1 次元調和振動子の Schrödinger 方程式を機械学習で解く

中村 祐介 (2022 年 11 月 19 日)

概要

論文 [1] を参考に、1 粒子 1 次元調和振動子の Schrödinger 方程式の基底状態を求める。数値計算の精度を調べるため、厳密解が分かっている系を扱うが、学習には厳密解の情報は一切使わない。

学習は 2 段階に分けて行う。step1 では基底状態に近いと期待される波動関数の候補 $\Psi_{\text{train}}(x)$ に近づくように、学習させる。具体的には出力された波動関数 $\psi(x)$ と $\Psi_{\text{train}}(x)$ との重なり積分が最大化するように、学習させる。step2 では波動関数 $\psi(x)$ によって計算するエネルギー期待値を最小化するように、学習させる。つまり step1 は計算が発散するのを防ぐための前処理であり、step2 が本命の学習である。

1 対象と表記

1.1 モデル

ハミルトニアンと厳密解は

$$\hat{H} = -\frac{\mathrm{d}^2}{\mathrm{d}x^2} + x^2\,, \qquad \Psi_{\text{exact}}(x) = \pi^{-1/4} e^{-x^2/2}\,, \qquad \text{最小固有値} = 1 \tag{1}$$

それに対して

$$\Psi_{\text{train}}(x) = \sqrt{\max\left(0,\ \frac{1}{2} - \frac{|x|}{4}\right)} \tag{2}$$

と選ぶ。$\Psi^2_{\text{train}}(x)$ は三角形になっている。この形である必然性は一切ないが、数値計算の堅牢性の確認のために、あえて下手に選んでみた。

1.2 ネットワーク

図 1 のような 1 層の隠れ層を有するニューラルネットを用いる。このネットワークの目標は「座標 x を入力したら、$\Psi_{\text{exact}}(x)$ が出力する」である。波動関数の一点だけが分かるだけだが、このネットワークが完成すれば、Simpson 積分やモンテカルロ積分を通して、任意の期待値が計算できるようになる。

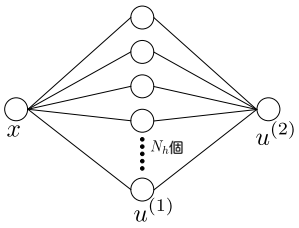

図 1　ネットワークの構造

具体的な関係式は以下の通り：

$$u_j^{(1)} = w_j^{(1)} x + b_j^{(1)}\,, \qquad u^{(2)} = \sum_{j=1}^{N_h} w_j^{(2)} \tanh u_j^{(1)} + b^{(2)}\,, \qquad \psi = \exp u^{(2)}\,, \tag{3}$$

ここで x は入力、ψ が出力であり、N_h は隠れ層のユニット数である。$w_j^{(1)}, w_j^{(2)}, b_j^{(1)}, b^{(2)}$ はニューラルネットワークの重みであり、これを学習で決定する。論文 [1] には $b^{(2)}$ がないが、ここでは数

値計算の安定性のために導入する。後述するが ψ を規格化する役割を果たす。物理的な意味はないが、浮動小数の発散を抑え、計算を安定させるために便利である。

活性化関数 $(\tanh, \exp)$ は論文 [1] をそのまま採用した。別の関数を模索する研究に方向性もあると思う。[1)]

2 計算式

2.1 メトロポリス法

期待値計算は論文 [1] に合わせて、モンテカルロ積分で行う。そのためのサンプリングにメトロポリス法を用いる。[2)]

確率分布関数 $P(x)$ を

$$P(x) \propto \psi^2(x)\,, \qquad \int \mathrm{d}x\, P(x) = 1 \tag{4}$$

に選ぶと、任意のオブサーバブル B について

$$\langle B \rangle = \int \mathrm{d}x\, P(x) B(x) \simeq \frac{1}{N_s} \sum_{i=1}^{N_s} B(x_i) \tag{5}$$

