

ニューラルネットワークによる ボーズハバードモデルのシミュレーション

渡部慎太郎

2019年6月10日

目次

1	概要	2
2	導入	2
2.1	機械学習とは	2
2.2	量子多体系と機械学習	2
3	ニューラルネットワークの仕組み	3
3.1	人工ニューロン単体	3
3.2	全体の構造	3
3.3	重みと閾値の算出	3
4	量子多体系 (bose-hubbard model)	4
4.1	bose-hubbard model	4
4.2	厳密対角化の方法	5
4.3	gutzwiller approximation	5
5	量子多体系への機械学習の応用 (bose-hubbard model の基底状態を求める)	7
5.1	ニューラルネットワークの構造	7
5.2	ハミルトニアン の期待値計算	7
5.3	ハミルトニアン についての逆誤差伝播法	8
5.4	メトロポリスヘイスティング法	10
6	量子多体系への機械学習の応用 2 (bose-hubbard model の励起状態を求める)	11
6.1	基底状態および励起状態以外の状態	11
6.2	第一励起状態であるかどうかの判別	12
6.3	条件付き最適化法	12
7	シミュレーション方法	13
8	結果	14
8.1	基底状態	14
8.2	励起状態	16
9	考察	19
9.1	基底状態	19
9.2	励起状態	19
10	まとめ	20

1 概要

今回、機械学習に興味を持っていた私は、物理学と機械学習の合わさった研究に取り組んだ。取り組む物理系はボーズハバードモデルで、そのモデルに対し、機械学習で使われるアプローチであるニューラルネットの構造を、利用する。まずボーズハバードモデルの基底状態を求める先行研究の再現を行い、次に励起状態を求める方法の構築に挑戦した。

2 導入

目覚ましい発展を遂げている機械学習の技術は、様々な領域への拡張が可能であり、その拡張性ゆえに注目されている。本来近似や厳密計算によって計算する物理系もニューラルネットワークを利用することで新たな物理学の手法の開発の可能性がある。物理学への拡張は主流ではないものの、大きな可能性があり、その研究には新たな物理学の手法の開発という意味で意義がある。今回は機械学習の量子多体系への、特にボーズハバードモデルと呼ばれる、ボーズ粒子が多数トラップされた系への応用を考える

2.1 機械学習とは

機械学習とは、人間に近い学習能力を得ようとする技術である。[1] 機械学習のアルゴリズムには種類があり、

- 教師あり学習
それぞれ対応する答えの決まっている入力データを大量に学習させることで、答えごとの入力データの特徴を分析し分類化して、次の入力に対する答えを出せるようにする。例えばマルバツの絵の入力データをマルバツの答えとともに学習させることで、マルバツの絵の特徴を学習し、次マルバツの絵を見せられたときにマルバツの答えを言えるようになる。
- 教師なし学習
それぞれ答えのない入力データを学習させる。答えのない入力データであってもアルゴリズムを工夫することでその特徴を学習することができる
- 強化学習
入力に対して、AI 側がアクションを起こしその結果を学習していく。結果の評価が良くなるように次の結果を変更していく。

さらに機械学習には技法がいくつかある。機械学習の技法の一つにニューラルネットワークがあり、神経の構造を模することで学習能力をえるものである。神経に対応するものは人工ニューロンであり、要素をつなぐシナプスによって、ネットワークが形成される。学習によってそのネットワークのつながりが変化する。今回はニューラルネットワークを使っていくことになる。

2.2 量子多体系と機械学習

ニューラルネットワークの中に波動関数の情報を格納することで、入力基底、出力として波動関数の係数を得ることが可能である。

3 ニューラルネットワークの仕組み

簡単にニューラルネットワークの仕組みを記述する [2][3]

3.1 人工ニューロン単体

人工ニューロン単体の仕組みは単純で、いくつかある入力の数値にそれぞれ重みをかけ足し合わせ、閾値を引いたものを関数に代入したものを出力する。入力 x_i 出力 y 重み w_i 閾値 θ とすると

$$y = f\left(\sum_{i=1}^n x_i * w_i - \theta\right) \quad (1)$$

関数はシグモイド関数がよく使われる。

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

重みは前のニューロンとのつながりの強さを表し、閾値は値の 0 点の調整の役割をになう
出力された値は次のニューロンの入力となる

3.2 全体の構造

一番簡素なネットワークは三層のニューロン群からなる。まず全体は対象のデータを要素に分けて要素ごとの数値を入力とする。一層目は要素の数と同じだけのニューロンからなり、対応するニューロンの値を入力とし、それをそのまま出力する。二層目は一層目のニューロンの出力を受取り、関数を通して、三層目に出力し、三層目も同様である。三層目の出力が全体の出力となる。

3.3 重みと閾値の算出

重みと閾値に仮の値を代入し、教師となるデータを入れる。出力は意味のないものとなる。教師データには出力の答えがあるので、その値との誤差が出てくる。誤差の算出方法は

$$Q_k = \sum_{i=1}^n (z - t)^2 \quad (3)$$

z は出力 t は答えである。データごとに Q_k を算出し足し合わせる。

$$Q_t = \sum_{k=1}^m Q_k \quad (4)$$

Q_t の値が最小となるように重みと閾値を算出する。誤差逆伝播法を使う。

$$w \leftarrow w - \eta \frac{\partial Q_t(w)}{\partial w} \quad (5)$$

によって、 w を変化させ、もう一度 Q_t および $\frac{\partial Q_t(w)}{\partial w}$ を算出する。そして一度 w を変化させる。これを繰り返すうちに、 Q_t は最小に限りなく近づく。

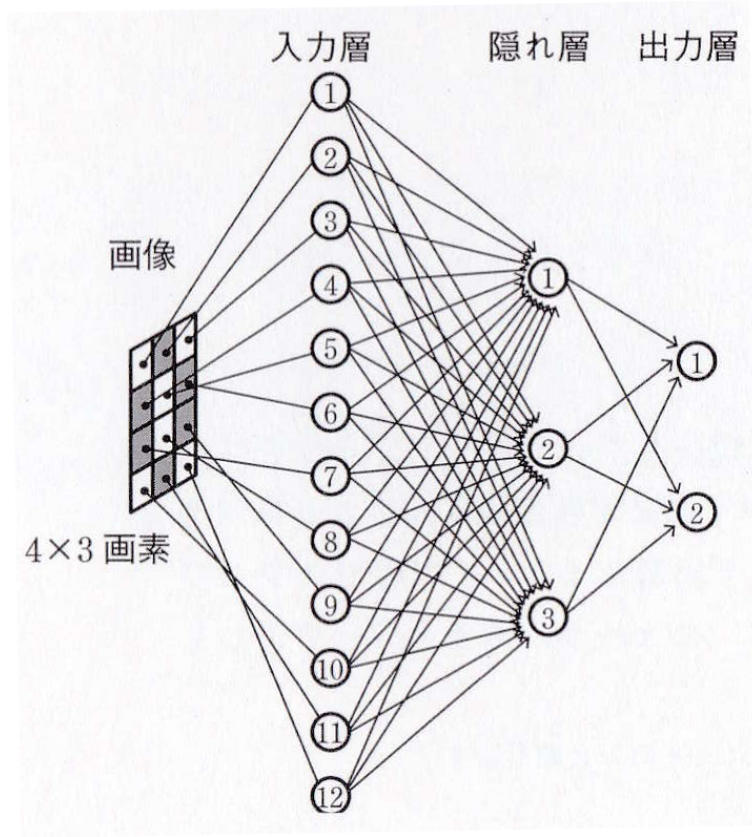


図1 ニューラルネットワーク [2]

4 量子多体系 (bose-hubbard model)

4.1 bose-hubbard model

今回は物理系についてボーズハバードモデルを使う。ボーズハバードモデルについて簡単に記述するボーズハバードモデルは光格子内での粒子のふるまいを説明するモデルである。格子点には格子点ごとに粒子がいくつか束縛されている。その粒子の系を記述するモデルである。ボーズハバードモデルのハミルトニアンは

$$\hat{H} = -J \sum_{i,j} \hat{a}_i \hat{a}_j^\dagger + \sum_i [V_i \hat{n}_i + \frac{U}{2} \hat{n}_i (\hat{n}_i - 1)] \quad (6)$$

であらわされる。一項目が格子点間のトンネリングを表す、二項目が格子点依存のポテンシャルの寄与、三項目が格子点内の粒子同士の相互作用の寄与をあらわす。また、 $\hat{n}_i = \hat{a}_i^\dagger \hat{a}_i$ である。

