題：より大きなシステムサイズを計算するには？　→ モンテカルロ法の導入

※ これまでの例では、物理量計算に必要なスピン配置の和は厳密に計算していた（下は物理量の例としてエネルギーの計算の仕方を示す）

$$\langle \mathcal{H} \rangle = \frac{\sum_\sigma |\psi_\theta(\sigma)|^2 \, E_{\mathrm{loc}}(\sigma)}{\sum_\sigma |\psi_\theta(\sigma)|^2}$$

モンテカルロ法の導入

モンテカルロ法による期待値計算（エネルギーを例にとって説明）

$$\langle \mathcal{H} \rangle = \frac{\sum_\sigma |\psi_\theta(\sigma)|^2 E_{\mathrm{loc}}(\sigma)}{\sum_\sigma |\psi_\theta(\sigma)|^2} \approx \frac{1}{N_\mathrm{s}} \sum_{s=1}^{N_\mathrm{s}} E_{\mathrm{loc}}(\sigma_s)$$

※ $|\psi_\theta(\sigma)|^2$ を重みとしたサンプル生成を行う

```
154         do iupdate = 1, N
155           call spin_flip_candidate(x,i1,i2)
156           xp(:) = x(:)
157           xp(i1) = -xp(i1)
158           xp(i2) = -xp(i2)
159           call calc_theta(xp,thetap)
160           call calc_amplitude_RBM(M,thetap,psi_xp)
161           if( grnd() > (psi_xp/psi_x)**2 ) cycle ! reject
162           !
163           ! accept => update configuration
164           !
165           x(:) = xp(:)
166           theta(:) = thetap(:)
167           psi_x = psi_xp
168         end do ! iupdate
```

total Sz=0を守るような配置の中でのサンプル更新の提案

詳細釣り合い（grnd()は乱数）

実習３：１次元ハイゼンベルク模型（16サイト）、モンテカルロ法による物理量計算

ソースコード：　`./src/MC_sampling/SR`

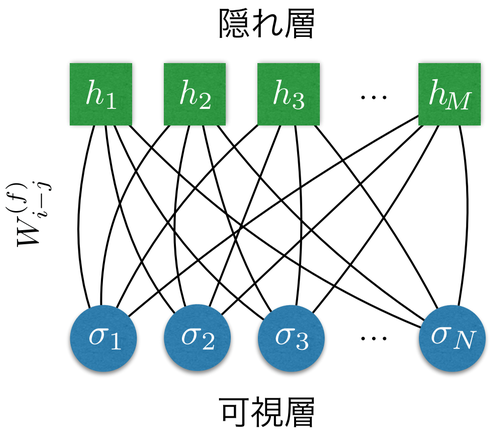
実習場所：　`./examples/1D_AF_chain/MC_sampling/16site`

実行コマンド：　`./RBM_solver_SR.x > log`

必要なファイル：　`RBM.input`（インプット）

ハミルトニアン：　$\mathcal{H} = \sum_i (-\sigma_i^x \sigma_{i+1}^x - \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z)$

変分波動関数：



$$\alpha(\text{hidden variable density}) = M/N$$

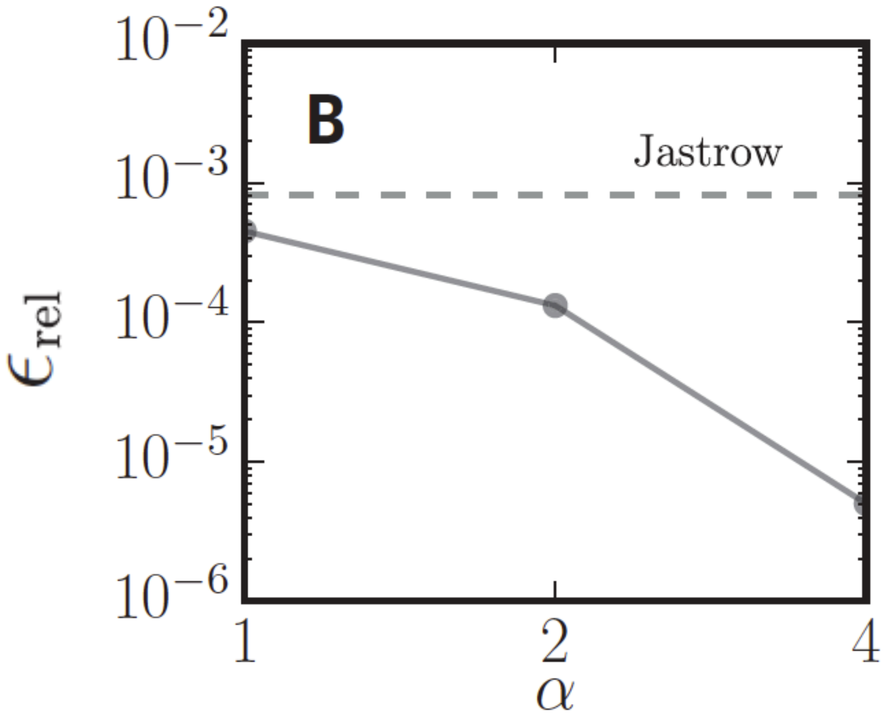
$$\psi_\theta(\sigma) = \prod_{f=1}^{\alpha} \prod_{j=1}^{N} 2\cosh\left(\sum_i W_{i-j}^{(f)} \sigma_i^z\right)$$