と計算できる。ここで N_s はサンプル数であり、$\{x_i\}$ は確率分布関数 P に従うサンプル列である。

2.1.1 小さな工夫

- ξ を確率変数として、$x_{\mathrm{new}} = x + \xi$ のようにランダムウォークで更新するか、単に $x_{\mathrm{new}} = \xi$ と更新するのはどちらが良いだろうか？格子系の量子モンテカルロなどでは、ランダムウォーク型なら $P(x)$ の再計算をする必要がなくなるので、便利である。今回の計算手法だとランダムウォーク型のメリットは少ないように思われる。
- ランダムウォーク型だと学習途中の平坦すぎる $\psi(x)$ のせいで非常に大きい x が採用されてしまい、計算が破綻したので、x が大きくなったら原点付近に引き戻す工夫が必要であった。

2.2 重なり積分

$$K = \frac{\left[\int \mathrm{d}x\, \Psi_{\mathrm{train}}(x)\psi(x)\right]^2}{\int \mathrm{d}x\, \Psi^2_{\mathrm{train}}(x) \cdot \int \mathrm{d}x\, \psi^2(x)} = \frac{\langle A \rangle^2}{\langle A^2 \rangle}\,, \qquad \text{ただし} \quad A = \frac{\Psi_{\mathrm{train}}}{\Psi} \tag{6}$$

一般に二つの関数がどれだけ似ているかは 2 乗平方積分などで評価できる:

$$K' = \int \mathrm{d}x\, \{\psi(x) - \Psi_{\mathrm{train}}(x)\}^2 \tag{7}$$

---

1) ただし適当に選んでも得るものがないので、後述するように活性化関数の役割を吟味すべきである。

2) 大きな自由度の系を狙うからモンテカルロ積分をしている。空間全体を舐めることできるのなら、Simpson 積分などでも良いが、それができるのなら、機械学習よりもっと良い方法がたくさんある。

ここで波動関数がどちらも規格化されていたら

$$K' = 2 - 2\int \mathrm{d}x\, \Psi_{\mathrm{train}}(x)\psi(x) \tag{8}$$

なので、K' を最小化することと、重なり積分 K を最大化することは本質的に同じである。K に注目した方が、規格化を考えなくて良い分優れているし、計算量も少ない。

2.3 エネルギー期待値

$$E = \frac{\int \mathrm{d}x\, \psi(x)\hat{H}\psi(x)}{\int \mathrm{d}x\, \psi^2(x)} = \left\langle \psi^{-1}\hat{H}\psi \right\rangle \tag{9}$$

$\hat{H}\psi$ の中の $\frac{\mathrm{d}^2\psi}{\mathrm{d}x^2}$ は中央差分で計算できる。つまり

$$\psi^{-1}(x)\hat{H}\psi(x) = \psi^{-1}(x)\left[+\left(\frac{2}{(\Delta x)^2} + x^2\right)\psi(x) - \frac{\psi(x+\Delta x) + \psi(x-\Delta x)}{(\Delta x)^2}\right] \tag{10}$$

$$= \frac{2}{(\Delta x)^2} + x^2 - \frac{\psi(x+\Delta x) + \psi(x-\Delta x)}{(\Delta x)^2\psi(x)} \tag{11}$$

ここで Δx は適切な小さい数である。[3)]

2.4 微分

重みパラメータ $w_j^{(1)}, w_j^{(2)}, b^{(1)}$ を総称して w と書く。ここで便利な表記 O_w を以下のように導入する：

$$O_w = \psi^{-1}\frac{\partial\psi}{\partial w} \tag{12}$$

すると

$$\frac{\partial K}{\partial w} = 2K(\langle AO_w\rangle\langle A\rangle - \langle O_w\rangle) \tag{13}$$

$$\frac{\partial E}{\partial w} = 2\left\langle O_w\psi^{-1}\hat{H}\psi\right\rangle - 2\langle O_w\rangle\left\langle\psi^{-1}\hat{H}\psi\right\rangle \tag{14}$$

と計算できる。

さて、今のモデルでは

$$O_w = \psi^{-1}\frac{\partial\psi}{\partial u^{(2)}}\frac{\partial u^{(2)}}{\partial w} = \frac{\partial u^{(2)}}{\partial w} \tag{15}$$