ここで、格子点 i の粒子数を n_i のように表すとする。またある状態の格子点の粒子数の組み合わせ (n_1, n_2, n_3, \dots) を \mathbf{n} のようにあらわす。 n_i がすべて整数の場合の状態を基底ベクトル \mathbf{n} とすると、 $|\Psi\rangle = \sum C(\mathbf{n}) |\mathbf{n}\rangle$ のようにあらわせる。また、全粒子数 N と、格子点数 M とすると、 \mathbf{n} の場合の数は $\frac{(N+M-1)!}{N!(M-1)!}$ となる。

4.2 厳密対角化の方法

$$|\Psi\rangle = \sum_n C(\mathbf{n}) |\mathbf{n}\rangle \quad (7)$$

および

$$\hat{H} |\Psi\rangle = E |\Psi\rangle \quad (8)$$

より、

$$\sum_n C(\mathbf{n}) \hat{H} |\mathbf{n}\rangle = \sum_n C(\mathbf{n}) E |\mathbf{n}\rangle \quad (9)$$

これに $\langle \mathbf{m} |$ を作用させると、

$$\sum_n C(\mathbf{n}) \langle \mathbf{m} | \hat{H} |\mathbf{n}\rangle = \sum_n C(\mathbf{n}) E \langle \mathbf{m} | \mathbf{n}\rangle \quad (10)$$

より行列とベクトルで表示すると

$$\begin{pmatrix} \langle \mathbf{m}_1 | \hat{H} | \mathbf{n}_1 \rangle & \langle \mathbf{m}_1 | \hat{H} | \mathbf{n}_2 \rangle & \langle \mathbf{m}_1 | \hat{H} | \mathbf{n}_3 \rangle & \dots & \dots \end{pmatrix} \begin{pmatrix} C(\mathbf{n}_1) \\ C(\mathbf{n}_2) \\ C(\mathbf{n}_3) \\ \vdots \\ \vdots \end{pmatrix} = E C(\mathbf{m}) \quad (11)$$

さらに

$$\begin{pmatrix} \langle \mathbf{m}_1 | \hat{H} | \mathbf{n}_1 \rangle & \langle \mathbf{m}_1 | \hat{H} | \mathbf{n}_2 \rangle & \langle \mathbf{m}_1 | \hat{H} | \mathbf{n}_3 \rangle & \dots & \dots \\ \langle \mathbf{m}_2 | \hat{H} | \mathbf{n}_1 \rangle & \langle \mathbf{m}_2 | \hat{H} | \mathbf{n}_2 \rangle & \langle \mathbf{m}_2 | \hat{H} | \mathbf{n}_3 \rangle & \dots & \dots \\ \langle \mathbf{m}_3 | \hat{H} | \mathbf{n}_1 \rangle & \langle \mathbf{m}_3 | \hat{H} | \mathbf{n}_2 \rangle & \langle \mathbf{m}_3 | \hat{H} | \mathbf{n}_3 \rangle & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} C(\mathbf{n}_1) \\ C(\mathbf{n}_2) \\ C(\mathbf{n}_3) \\ \vdots \\ \vdots \end{pmatrix} = E \begin{pmatrix} C(\mathbf{m}_1) \\ C(\mathbf{m}_2) \\ C(\mathbf{m}_3) \\ \vdots \\ \vdots \end{pmatrix} \quad (12)$$

と変化できる。この式を解くことで、 $C(\mathbf{n})$ および E を算出できる。いくつか出てきた E のうち、最小のものが基底状態、そのほかのものが励起状態のものである。この式を解くときは intel LAPACK を利用する。

4.3 gutzwiller approximation

グッツヴィラー近似とは $U = 0$ の極限で近似するものである。状態を

$$|\Psi\rangle = \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) \quad (13)$$

と表す。ここで $\langle \Psi | \hat{H} | \Psi \rangle$ は

$$\langle \Psi | \hat{H} | \Psi \rangle = \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} \langle n |_i \right) \hat{H} \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) \quad (14)$$

と表せる。

$$\hat{H} = -J \sum_{i,j} \hat{a}_i \hat{a}_j^\dagger + \sum_i \left[V_i \hat{n}_i + \frac{U}{2} \hat{n}_i (\hat{n}_i - 1) \right] \quad (15)$$

より、

$$\langle \Psi | \hat{H} | \Psi \rangle = \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} \langle n |_i \right) \left(-J \sum_{i,j} \hat{a}_i \hat{a}_j^\dagger + \sum_i [V_i \hat{n}_i + \frac{U}{2} \hat{n}_i (\hat{n}_i - 1)] \right) \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) \quad (16)$$

ここで

$$\begin{aligned} \hat{n}_i \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) &= \hat{n}_i (f_0^{(1)} |0\rangle_1 + f_1^{(1)} |1\rangle_1 + \dots) (f_0^{(2)} |0\rangle_2 + f_1^{(2)} |1\rangle_2 + \dots) \dots (f_0^{(i)} |0\rangle_i + f_1^{(i)} |1\rangle_i + \dots) \dots \\ &= (f_0^{(1)} |0\rangle_1 + f_1^{(1)} |1\rangle_1 + \dots) (f_0^{(2)} |0\rangle_2 + f_1^{(2)} |1\rangle_2 + \dots) \dots (1 * f_0^{(i)} |0\rangle_i + 2 * f_1^{(i)} |1\rangle_i + \dots) \dots \end{aligned} \quad (17)$$

となるので

$$\prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} \langle n |_i \right) \hat{n}_i \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) = \sum_{n=0}^{\infty} n |f_n^{(i)}|^2 \quad (18)$$

$$\prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} \langle n |_i \right) \sum_i [V_i \hat{n}_i + \frac{U}{2} \hat{n}_i (\hat{n}_i - 1)] \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) = \sum_i [V_i \sum_{n=0}^{\infty} n |f_n^{(i)}|^2 + \frac{U}{2} \sum_{n=0}^{\infty} n(n-1) |f_n^{(i)}|^2] \quad (19)$$

$$\begin{aligned} \hat{a}_i \hat{a}_j^\dagger \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) &= \hat{a}_i \hat{a}_j^\dagger (f_0^{(1)} |0\rangle_1 + f_1^{(1)} |1\rangle_1 + \dots) (f_0^{(2)} |0\rangle_2 + f_1^{(2)} |1\rangle_2 + \dots) \dots \\ &\dots (f_0^{(i)} |0\rangle_i + f_1^{(i)} |1\rangle_i + \dots) (f_0^{(j)} |0\rangle_j + f_1^{(j)} |1\rangle_j + \dots) \dots \\ &= (f_0^{(1)} |0\rangle_1 + f_1^{(1)} |1\rangle_1 + \dots) (f_0^{(2)} |0\rangle_2 + f_1^{(2)} |1\rangle_2 + \dots) \dots \\ &\dots (\sqrt{1} f_1^{(i)} |0\rangle_i + \sqrt{2} f_2^{(i)} |1\rangle_i + \dots) (\sqrt{1} f_0^{(j)} |1\rangle_j + \sqrt{2} f_1^{(j)} |2\rangle_j + \dots) \dots \end{aligned} \quad (20)$$

$$\prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} \langle n |_i \right) \hat{a}_i \hat{a}_j^\dagger \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) = \left(\sum_{n=0}^{\infty} \sqrt{n+1} f_n^{(i)} f_{n+1}^{(i)} \right) \left(\sum_{n=1}^{\infty} \sqrt{n} f_n^{(j)} f_{n-1}^{(j)} \right) \quad (21)$$

$$\begin{aligned} &\prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} \langle n |_i \right) \left\{ -J \sum_{i,j} \hat{a}_i \hat{a}_j^\dagger + \sum_i [V_i \hat{n}_i + \frac{U}{2} \hat{n}_i (\hat{n}_i - 1)] \right\} \prod_i \left(\sum_{n=0}^{\infty} f_n^{(i)} |n\rangle_i \right) \\ &= \sum_{i,j} \left\{ \left(\sum_{n=0}^{\infty} \sqrt{n+1} f_n^{(i)} f_{n+1}^{(i)} \right) \left(\sum_{n=1}^{\infty} \sqrt{n} f_n^{(j)} f_{n-1}^{(j)} \right) \right\} + \sum_i \left\{ V_i \sum_{n=0}^{\infty} n |f_n^{(i)}|^2 + \frac{U}{2} \sum_{n=0}^{\infty} n(n-1) |f_n^{(i)}|^2 \right\} \end{aligned} \quad (22)$$