Carleo and Troyer Science **355**, 602 (2017)とほぼ同じ設定

課題3-1（発展課題）：カルレオ論文Fig. 3Bと同じ計算をする

80サイトの1次元ハイゼンベルク模型（周期境界条件）に対して、カルレオ・トロイヤーと同じ計算をする

α = M/N (hidden variable density)



$$\epsilon_{\mathrm{rel}} = \frac{E_{\mathrm{RBM}}(\alpha) - E_{\mathrm{exact}}}{|E_{\mathrm{exact}}|}$$

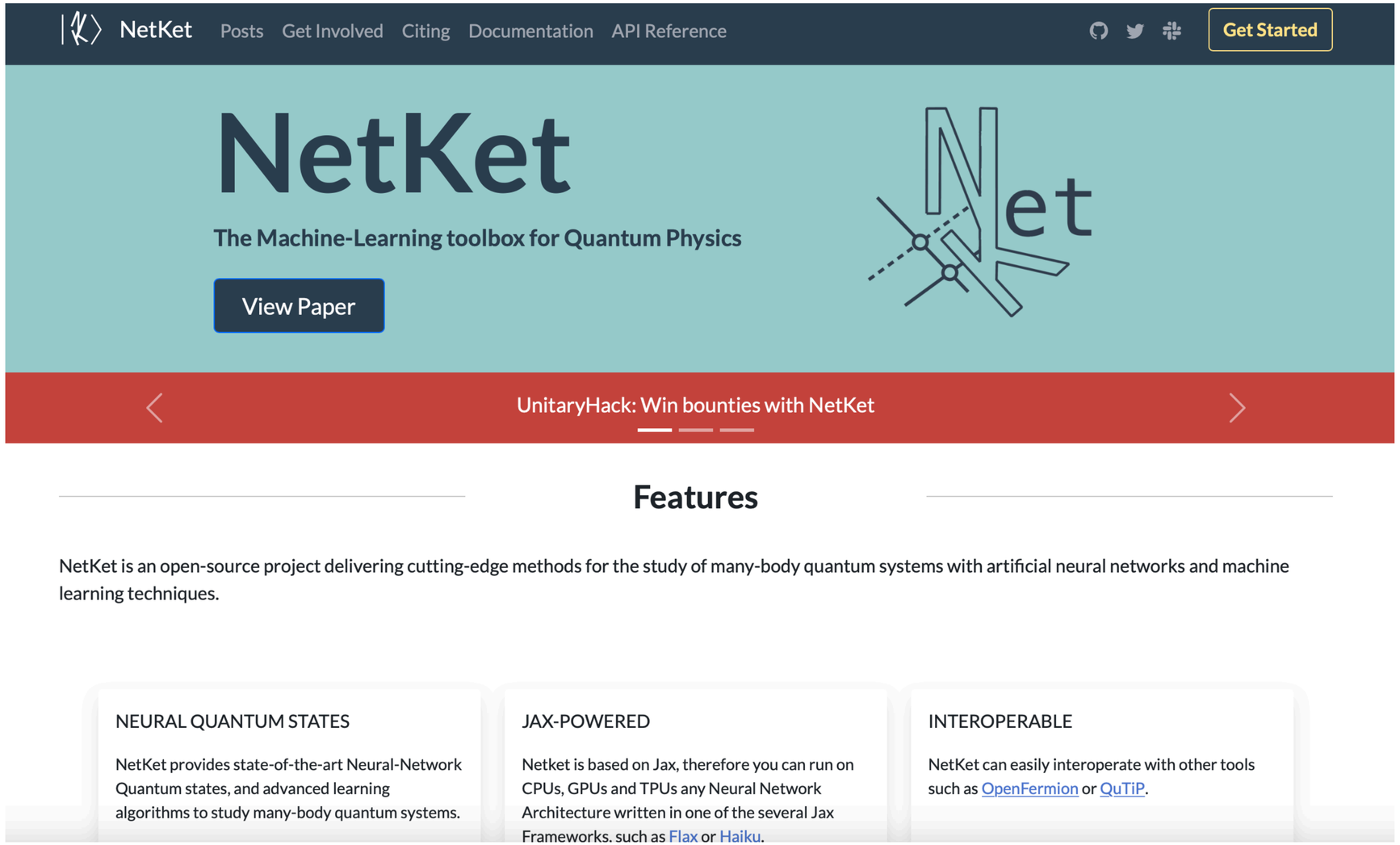
参照となる基底状態エネルギー：

Eexact = −141.84830424　（非常に大きなボンド次元のDMRGで計算）

Carleo and Troyer Science 355, 602 (2017)

※現状のプログラムを使うとすごい時間がかかる。少なくともファーストアップデートは取り入れるべき。できれば並列化も。

参考：NetKet



https://www.netket.org