となってることに注目。よって

$$O_{w_j^{(2)}} \equiv \frac{\partial u^{(2)}}{\partial w_j^{(2)}} = \tanh u_j^{(1)} \tag{16}$$

$$O_{b_j^{(1)}} \equiv \frac{\partial u^{(2)}}{\partial b_j^{(1)}} = w_j^{(2)}\left(1 - O^2_{w_j^{(2)}}\right) \tag{17}$$

$$O_{w_j^{(1)}} \equiv \frac{\partial u^{(2)}}{\partial w_j^{(1)}} = O_{b_j^{(1)}}\, x \tag{18}$$

---

3) 論文で $\frac{\mathrm{d}^2}{\mathrm{d}x^2}$ をどのように処理しているか不明だが、真面目にネットワークの出力を入力で 2 回微分するのは、大変な気がする。

となる。

以上、必要なすべての期待値が計算できるようになった。このくらいなら高級な誤差逆伝播法に頼らない方が間違いにくいと思う。

2.5 計算手順

1. 重みパラメータを初期化する。$b_j^{(1)}$ と $b^{(2)}$ とはゼロに、$w_j^{(1)}$ と $w_j^{(2)}$ は標準偏差が $1/\sqrt{N_h}$ の正規分布に従う割合でランダムに初期化する。[4)]
2. メトロポリス法で $\psi^2(x)$ に従う割合でサンプル列 $\{x_i\}$ を生成する。
3. K を最大化または H を最小化するように、重みパラメータ $b_j^{(1)}$、$w_j^{(1)}$、$w_j^{(2)}$ を更新する。更新は SGD や Adam 法などで行う。
4. $\frac{1}{N_s}\sum_i^{N_s}\psi^2(x_i) \simeq 1$ となるように緩く規格化をする。本来波動関数は規格化されている必要はないが、$u^{(2)}$ が 300 を超え始めると、浮動小数がオーバーフローするのでそれを防止するためである。それだけなので、このタイミングで簡単に

$$b^{(2)} \leftarrow b^{(2)} - \frac{1}{2}\log\left(\frac{\sum_i^{N_s}\psi^2(x_i)}{N_s}\right) \tag{19}$$

とすれば良い。この根拠は以下の通り：もし $\psi = e^u$ のノルム 2 乗が C であったらならば、規格化は $\psi \leftarrow \frac{1}{\sqrt{C}}\psi = \exp\left(u - \frac{1}{2}\log C\right)$ である。これは b を $b - \frac{1}{2}\log C$ に置き換えたことに対応する。厳密な規格化は不要なので ψ の値は、重みパラメータの更新前のもので十分であろう。

5. 満足するまで手順 2 に戻る。

3 問題提起

3.1 なぜこの方法で解けるか？

今回の系では $u^{(2)}(x) = -x^2/2$ となれば厳密に解けたことになる。ネットワークの構造から $\tanh u_j^{(1)}(x)$ の線型結合で $u^{(2)}(x)$ が構成されているので、$\{\tanh u_j^{(1)}\}$ が完全系をなすことが期待されている。実際に完全系にすることは不可能だが、できるだけ上手に選ぶべきである。上手に選べていれば、任意の関数を表現できるので、あとは重みパラメータを上手に選ぶ問題に帰着する。