となる。

$$\frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} - \mu \sum \frac{\langle \Psi | \hat{n} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \quad (23)$$

を最小化することで基底状態を得る。 μ は値が決まっていない。 μ を変化させていって、粒子数 N が $\sum \frac{\langle \Psi | \hat{n} | \Psi \rangle}{\langle \Psi | \Psi \rangle} = N$ となる μ を探す。

5 量子多体系への機械学習の応用 (bose-hubbard model の基底状態を求める)

ボーズハーバードモデルの問題をニューラルネットワークで解いていく。[4]

5.1 ニューラルネットワークの構造

状態 $|\Psi\rangle = \sum C(\mathbf{n}) |\mathbf{n}\rangle$ において、 $|\mathbf{n}\rangle$ を入力すると、 $C(\mathbf{n})$ を出力するニューラルネットワークを考える。出力された値から、ハミルトニアン の期待値を算出し、これが最小値となるようにパラメーターを決定した時の、 $\sum C(\mathbf{n}) |\mathbf{n}\rangle$ が基底状態である。

次にネットワークの構造について記述する。一層目のニューロンの式は

$$u_k^{(1)}(\mathbf{n}) = \tanh\left(\sum_{j=1}^M W_{kj}^{(1)} n_j + h_k^{(1)}\right) \quad (24)$$

とする。ここではシグモイド関数ではなく、ハイパボリックタンジェントを利用している。二層目の式は

$$u_m^{(2)}(\mathbf{n}) = \sum_{k=1}^{N_H} W_{mk}^{(2)} u_k^{(1)}(\mathbf{n}) + h_m^{(2)} \quad (25)$$

とする。W および h は重みであり、実数。 N_H は二層目のニューロンの数である。そして最終的な出力は

$$C(\mathbf{n}) = \exp[u_1^{(2)}(\mathbf{n}) + iu_2^{(2)}(\mathbf{n})] \quad (26)$$

5.2 ハミルトニアン の期待値計算

その次はハミルトニアン の期待値の算出である。

$$\langle A \rangle = \frac{\langle \Psi | A | \Psi \rangle}{\langle \Psi | \Psi \rangle} \quad (27)$$

これに $|\Psi\rangle = \sum C(\mathbf{n}) |\mathbf{n}\rangle$ を代入すると

$$\langle A \rangle = \frac{\sum_{\mathbf{n}, \mathbf{n}'} C^*(\mathbf{n}) \langle \mathbf{n} | A | \mathbf{n}' \rangle C(\mathbf{n}')}{\sum_{\mathbf{n}} |C(\mathbf{n})|^2} \quad (28)$$

となる。これによって期待値を算出する。さらに、

$$\langle A \rangle = \sum_{\mathbf{n}, \mathbf{n}'} \frac{|C(\mathbf{n})|^2}{\sum_{\mathbf{n}''} |C(\mathbf{n}'')|^2} \langle \mathbf{n} | A | \mathbf{n}' \rangle \frac{C(\mathbf{n}')}{C(\mathbf{n})} \quad (29)$$

と変形できる。ここで、 $\frac{|C(\mathbf{n})|^2}{\sum_{\mathbf{n}''} |C(\mathbf{n}'')|^2} = P(\mathbf{n})$ とおくと

$$\langle A \rangle = \sum_{\mathbf{n}, \mathbf{n}'} P(\mathbf{n}) \langle \mathbf{n} | A | \mathbf{n}' \rangle \frac{C(\mathbf{n}')}{C(\mathbf{n})} \simeq \left\langle \sum_{\mathbf{n}'} \langle \mathbf{n} | A | \mathbf{n}' \rangle \frac{C(\mathbf{n}')}{C(\mathbf{n})} \right\rangle_M = \langle A \rangle_M \quad (30)$$

とあらわせる。ハミルトニアンの期待値は

$$\langle \hat{H} \rangle \simeq \left\langle \sum_{n'} \langle n | \hat{H} | n' \rangle \frac{C(\mathbf{n}')}{C(\mathbf{n})} \right\rangle_M = \langle \hat{H} \rangle_M \quad (31)$$

となる。ここで、 \mathbf{n} について、すべてサンプルするのではなく、メトロポリスヘイスティング法をつかうことによって、計算量を減らしている。

5.3 ハミルトニアンの逆誤差伝播法

このハミルトニアンの期待値の最小値はハミルトニアンの期待値をパラメーターで微分することによって求める。

$$w_i \rightarrow w_i - \alpha \frac{\partial \langle \hat{H} \rangle}{\partial w_i} \quad (32)$$

ベクトルにすると

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \end{pmatrix} \rightarrow \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \end{pmatrix} - \alpha \begin{pmatrix} \frac{\partial \langle \hat{H} \rangle}{\partial w_1} \\ \frac{\partial \langle \hat{H} \rangle}{\partial w_2} \\ \frac{\partial \langle \hat{H} \rangle}{\partial w_3} \\ \vdots \end{pmatrix} \quad (33)$$

となる。これを計算していくことで、ハミルトニアン \hat{H} の期待値は限りなく最小に近づく。

$$\begin{aligned}
\frac{\partial \langle \hat{H} \rangle}{\partial w} &= \frac{\partial}{\partial w} \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}')}{\sum_n |C(\mathbf{n})|^2} \\
&= \frac{(\sum_n |C(\mathbf{n})|^2) \frac{\partial (\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}'))}{\partial w}}{(\sum_n |C(\mathbf{n})|^2)^2} - \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}') \frac{\partial (\sum_n |C(\mathbf{n})|^2)}{\partial w}}{(\sum_n |C(\mathbf{n})|^2)^2} \\
&= \frac{1}{\sum_n |C(\mathbf{n})|^2} \frac{\partial (\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}'))}{\partial w} - \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}') \frac{\partial (\sum_n |C(\mathbf{n})|^2)}{\partial w}}{(\sum_n |C(\mathbf{n})|^2)^2} \\
&= \frac{1}{\sum_n |C(\mathbf{n})|^2} \left(\sum_{n,n'} \frac{\partial C^*(\mathbf{n})}{\partial w} \langle n|A|n' \rangle C(\mathbf{n}') + \frac{1}{\sum_n |C(\mathbf{n})|^2} \left(\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle \frac{\partial C(\mathbf{n}')}{\partial w} \right) \right. \\
&\quad \left. - \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}')}{(\sum_n |C(\mathbf{n})|^2)^2} \left(\sum_n \frac{\partial C^*(\mathbf{n})}{\partial w} C(\mathbf{n}) \right) - \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}')}{(\sum_n |C(\mathbf{n})|^2)^2} \left(\sum_n C^*(\mathbf{n}) \frac{\partial C(\mathbf{n})}{\partial w} \right) \right) \\
&= \frac{1}{\sum_n |C(\mathbf{n})|^2} \left(\sum_{n,n'} \frac{\partial C^*(\mathbf{n})}{\partial w} \frac{1}{C^*(\mathbf{n})} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}') \right) + \frac{1}{\sum_n |C(\mathbf{n})|^2} \left(\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle \frac{\partial C(\mathbf{n}')}{\partial w} \frac{1}{C(\mathbf{n}')} C(\mathbf{n}') \right) \\
&\quad - \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}')}{(\sum_n |C(\mathbf{n})|^2)^2} \left(\sum_n \frac{\partial C^*(\mathbf{n})}{\partial w} \frac{1}{C^*(\mathbf{n})} |C(\mathbf{n})|^2 \right) \\
&\quad - \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}')}{(\sum_n |C(\mathbf{n})|^2)^2} \left(\sum_n |C(\mathbf{n})|^2 \frac{\partial C(\mathbf{n})}{\partial w} \frac{1}{C(\mathbf{n})} \right) \\
&= \left(\sum_{n,n'} \frac{|C(\mathbf{n})|^2}{\sum_{n'''} |C(\mathbf{n}''')|^2} \frac{\partial C^*(\mathbf{n})}{\partial w} \frac{1}{C^*(\mathbf{n})} \langle n|A|n' \rangle \frac{C(\mathbf{n})}{C(\mathbf{n}')} \right) + \left(\sum_{n,n'} \frac{C(\mathbf{n})^*}{C(\mathbf{n}')^*} \langle n|A|n' \rangle \frac{\partial C(\mathbf{n}')}{\partial w} \frac{1}{C(\mathbf{n}')} \frac{|C(\mathbf{n}')|^2}{\sum_{n'''} |C(\mathbf{n}''')|^2} \right) \\
&\quad - \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}')}{\sum_n |C(\mathbf{n})|^2} \left(\sum_n \frac{\partial C^*(\mathbf{n})}{\partial w} \frac{1}{C^*(\mathbf{n})} \frac{|C(\mathbf{n})|^2}{\sum_{n'''} |C(\mathbf{n}''')|^2} \right) \\
&\quad - \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n|A|n' \rangle C(\mathbf{n}')}{\sum_n |C(\mathbf{n})|^2} \left(\sum_n \frac{|C(\mathbf{n})|^2}{\sum_{n'''} |C(\mathbf{n}''')|^2} \frac{\partial C(\mathbf{n})}{\partial w} \frac{1}{C(\mathbf{n})} \right) \\
&= \sum_n \frac{|C(\mathbf{n})|^2}{\sum_{n'''} |C(\mathbf{n}''')|^2} \frac{\partial C(\mathbf{n})^*}{\partial w} \frac{1}{C(\mathbf{n})^*} \langle n|A|n' \rangle \frac{C(\mathbf{n}')}{C(\mathbf{n})} + \left\{ \sum_n \frac{|C(\mathbf{n})|^2}{\sum_{n'''} |C(\mathbf{n}''')|^2} \frac{\partial C(\mathbf{n})^*}{\partial w} \frac{1}{C(\mathbf{n})^*} \langle n|A|n' \rangle \frac{C(\mathbf{n}')}{C(\mathbf{n})} \right\}^* \\
&\quad - \langle \hat{H} \rangle \sum_n \frac{|C(\mathbf{n})|^2}{\sum_{n'''} |C(\mathbf{n}''')|^2} \frac{\partial C(\mathbf{n})}{\partial w} \frac{1}{C(\mathbf{n})} - \langle \hat{H} \rangle \sum_n \frac{|C(\mathbf{n})|^2}{\sum_{n'''} |C(\mathbf{n}''')|^2} \frac{\partial C(\mathbf{n})^*}{\partial w} \frac{1}{C(\mathbf{n})^*}
\end{aligned} \tag{34}$$

ここで $\frac{\partial C \mathbf{n}}{\partial w} \frac{1}{C(\mathbf{n})} = O_w(\mathbf{n})$ とおくと、

$$\begin{aligned}
\frac{\partial \langle \hat{H} \rangle}{\partial w} &\simeq \langle \hat{H} O_w^* \rangle_M + \left\{ \langle \hat{H} O_w^* \rangle_M \right\}^* - \langle \hat{H} \rangle_M \langle O_w \rangle_M - \langle \hat{H} \rangle_M \langle O_w^* \rangle_M \\
&= 2 \text{Re}[\langle \hat{H} O_w^* \rangle_M - \langle \hat{H} \rangle_M \langle O_w^* \rangle_M]
\end{aligned} \tag{35}$$

$O_w(\mathbf{n})$ について、

$$\frac{\partial C \mathbf{n}}{\partial w} \frac{1}{C(\mathbf{n})} = O_w(\mathbf{n}) \tag{36}$$

$$C(\mathbf{n}) = \exp[u_1^{(2)}(\mathbf{n}) + i u_2^{(2)}(\mathbf{n})] \tag{37}$$

$$u_m^{(2)}(\mathbf{n}) = \sum_{k=1}^{N_H} W_{mk}^{(2)} u_k^{(1)}(\mathbf{n}) + h_m^{(2)} \quad (38)$$

$$u_k^{(1)}(\mathbf{n}) = \tanh\left(\sum_{j=1}^M W_{kj}^{(1)} n_j + h_k^{(1)}\right) \quad (39)$$

より、

$$C(\mathbf{n}) = \exp\left[\left\{\sum_{k=1}^{N_H} W_{1k}^{(2)} [\tanh(\sum_{j=1}^M W_{kj}^{(1)} n_j + h_k^{(1)})] + h_1^{(2)}\right\} + i \left\{\sum_{k=1}^{N_H} W_{2k}^{(2)} [\tanh(\sum_{j=1}^M W_{kj}^{(1)} n_j + h_k^{(1)})] + h_2^{(2)}\right\}\right] \quad (40)$$

これの微分は、 $\left\{\sum_{k=1}^{N_H} W_{1k}^{(2)} [\tanh(\sum_{j=1}^M W_{kj}^{(1)} n_j + h_k^{(1)})] + h_1^{(2)}\right\} + i \left\{\sum_{k=1}^{N_H} W_{2k}^{(2)} [\tanh(\sum_{j=1}^M W_{kj}^{(1)} n_j + h_k^{(1)})] + h_2^{(2)}\right\} = a$ とおくと、

$$\frac{\partial C(\mathbf{n})}{\partial w} = \frac{\partial a}{\partial w} \exp[a] \quad (41)$$

$$O_w(\mathbf{n}) = \frac{\partial C(\mathbf{n})}{\partial w} \frac{1}{C(\mathbf{n})} = \frac{\partial a}{\partial w} \quad (42)$$

$\frac{\partial a}{\partial w}$ について、 w ごとにその中身は変わる。

$$\frac{\partial a}{\partial W_{1m}^{(2)}} = [\tanh(\sum_{j=1}^M W_{mj}^{(1)} n_j + h_m^{(1)})] \quad (43)$$

$$\frac{\partial a}{\partial W_{2m}^{(2)}} = i [\tanh(\sum_{j=1}^M W_{mj}^{(1)} n_j + h_m^{(1)})] \quad (44)$$

$$\frac{\partial a}{\partial h_1^{(2)}} = 1 \quad (45)$$

$$\frac{\partial a}{\partial h_2^{(2)}} = i \quad (46)$$

$$\frac{\partial a}{\partial W_{ml}^{(1)}} = \left\{ W_{1m}^{(2)} \frac{n_l}{\cosh^2(\sum_{j=1}^M W_{mj}^{(1)} n_j + h_m^{(1)})} \right\} + i \left\{ W_{2m}^{(2)} \frac{n_l}{\cosh^2(\sum_{j=1}^M W_{mj}^{(1)} n_j + h_m^{(1)})} \right\} \quad (47)$$

$$\frac{\partial a}{\partial h_m^{(1)}} = + \left\{ W_{1m}^{(2)} \frac{1}{\cosh^2(\sum_{j=1}^M W_{mj}^{(1)} n_j + h_m^{(1)})} \right\} + i \left\{ W_{2m}^{(2)} \frac{1}{\cosh^2(\sum_{j=1}^M W_{mj}^{(1)} n_j + h_m^{(1)})} \right\} \quad (48)$$

これによって $O_w(\mathbf{n})$ が計算できるようになった。この期待値メトロポリスヘイスティング法でを計算し、(35) に代入することにより (32) を実行することができるようになる。

5.4 メトロポリスヘイスティング法

次に期待値計算に使うメトロポリスヘイスティング法について記述するメトロポリスヘイスティング法とはモンテカルロ法の一つで、「直接サンプリングするのが難しい確率分布から統計標本の配列を生成するのに用いられるマルコフ連鎖を構築するのに用いられる手法」 [5] である。

この方法ではまず提案分布と呼ばれる別の確率分布に従う候補サンプルを生成し、それを目標分布の関数に

従って採択するかを決定する。

まず提案分布 $Q(x|y)$ を用意する。関数は x, y に依存する関数である。初期値 x_0 を決める。目標関数を $P(x)$ とする。

t 番目のサンプリングにおいて、 $Q(x|x_{t-1})$ から確率分布に従って x^* を生成し、採択率 $\alpha = \frac{Q(x_{t-1}|x^*)P(x^*)}{Q(x^*|x_{t-1})P(x_{t-1})}$ を計算し、確率 α で候補を採択し、 $x_t = x^*$ とする。候補が棄却された場合 $x_t = x_{t-1}$ とする。これを繰り返す。

例として提案分布は y を中心とするガウス分布 $Q(x|y)$ があげられる。初期値 x_0 を決める。t 番目のサンプリングにおいて、 x^* を x_{t-1} を中心とするガウス分布で生成し、採択率 $\alpha = \frac{Q(x_{t-1}|x^*)P(x^*)}{Q(x^*|x_{t-1})P(x_{t-1})}$ を計算し、0 から 1 ランダムな値を生成し、 α より小さかったら採択し、大きかったら棄却する。これを繰り返す。得られた x_i のうち最初のいくつかは棄却することで初期値 x_0 の選び方の依存性をへらすことができる。