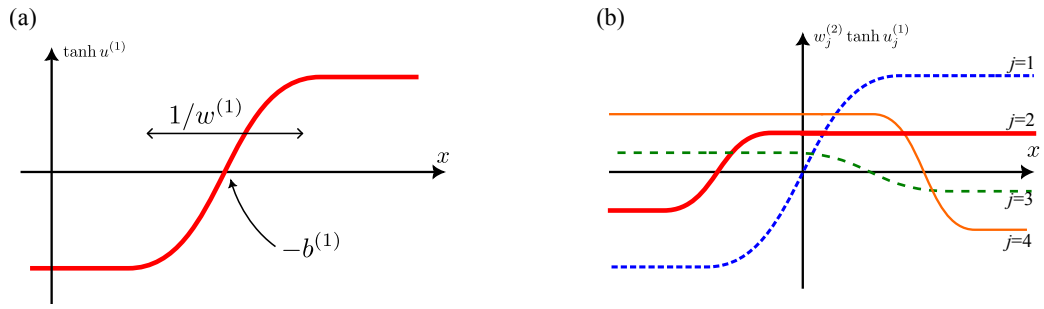

図 2 (a) $\tanh u^{(1)}(x)$ **の形。 (b)** $w_j^{(2)}\tanh u_j^{(1)}(x)$ **のイメージ。これらの重ね合わせが** $-x^2/2$ **になるように重みパラメータを調整していく。**

4) 文献 [2] の p.182 の Xavier の初期値

さて、活性化関数 $f^{(1)}(x) = \tanh(x)$ と $u_j^{(1)}(x) = w_j^{(1)}x + b_j^{(1)}$ の構造から、重みパラメータ $w^{(1)}$ は $f^{(1)}$ の幅、$b_j^{(1)}$ は $f^{(1)}$ の中心位置を意味することが分かる [図 2(a)]。それを $w^{(2)}$ の重さで重ね合わせて $u^{(2)}$ を作ろう、という訳である [図 2(b)]。果たしてこれは最善だろうか？活性化関数 $f^{(1)}(x) = \tanh(x)$, $f^{(2)} = \exp(x)$ をもっと賢く選ぶことができそうである。

主なコメントは以下の通り：

1. $u^{(2)}$ の目標が偶関数なのに、$f^{(1)}$ を奇関数にするメリットが不明である。
2. 複数の j に対して、x が共通になった場合、実質的にユニット数が減ることになり、無駄である。一度 x が同じになってしまったら抜け出せない可能性がある。
3. $f^{(1)}$ を全ての j に対して共通にする必要性がないと思われる。$f^{(1)}$ に j 依存性を持たせてはどうだろうか？

例えば x を Δx で差分化した上で、$w^{(1)} = 1, b_j^{(1)} = 0$ に固定し、活性化関数を j 依存させ、$f_j^{(1)} = \delta_{x,j\cdot\Delta x}, f_j^{(2)}(x) = x$ のようにする。つまり Kronecker delta の重ね合わせで波動関数を作る。するとこれはもはや、差分化後の行列の対角化問題と等価になる（多分)。ただし行列対角化に関する有力な方法を用いないので、メリットはほとんど何もないだろう。しかしこのことから分かる通り、ニューラルネットワークを用いた方法は、いわゆる差分法を含んでいるのである。従って、より効率的な方法を含んでいる可能性がある。

3.2 改良案

活性化関数の役割をはっきりとさせた上で、改良することができそうである。すぐに思いつく改良案は以下の通り。Bose–Hubbard 模型でも同様に工夫できる。いろいろ試して比較するのは、卒論テーマにどうだろうか？

1. 波束の重ね合わせで、波動関数を作る： $f^{(1)}(x) = e^{-x^2}$, $f^{(2)}(x) = x$
2. Taylor 展開のノリで作る： $f_j^{(1)}(x) = x^{j-1}$, $f^{(2)}(x) = x$ (または e^x)
3. エルミート多項式 $H_j(x)$ で展開する： $f_j^{(1)}(x) = H_j(x)$, $f^{(2)}(x) = e^{-x}$

3.3 その他

- 隠れ層の $u_j^{(1)}$ のヒストグラムを監視すると、活性化関数や重みの初期分布が適切か分かるかも？ [5)]
- GP 方程式を解く場合は、規格化と化学ポテンシャルの調整が重要になる。虚時間発展法の知見で、アルゴリズムの提案はすぐできるだろう。たぶん誰もやっていない？

参考文献

[1] Hiroki Saito, J. Phys. Soc. Jpn. **87**, 074002 (2018).

[2] 斉藤 康毅, ゼロから作る Deep Learning, オライリー (2016).

5) 文献 [2] の p.179