初期値 $|\mathbf{n}_0\rangle$ を決める。t 番目のサンプリングにおいて、 $|\mathbf{n}_{t-1}\rangle$ の粒子数を、変化させ、 $|\mathbf{n}_a\rangle$ をつくる。採択率 $\alpha = \frac{Q(|\mathbf{n}_{t-1}\rangle||\mathbf{n}_a\rangle)|C(\mathbf{n}_a)|^2}{Q(|\mathbf{n}_a\rangle||\mathbf{n}_{t-1}\rangle)|C(\mathbf{n}_{t-1})|^2}$ を計算し、0 から 1 ランダムな値を生成し、 α より小さかったら採択し、大きかったら棄却する。候補が採択されたら、 $|\mathbf{n}_t\rangle = |\mathbf{n}_a\rangle$ とし、棄却した場合 $|\mathbf{n}_t\rangle = |\mathbf{n}_{t-1}\rangle$ とする。これを繰り返す。何回かに一回サンプルとして利用する。得られた $|\mathbf{n}_i\rangle$ のうち最初のいくつかは棄却することで初期値 $|\mathbf{n}_0\rangle$ の選び方の依存性をへらすことができる。 $|\mathbf{n}\rangle$ の変化のさせ方の例として、 $|\dots, n_i, \dots, n_j, \dots\rangle$ を $|\dots, n_i + 1, \dots, n_j - 1, \dots\rangle$ と変化させる方法がある。この時の $\frac{Q(|\mathbf{n}_{t-1}\rangle||\mathbf{n}_a\rangle)}{Q(|\mathbf{n}_a\rangle||\mathbf{n}_{t-1}\rangle)}$ は \mathbf{n}_a に含まれる 0 の数を q_a 、 \mathbf{n}_{t-1} に含まれる 0 の数を q_{t-1} とすると、 $\frac{Q(|\mathbf{n}_{t-1}\rangle||\mathbf{n}_a\rangle)}{Q(|\mathbf{n}_a\rangle||\mathbf{n}_{t-1}\rangle)} = \frac{q_{t-1}}{q_a}$ となる。

$$\langle A \rangle = \sum_{\mathbf{n}} P(\mathbf{n}) F(\mathbf{n}) \simeq \frac{1}{N_{\text{sample}}} \sum_{\mathbf{n}=\text{sample}} F(\mathbf{n}) = \langle A \rangle_M \quad (49)$$

$F(\mathbf{n})$ をメトロポリスヘイスティングでサンプリングした \mathbf{n} ごとにたしていく。

6 量子多体系への機械学習の応用 2 (bose-hubbard model の励起状態を求める)

6.1 基底状態および励起状態以外の状態

$$\hat{H} |\psi_n\rangle = E |\psi_n\rangle \quad (50)$$

およびより、

$$|\psi(t)\rangle = \sum D_n(t) |\psi_n\rangle \quad (51)$$

と表すことができる。ボーズハバードモデルでは、 $|\Psi\rangle = \sum_{\mathbf{n}} C(\mathbf{n}) |\mathbf{n}\rangle$ であった、この表記を少し変えて $|\psi_m\rangle = \sum_{\mathbf{n}} C_{mn} |\mathbf{n}\rangle$ と表すと、

$$D_m(t) |\psi_m\rangle = D_m(t) \sum_{\mathbf{n}} C_{mn} |\mathbf{n}\rangle \quad (52)$$

$$|\psi(t)\rangle = \sum_{\mathbf{m}} D_{\mathbf{m}}(t) |\psi_{\mathbf{m}}\rangle = \sum_{\mathbf{m}} D_{\mathbf{m}}(t) \sum_{\mathbf{n}} C_{\mathbf{m}\mathbf{n}} |\mathbf{n}\rangle = \sum_{\mathbf{n}} \left(\sum_{\mathbf{m}} D_{\mathbf{m}}(t) C_{\mathbf{m}\mathbf{n}} \right) |\mathbf{n}\rangle = \sum_{\mathbf{n}} C'_{\mathbf{n}}(t) |\mathbf{n}\rangle \quad (53)$$

と表すことができる。一方、

$$\langle \psi(t) | \hat{H} | \psi(t) \rangle = \sum D_{\mathbf{m}}(t)^* \langle \psi_{\mathbf{m}} | \hat{H} \sum D_{\mathbf{n}}(t) |\psi_{\mathbf{n}}\rangle \quad (54)$$

$$\langle \psi(t) | \hat{H} | \psi(t) \rangle = \sum |D_n(t)|^2 E_n \quad (55)$$

となっている。

6.2 第一励起状態であるかどうかの判別

$E_0 < \langle \psi(t) | \hat{H} | \psi(t) \rangle < E_1$ の時、 $|D_0(t)|^2 = 0$ であると、 $\langle \psi(t) | \hat{H} | \psi(t) \rangle < \sum_{n=0}^{\infty} |D_n(t)|^2 E_n$ となってしまうことから、

$$|D_0(t)|^2 > 0 \quad (56)$$

$$\langle \psi(t) | \psi_0 \rangle = \sum_{m=0}^{\infty} D_m^*(t) \langle \psi_m | \psi_0 \rangle \quad (57)$$

$$\sum_n C'_n(t)^* \langle n | \sum_n C_{0n} | n \rangle = \sum_{m=0}^{\infty} D_m^*(t) \langle \psi_m | \psi_0 \rangle \quad (58)$$

$$\sum_n \sum_{n'} C'_n(t)^* C_{0n'} \langle n | n' \rangle = D_0^*(t) \quad (59)$$

$$\sum_n C'_n(t)^* C_{0n} = D_0^*(t) \quad (60)$$

つまり、 $E_0 < \langle \psi(t) | \hat{H} | \psi(t) \rangle < E_1$ のとき、 $\sum_n C'_n(t) C_{0n} \neq 0$ であり、 $|\psi(t)\rangle$ が励起状態の時、 $\sum_n C'_n(t) C_{0n} = 0$ となる。

つまり、 $\sum_n C'_n(t) C_{0n} = 0$ を制約条件として、エネルギーの最小値を求めればよいことがわかる。これは第二励起状態以上へも拡張できる。

6.3 条件付き最適化法

制約条件付きの最適化問題をシミュレーションするにはいくつかの方法がある。今回はペナルティ関数法を使う。外点法とも呼ばれる。ペナルティ関数法とは、制約条件のある最適化問題を、制約条件のない最適化問題に変換する方法の一つである。[6] $g(\mathbf{x}) = 0$ の制限下で $f(\mathbf{x})$ の最小を求めるとき、

$$G(\mathbf{x}) = |g|^2 \quad (61)$$

とおき、変換後の式を

$$F(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda G(\mathbf{x}) \quad (62)$$

とする。この最小値を求めることで、制限付き最小値問題を解くことができる。 λ は適当な値から徐々に増加させていく。今回、条件は

$$g = \sum_n C'_n(t) C_{0n} = \frac{\sum_n C'^*(n) C_0(n)}{\sqrt{\sum_n |C'(n)|^2} \sqrt{\sum_n |C_0(n)|^2}} \quad (63)$$

であり、

$$G = |g|^2 = \frac{(\sum_n C'^*(n) C_0(n)) (\sum_n C'(n) C_0^*(n))}{(\sum_n |C'(n)|^2) (\sum_n |C_0(n)|^2)} \quad (64)$$

となる。ここから再急降下法を使用する。

$$w_i \rightarrow w_i - \alpha \frac{\partial F(\mathbf{x}, \lambda)}{\partial w_i} \quad (65)$$

ここで、

$$\frac{\partial F(\mathbf{x}, \lambda)}{\partial w_i} = \frac{\partial f(\mathbf{x})}{\partial w_i} + \lambda \frac{\partial G(\mathbf{x})}{\partial w_i} \quad (66)$$

である。

$$\begin{aligned} \frac{\partial G}{\partial w} &= \frac{\partial}{\partial w} \frac{(\sum_n C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n))}{(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)} \\ &= \frac{1}{(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)} \left\{ \frac{\partial(\sum_n C'^*(n)C_0(n))}{\partial w} (\sum_n C'(n)C_0^*(n)) + (\sum_n C'^*(n)C_0(n)) \frac{\partial(\sum_n C'(n)C_0^*(n))}{\partial w} \right\} \\ &\quad - \frac{1}{(\sum_n |C'(n)|^2)^2(\sum_n |C_0(n)|^2)^2} \frac{\partial(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)}{\partial w} (\sum_n C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n)) \\ &= \frac{1}{(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)} \left\{ (\sum_n \frac{\partial C'^*(n)}{\partial w} C_0(n))(\sum_n C'(n)C_0^*(n)) + (\sum_n C'^*(n)C_0(n))(\sum_n \frac{\partial C'(n)}{\partial w} C_0^*(n)) \right\} \\ &\quad - \frac{1}{(\sum_n |C'(n)|^2)^2(\sum_n |C_0(n)|^2)^2} \frac{\partial(\sum_n |C'(n)|^2)}{\partial w} (\sum_n C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n)) \\ &= \frac{1}{(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)} \left\{ (\sum_n \frac{\partial C'^*(n)}{\partial w} \frac{1}{C'^*(n)} C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n)) + (\sum_n C'^*(n)C_0(n))(\sum_n \frac{\partial C'(n)}{\partial w} \frac{1}{C'(n)} C'(n)C_0^*(n)) \right\} \\ &\quad - \frac{1}{(\sum_n |C'(n)|^2)^2(\sum_n |C_0(n)|^2)^2} \frac{\partial(\sum_n C'^*(n)C'(n))}{\partial w} (\sum_n C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n)) \\ &= \frac{1}{(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)} \left\{ (\sum_n Ow'^*(n)C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n)) + (\sum_n C'^*(n)C_0(n))(\sum_n Ow'(n)C'(n)C_0^*(n)) \right. \\ &\quad \left. - \frac{(\sum_n C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n))}{(\sum_n |C'(n)|^2)^2(\sum_n |C_0(n)|^2)^2} \left\{ (\sum_n \frac{\partial C'^*(n)}{\partial w} C'(n)) + (\sum_n C'^*(n) \frac{\partial C'(n)}{\partial w}) \right\} \right\} \\ &= \frac{2\text{Re}[(\sum_n Ow'^*(n)C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n))]}{(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)} \\ &\quad - \frac{(\sum_n C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n))}{(\sum_n |C'(n)|^2)^2(\sum_n |C_0(n)|^2)^2} \left\{ (\sum_n Ow'^*(n)C'^*(n)C'(n)) + (\sum_n C'^*(n)Ow'(n)C'(n)) \right\} \\ &= 2\text{Re} \left\{ \frac{(\sum_n Ow'^*(n)C'^*(n)C_0(n))(\sum_n C'(n)C_0^*(n))}{(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)} \right\} \\ &\quad - \frac{|\sum_n C'^*(n)C_0(n)|^2}{(\sum_n |C'(n)|^2)(\sum_n |C_0(n)|^2)^2} 2\text{Re} \left\{ \frac{\sum_n Ow'^*(n)C'^*(n)C'(n)}{\sum_n |C'(n)|^2} \right\} \end{aligned} \quad (67)$$

ここから、励起状態を求めていく。

7 シミュレーション方法

今回は一次元固定端ボーズハバードモデルを使い、格子点は M 、粒子数は N で固定した。ハミルトニアン

$$\hat{H} = -J \sum_{i,j} \hat{a}_i \hat{a}_j^\dagger + \sum_i [V_i \hat{n}_i + \frac{U}{2} \hat{n}_i (\hat{n}_i - 1)] \quad (68)$$

の V_i は $V_i = J(i - \frac{M-1}{2})$ で計算した。ただし、 i は格子点の位置を表す。メトロポリスヘイスティング法のサンプリング数を再急降下法計算時は 1000 回とし、最終結果計算時は 10000 回とした。また、隠れ層のニューロンの数を 20 とした。再急降下法において、

$$w_i \rightarrow w_i - \alpha \frac{\partial \langle \hat{H} \rangle}{\partial w_i} \quad (69)$$

の計算を基底状態の計算時は 2000 ステップ行った。励起状態の計算時は、基底状態の状態をまず 2000 ステップで計算し、励起状態の計算を 320000 ステップで行った。 α は 0.001 とした。 $F(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda G(\mathbf{x})$ の λ はステップ数 i とすると、 $(i+1)/200$ とした。

8 結果

8.1 基底状態

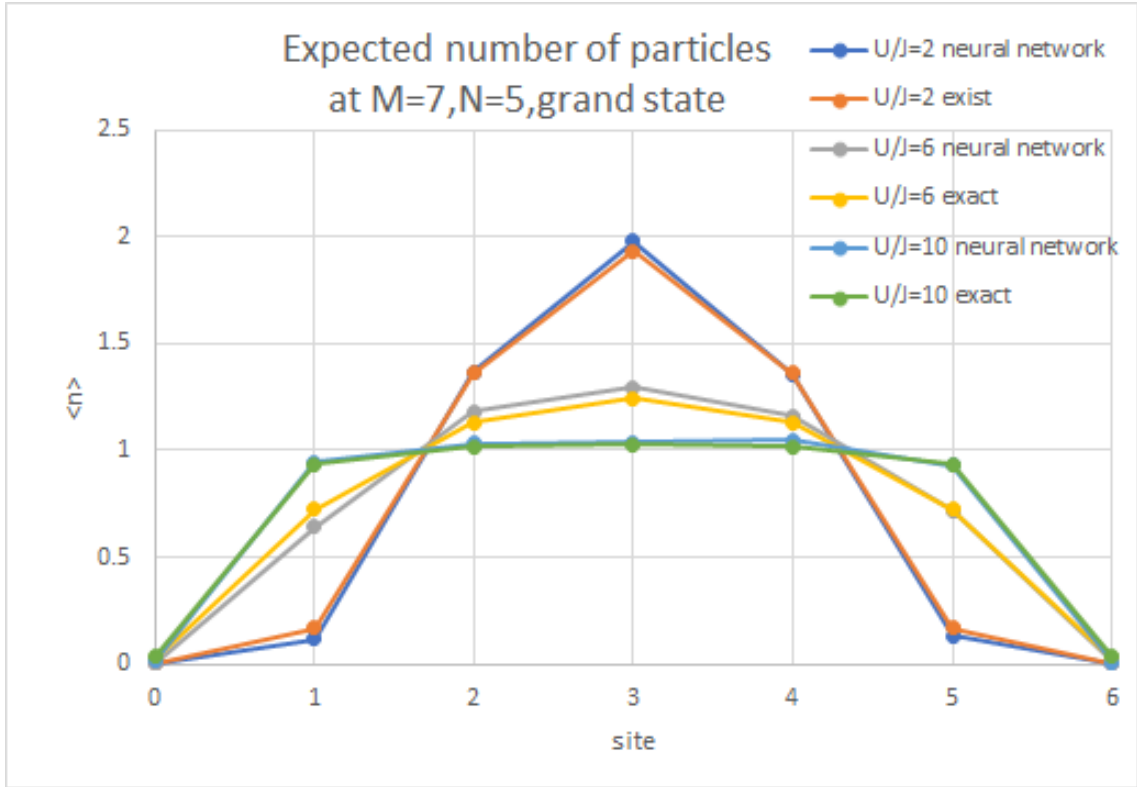


図2 厳密対角化とニューラルネットワークを利用したものの粒子数の期待値

図2は基底状態の格子点ごとの粒子数期待値を表す。U/J=2,6,10の厳密対角化と今回の方法の二つを比較している。M=7,N=5で計算した。粒子数期待値の計算は

$$\langle n_i \rangle = \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n | \hat{n}_i | n' \rangle C(\mathbf{n}')}{\sum_n |C(\mathbf{n})|^2} \quad (70)$$

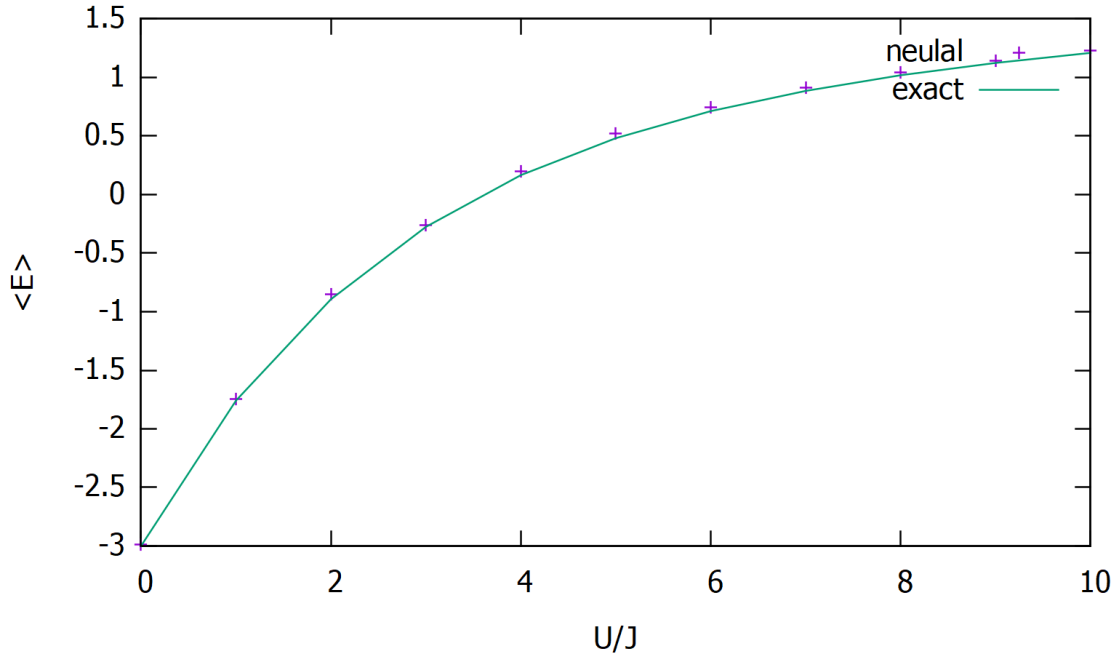


図3 厳密対角化とニューラルネットワークを利用したものの基底状態のエネルギーの期待値

で計算する。図3は基底状態の U/J が変化したときの、基底状態のエネルギー期待値を表している。 $M=3, N=3$ で計算した。エネルギー期待値は

$$\langle E \rangle = \frac{\sum_{n,n'} C^*(\mathbf{n}) \langle n | \hat{H} | n' \rangle C(\mathbf{n}')}{\sum_n |C(\mathbf{n})|^2} \text{で計算する。} \quad (71)$$

8.2 励起状態

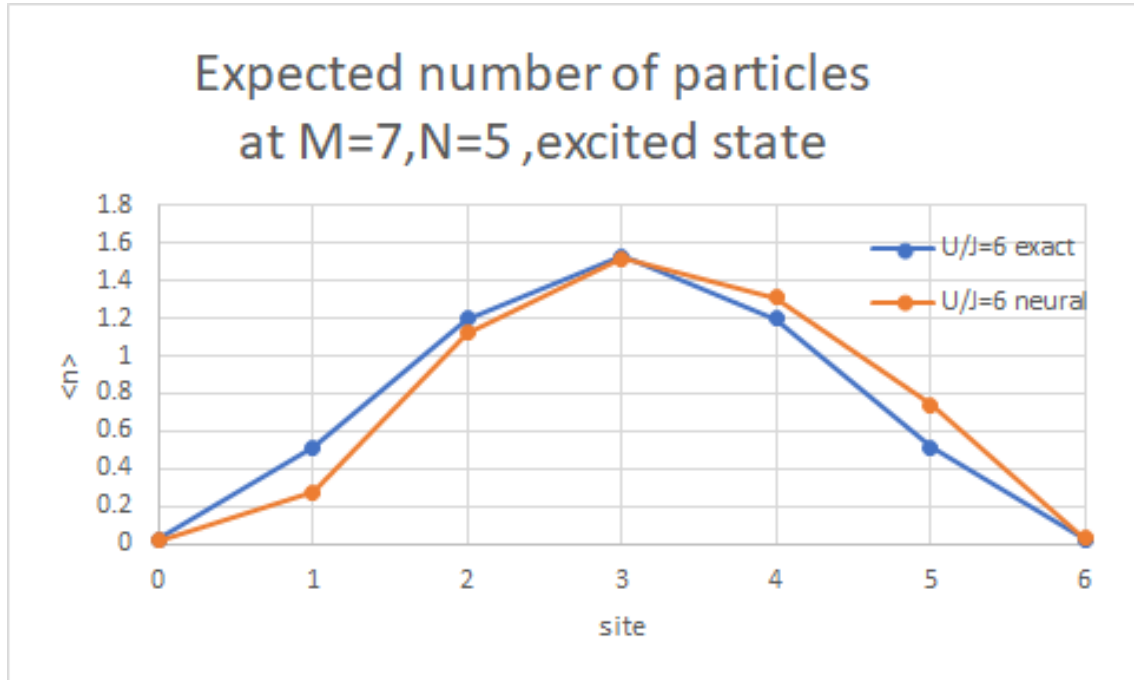


図4 厳密対角化とニューラルネットワークを利用したものの粒子数の期待値

図4は励起状態の格子点ごとの粒子数期待値を表す。U/J=6の厳密対角化と今回の方法の二つを比べている。M=7,N=5で計算した。 $|\sum_n Cn_{exact} * Cn_{neural}|$ の値は0.918となった。

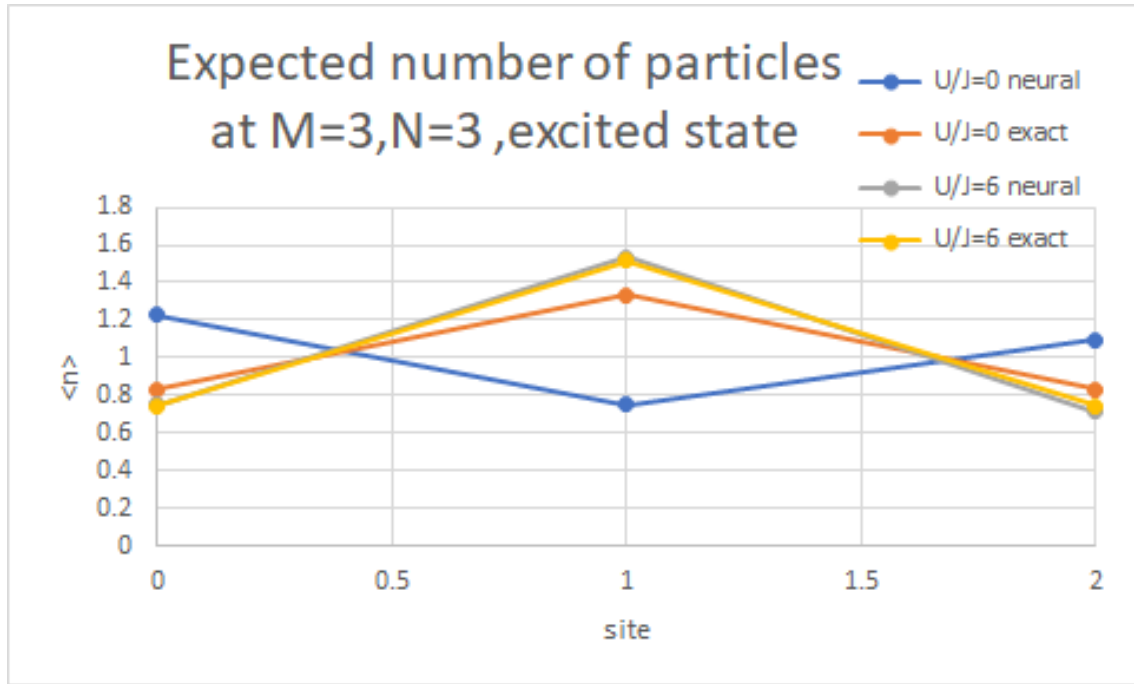


図 5 厳密対角化とニューラルネットワークを利用したものの粒子数の期待値

図 5 は励起状態の格子点ごとの粒子数期待値を表す。U/J=0,6 の厳密対角化と今回の方法の二つを比較している。M=3,N=3 で計算した。

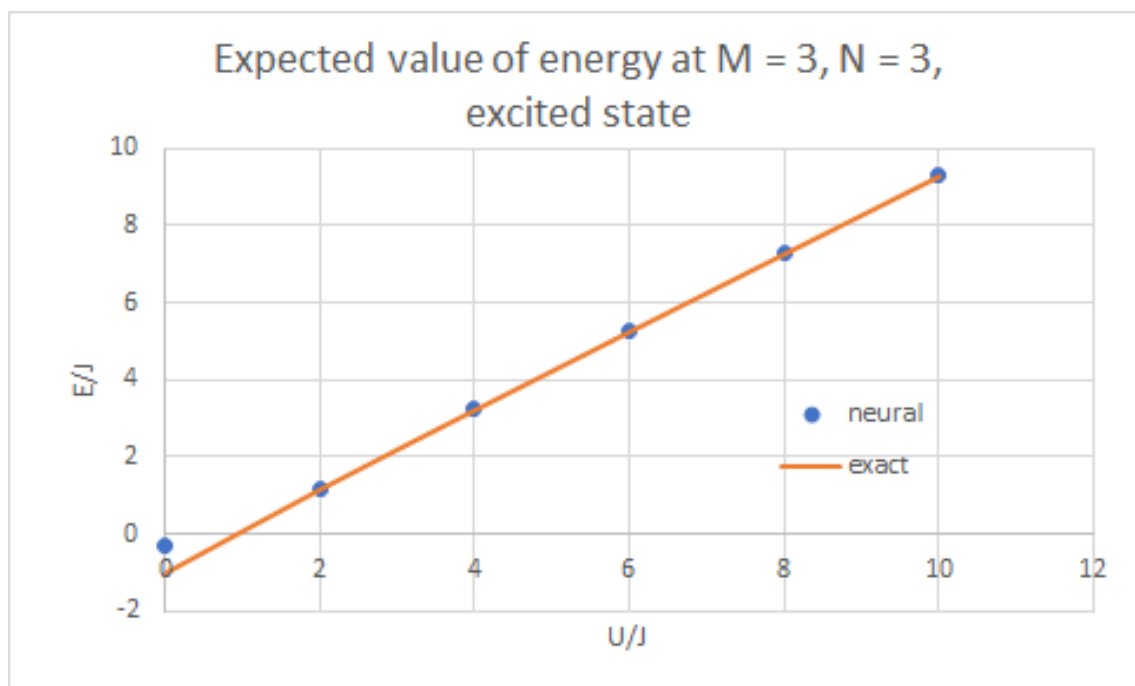


図6 厳密対角化とニューラルネットワークを利用したものの励起状態のエネルギーの期待値

図6は基底状態の U/J が変化したときの、励起状態のエネルギー期待値を表している。

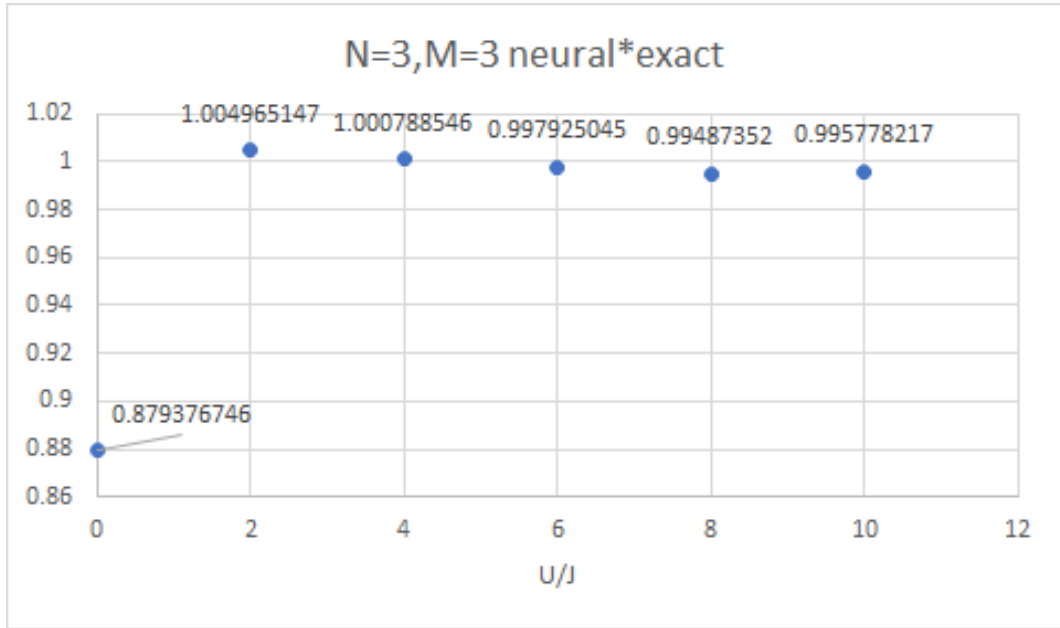


図 7 厳密対角化とニューラルネットワークの波動関数の積

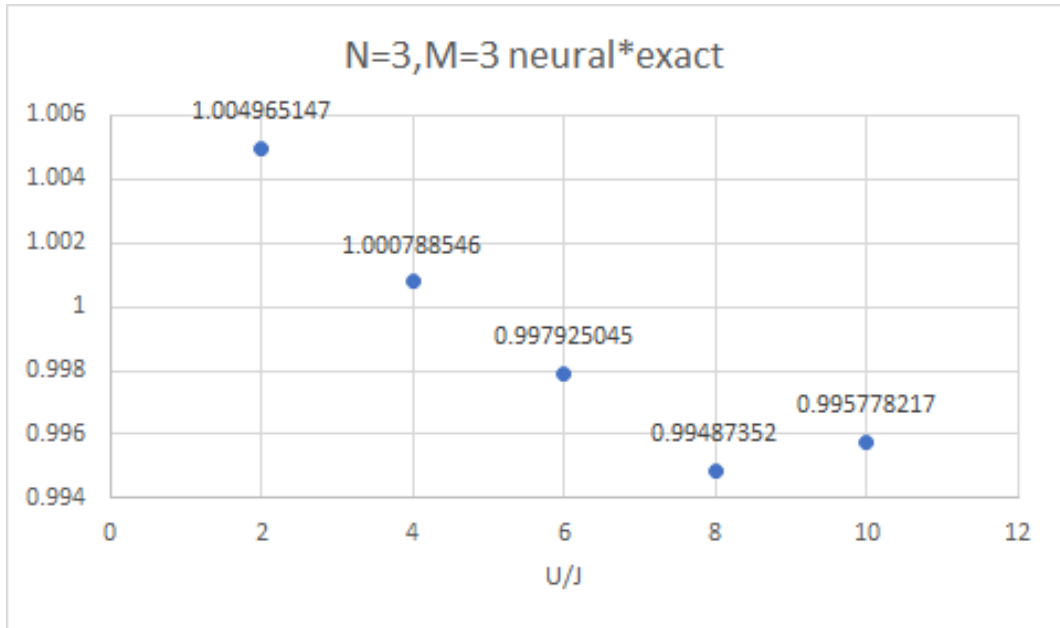


図 8 厳密対角化とニューラルネットワークの波動関数の積

図 7 図 8 は $|\sum_n C_{n_{exact}} * C_{n_{neural}}|$ の値である。

9 考察

9.1 基底状態

U が大きいと粒子が反発しあい、各格子点に一粒子ずつ入るが、 J が大きいと粒子移動しやすくなり、外場にいぞんして中心によったといえる。図 2 において、厳密対角化と、ニューラルネットワークを利用したものを比べた精度は、定性的な考察をするには十分な精度であるといえるが、厳密な計算をするには精度が悪いといえる。先行研究ではより精度が保たれており、研究課題として残った。

図 3 において、厳密対角化と、ニューラルネットワークを利用したものを比べた精度は、十分あるといえる。ニューラルネットワークを利用したもののほうがエネルギーが大きかった。これは最低値を求めるのに、最低値から少しだけずれた値をとったためであると思われる。計算速度は厳密対角化に比べて、ニューラルネットワークを利用したものの非常に遅く、これも課題である。

9.2 励起状態

図 4 において、厳密対角化と、ニューラルネットワークを利用したものを比べた精度は、あまりよくないといえる。図 6 において、厳密対角化と、ニューラルネットワークを利用したものを比べた精度は、十分あるといえる。計算速度は厳密対角化に比べて、ニューラルネットワークを利用したものの非常に遅く、これも課題である。また、計算の失敗確率が非常に高く、 $N=3, M=3$ では数度に一回失敗し、 $N=5, M=7$ では数度に一回しか成功しなかった。

10 まとめ

このように人工知能の応用は物理学のような基礎学問領域においても可能である。このような物理学と機械学習の融合の試みは、成功すれば、近似と厳密計算が主な手法だった物理学に、新たな手法をもたらすこととなり、意義がある。しかし今のところ、先行研究では速い計算速度を実現しているものの、本研究では計算速度が遅く、課題である。

謝辞

この研究をするにあたって様々なことを教え協力してくださった、二国先生、渡部先生、他二国研究室の院生、学部生たちに感謝申し上げます。

参考文献

- [1] wikipedia 機械学習 <https://ja.wikipedia.org/wiki/%E6%A9%9F%E6%A2%B0%E5%AD%A6%E7%BF%92>
- [2] Excel でわかるディープラーニング超入門
- [3] 深層学習 近代科学社
- [4] hiroki saito solving the bose-hubbard model with machine learning
- [5] wikipedia メトロポリス・ヘイスティングス法
- [6] 非線形計画法 (II) : 制約付き問題の最適化法 <http://dse.ssi.ist.hokudai.ac.jp/~onosato/lectures/Opt19/sec06.pdf>